

T.C.  
DOKUZ EYLÜL ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜSÜ  
İŞLETME ANABİLİM DALI  
ÜRETİM YÖNETİMİ VE ENDÜSTRİ İŞLETMECİLİĞİ PROGRAMI  
YÜKSEK LİSANS TEZİ

**ÇOK MODLU BİR ULAŞTIRMA SİSTEMİ  
ALGORİTMASI GELİŞTİRİLMESİ VE PROJE YÖNETİMİ  
KAPSAMINDA PROJELENDİRİLMESİ**

**Beyazıt B. DEMİR**

Danışman  
**Doç. Dr. Özlem DOĞAN**

2010

**YÜKSEK LİSANS**  
**TEZ/ PROJE ONAY SAYFASI**

2003800070

**Üniversite** : Dokuz Eylül Üniversitesi  
**Enstitü** : Sosyal Bilimler Enstitüsü  
**Adı ve Soyadı** : Beyazıt Bestami DEMİR  
**Tez Başlığı** : Çok Modlu Bir Ulaştırma Sistemi Algoritması Geliştirilmesi ve Proje Yönetimi Kapsamında Projelendirilmesi  
**Savunma Tarihi** : 29.09.2010  
**Danışmanı** : Doç.Dr.Özlem DOĞAN

**JÜRİ ÜYELERİ**

| <b><u>Ünvanı, Adı, Soyadı</u></b> | <b><u>Üniversitesi</u></b> | <b><u>İmza</u></b>  |
|-----------------------------------|----------------------------|---|
| Doç.Dr.Özlem DOĞAN                | - DOKUZ EYLÜL ÜNİVERSİTESİ |  |
| Doç.Dr.Hilmi YÜKSEL               | - DOKUZ EYLÜL ÜNİVERSİTESİ |  |
| Yrd.Doç.Dr.Habil GÖKMEN           | - DOKUZ EYLÜL ÜNİVERSİTESİ |  |

Oybirliği (X)

Oy Çokluğu ( )

Beyazıt Bestami DEMİR tarafından hazırlanmış ve sunulmuş "**Çok Modlu Bir Ulaştırma Sistemi Algoritması Geliştirilmesi ve Proje Yönetimi Kapsamında Projelendirilmesi**" başlıklı Tezi ( ) / Projesi ( ) kabul edilmiştir.

**Prof.Dr. Utku UTKULU**  
**Enstitü Müdürü**

## Yemin Metni

Yüksek Lisans Tezi olarak sunduđum “**Çok Modlu Bir Ulařtırma Sistemi Algoritması Geliřtirilmesi ve Proje Yönetimi Kapsamında Projelendirilmesi**” adlı çalıřmanın, tarafımdan, bilimsel ahlak ve geleneklere aykırı düřecek bir yardıma bařvurmaksızın yazıldıđını ve yararlandıđım eserlerin bibliyografyada gösterilenlerden olduđunu bunlara atıf yapılarak yararlanılmıř olduđunu belirtir ve bunu onurumla dođrularım.

...../...../2010

Beyazıt B. DEMİR

## ÖZET

Yüksek Lisans Tezi

Çok Modlu Bir Ulaştırma Sistemi Algoritması Geliştirilmesi ve Proje Yönetimi  
Kapsamında Projelendirilmesi

Beyazıt B. DEMİR

Dokuz Eylül Üniversitesi  
Sosyal Bilimler Enstitüsü  
İşletme Anabilim Dalı

Üretim Yönetimi ve Endüstri İşletmeciliği Programı

Günümüzde ihtiyaç duyulan bütün kaynaklar, projelerin önceden belirlenen süre, maliyet ve nitelikler dâhilinde tamamlanmasını sağlamak amacıyla bir araya getirilmektedir. İçlerinde yer alan faaliyetlerin tekrarsız ve kendine özgü yapılarının yarattığı karmaşıklığa karşın projeler; belirlenmiş hedeflere, öngörülen bir zaman içinde ve üzerinde uzlaşa sağlanmış bir bütçe dâhilinde ulaşmayı amaçladığından, başlatılmaları yapısal organizasyon değişikliklerini ya da yeni organizasyon oluşumunu zorunlu kılmaktadır.

Proje yönetimi, içinde genel yönetim esaslarını bulundurmakla birlikte olağan işletme faaliyetleri yönetiminden oldukça farklı yapıdadır. Özel organizasyon biçimlerinin teşkil edilmesini, farklı planlama ve kontrol yöntemlerinin kullanılmasını ve özel eğitim alınmasını gerektirmektedir.

Lojistik işlev sahaları içinde yer alan taşıma faaliyetleri günümüzde uygun şekilde ve gerçekçi planlanmadığı takdirde maliyetleri önemli oranda artırmaktadır. Bunun önüne geçilmesi maliyet etkinliğini göz önüne alarak planlama yapılmasına imkân sağlayan ve kullanıcılara karar desteği sağlayan yazılımların tedarik edilmesi ile mümkün olabilmektedir.

Bu yazılımların lojistik organizasyonlarda arzu edilen etkiyi sağlayabilmesi kullanıcı ihtiyaçlarının üst seviyede ayrıntılandırılması, müşterinin gerçekte ne istediğinin net ve açık şekilde ortaya konması ve uygun kaynak, yetki ve yapılanmaların sağlanmasını gerektirir. Bu noktada proje yönetimi temel prensip ve usullerini kullanmakla birlikte farklı alanlardan uzmanları içine alan özel yapılar oluşturulmasını gerekli kılan yazılım proje yönetimi devreye girmektedir.

**Bu alıřmada ncelikle btn ulařtırma eřitlerinin aynı anda (multimodal), bir arada arka arkaya (intermodal), periyodik (kurye/ring) ya da birbiri iinde (combined) tařınmalarına imkn saėlayan maliyet etkin bir ulařtırma modeli geliřtirilmeye ve mteakiben proje ynetim tasarımı hazırlık ve uygulama sreleri kapsamında uygulanmaya hazır bir yazılım projesine dnřtrlmeye alıřılmıřtır.**

**Anahtar Kelimeler: Proje Ynetimi, Lojistik İřlev Sahaları, Yazılım Proje Ynetimi, Maliyet Etkinliėi.**

## **ABSTRACT**

### **Master Thesis**

**Development of a Multi-Modal Transportation System Algorithm and  
Realization as a Project Under the Project Management Considerations**

**Beyazıt B. DEMİR**

**Dokuz Eylül University  
Institute of Social Sciences  
Department of Business Administration  
Production Management and Industrial Business Administration Program**

Today, to finalize projects within the pre-determined time, cost and specs limits, all needed resources are brought together. Because projects, despite the complexities caused by their non-repeated and self-styled structures, aim to reach the determined goals within an accepted time frame and fully decided budget, their commencements force structural changes in the organization to be made or new organizations to be created.

Project management, having general management procedures in itself, is very different in structure compared to other routine business administration activities. It requires creation of special organizational structures, usage of different planning and control methods and private education.

Today to refrain from extra costs, transportation activities, as a part of logistic functional areas, need to be planned realistically and accordingly, and softwares that assure cost-efficient transportation planning and give decision support to users need to be procured.

In order these softwares to perform desired effect on logistic organizations, to set highly detailed user requirements, to figure out what the costumer really want clearly and openly and to ensure optimum source authority and restructuring are needed. At this point, we see software project management which requires special structures made up of specialists from different fields, besides using basic project management principles and techniques.

In this study, primarily a cost-efficient transportation model by which all transportation modes to be used at the same time (multimodal), together

**but successively (intermodal), periodically (courier/ring) or one inside another (combined) is aimed to be developed and then designed, prepared and handled so as to be a ready-to-implement software project within the project management procedures and considerations.**

**Key Words: Project Management, Logistic Functional Areas, Software Project Management, Cost-Efficiency.**

**ÇOK MODLU BİR ULAŞTIRMA SİSTEMİ ALGORİTMASI GELİŞTİRİLMESİ VE  
PROJE YÖNETİMİ KAPSAMINDA PROJELENDİRİLMESİ**

|                         |              |
|-------------------------|--------------|
| <b>TEZ ONAY SAYFASI</b> | <b>İi</b>    |
| <b>YEMİN METNİ</b>      | <b>İii</b>   |
| <b>ÖZET</b>             | <b>İv</b>    |
| <b>ABSTRACT</b>         | <b>İvi</b>   |
| <b>İÇİNDEKİLER</b>      | <b>İviii</b> |
| <b>KISALTMALAR</b>      | <b>İxi</b>   |
| <b>TABLolar LİSTESİ</b> | <b>İxii</b>  |
| <b>ŞEKİLLER LİSTESİ</b> | <b>İxiii</b> |
| <b>EKLER LİSTESİ</b>    | <b>İxv</b>   |

**İÇİNDEKİLER**

|              |          |
|--------------|----------|
| <b>GİRİŞ</b> | <b>1</b> |
|--------------|----------|

**BİRİNCİ BÖLÜM**

**PROJE VE PROJE YÖNETİMİ**

|  |           |
|--|-----------|
| <b>1.1. PROJE VE PROJE YÖNETİMİ KAVRAMLARI</b>                                 | <b>3</b>  |
| 1.1.1. Proje Kavramı ve Özellikleri  | 3         |
| 1.1.2. Proje Yönetimi Kavramı ve Özellikleri                                   | 5         |
| 1.1.3. Proje Ömür Devri  | 8         |
| 1.1.4. Projelerin Aşamaları  | 11        |
| 1.1.5. Proje Yönetimine Olan İhtiyaç ve Proje Yönetiminin Sağlayacağı Faydalar | 13        |
| <b>1.2. PROJE ÖRGÜT YAPILARI</b>   | <b>16</b> |
| 1.2.1. İşlevsel Örgütlenme   | 17        |
| 1.2.2. Proje Esaslı Örgütlenme   | 19        |
| 1.2.3. Matris Örgütlenme   | 20        |
| <b>1.3. PROJE YÖNETİMİNDE BAŞARISIZLIK VE NEDENLERİ</b>                        | <b>23</b> |
| <b>1.4. PROJE YÖNETİMİ BİLGİ ALANLARI</b>                                      | <b>25</b> |
| 1.4.1. Bütünleştirme (Integration) Yönetimi                                    | 25        |
| 1.4.2. Kapsam Yönetimi   | 25        |
| 1.4.3. Zaman Yönetimi  | 26        |



|                                  |    |
|----------------------------------|----|
| 1.4.4. Maliyet Yönetimi          | 27 |
| 1.4.5. Kalite Yönetimi           | 27 |
| 1.4.6. İnsan Kaynakları Yönetimi | 27 |
| 1.4.7. İletişim Yönetimi         | 28 |
| 1.4.8. Tedarik Yönetimi          | 29 |
| 1.4.9. Risk Yönetimi             | 29 |

## **İKİNCİ BÖLÜM**

### **PROJE YÖNETİM TEKNİKLERİ**

|   |    |
|---|----|
| 2.1. GENEL  | 31 |
| 2.2. PROJE YÖNETİM TEKNİKLERİ                                   | 32 |
| 2.2.1. Proje Kontrol Tabloları                                  | 32 |
| 2.2.2. İş Dağılım Ağacı (Work Breakdown Structure-WBS)          | 33 |
| 2.2.3. GANTT Şemaları ve Denge Hattı Tekniği                    | 34 |
| 2.2.4. Ağ Analizi (Network Analysis) Teknikleri                 | 36 |
| 2.2.4.1. Kritik Yol Metodu (CPM)                                | 37 |
| 2.2.4.2. Program Değerlendirme ve Gözden Geçirme Tekniği (PERT) | 38 |
| 2.2.4.3. Ağ Analizi Kullanılma Esasları                         | 40 |
| 2.2.4.4. Ağ Analizi ile Sağlanabilecek Faydalar                 | 41 |
| 2.2.5. Proje Planlama Diyagramları (PPD)                        | 42 |
| 2.2.6. Kayma Şemaları   | 43 |
| 2.2.7. Proje Yönetim Tekniklerindeki Son Gelişmeler             | 44 |
| 2.2.7.1. CPM/PERT Tekniklerinin Sınırlılıkları                  | 45 |
| 2.2.7.2. Proje Programlama                                      | 45 |
| 2.2.7.3. Faaliyet Sürelerindeki Belirsizlikler                  | 46 |
| 2.2.7.4. Zaman Maliyet Optimizasyonu                            | 47 |
| 2.2.7.5. Kaynakların Tahsisi                                    | 47 |
| 2.2.7.6. Yönetim  | 47 |
| 2.2.7.7. Proje İzleme ve Kontrol                                | 48 |

## **ÜÇÜNCÜ BÖLÜM**

### **YAZILIM PROJE YÖNETİMİ**

|  |    |
|--|----|
| 3.1. BİLGİ SİSTEMLERİ                                | 49 |
| 3.1.1. İşlemsel Bilgi Sistemleri (Yazılımları) (İBS) | 51 |
| 3.1.2. Ofis Otomasyon Sistemleri (Yazılımları) (OOS) | 51 |

|          |  |    |
|----------|--|----|
| 3.1.3.   | Bilgi Tabanlı İş Sistemleri (Yazılımları) (BTİS)   | 52 |
| 3.1.4.   | Karar Destek Sistemleri (Yazılımları) (KDS)  | 52 |
| 3.1.5.   | Yönetim Bilgi Sistemleri (Yazılımları) (YBS)   | 52 |
| 3.1.6.   | Üst Yönetim Bilgi Sistemleri (Yazılımları) (ÜBS)   | 53 |
| 3.2.     | <b>YAZILIM PROJELERİ</b>   | 53 |
| 3.2.1.   | Yazılım Projeleri ve Özellikleri   | 54 |
| 3.2.2.   | Yazılım Projelerinin Problemleri   | 56 |
| 3.2.3.   | Yazılım Projelerinin Başarısızlıkla Sonuçlanma Nedenleri   | 58 |
| 3.2.3.1. | Yönetimin Sorumluluğu  | 58 |
| 3.2.3.2. | İhtiyaçların Değişmesi   | 58 |
| 3.2.3.3. | Yazılım İhtiyaçlarının Tam olarak Anlaşılmasına  | 59 |
| 3.2.3.4. | Yazılım Karmaşıklığı   | 59 |
| 3.2.3.5. | Eksik Tahminler  | 60 |
| 3.2.3.6. | Yazılım Projelerinde Sıkça Yapılan Hatalar   | 60 |
| 3.2.4.   | Yazılım Projelerinde Doğru Eleman Seçimi ve Yazılım Proje Ekibinin Oluşumu ve Gelişimi Aşamaları | 62 |
| 3.2.5.   | Yazılım Proje Geliştirme ve Yönetim Süreci   | 62 |
| 3.2.6.   | Yazılım Geliştirme Model, Metodoloji ve Teknikleri   | 63 |
| 3.2.7.   | Yazılım Proje Yönetimi Aşama ve Adımları   | 63 |
| 3.2.8.   | Yazılım Proje Yönetimini Destekleyen Disiplinler   | 63 |
| 3.2.8.1. | Değişiklik Yönetimi  | 63 |
| 3.2.8.2. | Yazılım Konfigürasyon Yönetimi   | 64 |
| 3.2.8.3. | Yazılım Risk Yönetimi  | 66 |

## **DÖRDÜNCÜ BÖLÜM**

### **ÇOK MODLU ULAŞTIRMA SİSTEMİ**

|        |  |    |
|--------|--|----|
| 4.1.   | <b>AMAÇ</b>  | 69 |
| 4.2.   | <b>KAPSAM</b>  | 69 |
| 4.3.   | <b>KONSEPT</b>   | 69 |
| 4.4.   | <b>SİSTEM PLANLAMA ÖZELLİKLERİ</b>                               | 70 |
| 4.4.1. | İdari Planlama   | 70 |
| 4.4.2. | Lojistik (Taşıma) Planlar ile İdari Planların İlişkilendirilmesi | 72 |

|          |  |    |
|----------|--|----|
| 4.4.3.   | Lojistik (Taşıma) Planlama                             | 73 |
| 4.4.3.1. | Bir Defaya Mahsus Taşımaların Planlanması              | 74 |
| 4.4.3.2. | Tekrarlı Taşımaların Planlanması                       | 77 |
| 4.5.     | <b>ALGORİTMADAN BEKLENEN KARAR DESTEK İŞLEVLERİ</b>    | 78 |
| 4.5.1.   | Maliyet Optimizasyonu                                  | 78 |
| 4.5.2.   | Güzergâh Optimizasyonu                                 | 78 |
| 4.5.2.1. | Ulaştırma Altyapı Optimizasyonu                        | 78 |
| 4.5.2.2. | Kullanılacak Güzergâh Optimizasyonu                    | 79 |
| 4.5.3.   | Ulaştırma Aracı Optimizasyonu                          | 81 |
| 4.5.4.   | Yükleme Optimizasyonu                                  | 82 |
| 4.5.5.   | Taşıma Optimizasyonu                                   | 85 |
| 4.6.     | <b>DİĞER ALT MODÜLLER</b>                              | 86 |
| 4.6.1.   | VTYM'nin Sahip olacağı Alt Modüller                    | 86 |
| 4.6.1.1. | Coğrafi Bilgi Sistemi (CBS) Alt Modülü                 | 86 |
| 4.6.1.2. | Raporlanabilir Envanter Kodu (REK) Alt Modülü          | 89 |
| 4.6.1.3. | Taşımaya Esas Hale Getirme (Paketleme) Alt Modülü      | 89 |
| 4.6.1.4. | Taşıma Talep Alt Modülü                                | 89 |
| 4.6.1.5. | Plan Hiyerarşisi Oluşturma Alt Modülü                  | 89 |
| 4.6.1.6. | Yetkilendirme ve Güvenlik Modülü                       | 89 |
| 4.6.2.   | TPM'in Sahip Olacağı Alt Modüller                      | 90 |
| 4.6.2.1. | Plan Görüntüleme/Animasyon Alt Modülü                  | 90 |
| 4.6.2.2. | Plan Birleştirme Alt Modülü                            | 90 |
| 4.6.2.3. | Plan Simülasyon ve İstatistik Değerlendirme Alt Modülü | 90 |
| 4.6.2.4. | Plan Çatışmalarını Görüntüleme ve Giderme Alt Modülü   | 90 |
|          | <b>SONUÇ</b>   | 91 |
|          | <b>KAYNAKLAR</b>                                       | 94 |

## **KISALTMALAR**

EK-1'de sunulmuştur.

## TABLolar LİSTESİ

|   |        |
|---|--------|
| Tablo-1.1: Proje Karakteristiklerinin Projelere Göre Sınıflandırılması                      | 7      |
| Tablo-1.2: Teşkilat Yapısının Projelere Etkisi  | 17     |
| Tablo-3.1 Temel Proje Aşamaları   | 55     |
| Tablo-3.2: DACS ROMA Rapor Sonuçları  | 57     |
| Tablo-3.3: Yazılım Problemleri Örnekler   | 57     |
| Tablo-3.4: Proje Büyüklükleri ve Başarı   | 60     |
| Tablo-E2.1: Aşamalara Göre Hataların Giderilme Maliyetleri                                  | E2-44  |
| Tablo-E4.1: Yapısal Proje Yönetimine Göre On Temel Adım                                     | E4-2   |
| Tablo-E5.1: Proje Yönetim Süreçlerinin Proje Grupları ve Bilgi Alanlarına göre Düzenlenmesi | E5-48  |
| Tablo-E8.1: Ulaştırma Modları Varsayılan Dönüşüm Katsayıları                                | E8-14  |
| Tablo-E8.2: Çaprazlama Operatörünün Kullanımı Örneği  | E8-18  |
| Tablo-E13.1: Modül Bütünleştirme Testleri Genel Özellikleri                                 | E13-19 |
| Tablo-E13.2: Alfa Testi Genel Özellikleri   | E13-20 |
| Tablo-E13.3: Alfa Testi Genel Özellikleri   | E13-20 |
| Tablo-E13.4: Proje Dokümanları ve Çalışma Kapsamında Hazırlanma Durumu                      | E13-28 |
| Tablo-E13.5.: Taslak Proje Takvimi  | E13-29 |
| Tablo-E14.1 Eğitim Planı  | E14-2  |

## ŞEKİLLER LİSTESİ

|   |       |
|---|-------|
| Şekil-1.1: Proje Yönetimi Bilgi Alanları  | 6     |
| Şekil-1.2: Proje Yaşam Döngüsü  | 9     |
| Şekil-1.3: Proje Ömür Devri Aşamaları   | 10    |
| Şekil-1.4: Proje aşamaları boyunca zaman ve maliyet belirsizliklerinin değişimi | 11    |
| Şekil-1.5: Proje Ömür Devri   | 12    |
| Şekil-1.6: İşlevsel Teşkilatlanma   | 17    |
| Şekil-1-7: Proje Esaslı Teşkilatlanma   | 19    |
| Şekil-1.8:Güçlü Matris Teşkilatlanma  | 21    |
| Şekil-3.1: Bilgi Sistemi Türleri ve Organizasyonlardaki Bilgi Sistemleri        | 50    |
| Şekil-3.2: Yazılım içeren ve içermeyen sistemler arasındaki süre aşımı oranları | 57    |
| Şekil-3.3: Hataların birikimli maliyetleri                                      | 59    |
| Şekil-3.4: Projelerde Başarısızlık Nedenleri                                    | 61    |
| Şekil-3.5: Risk Yönetiminde "Yedi Ana Adım"                                     | 68    |
| Şekil-4.1: Plan Hiyerarşisi   | 71    |
| Şekil-4.2: Karar destek sistemi ve CBS entegrasyonunun değerlendirilmesi        | 88    |
| Şekil-E4.1: Yazılım Projelerinde Doğru Eleman Seçimi                            | E4-6  |
| Şekil-E5.1: Şelale Modeli   | E5-2  |
| Şekil-E5.2: Spiral model  | E5-6  |
| Şekil-E5.3: Tekrarlı Gelişim Süreci   | E5-9  |
| Şekil-E5.4: Tekrarlı ve Artırımlı Geliştirme Modeli Döngüsü                     | E5-15 |
| Şekil-E5.5:Tekrarlı Yazılım Geliştirme Yaklaşımı                                | E5-16 |
| Şekil-E5.6: SSADM Süreç Modeli  | E5-19 |

|   |        |
|---|--------|
| Şekil-E5.7: PRINCE2 Süreç Modeli  | E5-23  |
| Şekil-E5.8: Scrum İş Akışı  | E5-37  |
| Şekil-E7.1: ASP.NET Yapılanma   | E7-6   |
| Şekil-E7.2: N Katmanlı Mimari   | E7-10  |
| Şekil-E7.3: Uygulama Yazılımı Planlanan Katmanları                              | E7-12  |
| Şekil-E7.4 Sistem Veri Tabanı, Donanım ve İletişim Altyapısı Mimarisi           | E7-24  |
| Şekil-E8.1: Genetik Algoritma Akış Diyagramı                                    | E8-6   |
| Şekil-E8.2: Kromozom Yapısı   | E8-12  |
| Şekil-E8.3: Simetrik Üçgen Dağılımı   | E8-13  |
| Şekil-E8.4: Algoritma İş Akış Şeması  | E8-15  |
| Şekil-E8.5: Alt-Eniyileme Bileşenleri ile Sistem Simulasyonu Etkileşimi         | E8-20  |
| Şekil-E8.6: Kısıt Gevşetme Algoritması  | E8-28  |
| Şekil-E9.1: Serim örneği  | E9-2   |
| Şekil-E9.2: Üç ring için bir serim örneği                                       | E9-6   |
| Şekil-E13.1: Zaman ve harcanan iş gücüne göre PMI proje yönetim süreçleri       | E13-6  |
| Şekil-E13.2: MSF/SDD, PMI ve IEEE standartlarına uygun proje ekibi yapılanması. | E13-8  |
| Şekil-E13.3: Uygulanabilecek yazılım geliştirme metodolojisi.                   | E13-12 |
| Şekil-E13.4: Ekip Modeli  | E13-13 |
| Şekil-E13.5.: Önaktif Risk Yönetim Süreci.                                      | E13-13 |
| Şekil-E13.6.: Uygulanacak yazılım geliştirme metodolojisi genel özellikleri     | E13-16 |
| Şekil-E13.7.: Risk Yönetimi Planlaması  | E13-23 |
| Şekil-E13.8: Risk Tanımlama   | E13-23 |
| Şekil-E13.9: Risk Niteleme/Niceleme Analizi                                     | E13-24 |
| Şekil-E13.10: Risk Tepki Planlaması   | E13-24 |
| Şekil-E13.11: Risk Gözleme ve Kontrol   | E13-25 |

## **EKLER LİSTESİ**

- EK-1 Tanımlar, Kısa Adlar ve Kısaltmalar
- EK-2 Yazılım Proje Geliştirme ve Yönetim Süreç, Aşama ve Adımları
- EK-3 Yazılım Proje Risk Yönetimi Adımları
- EK-4 Yazılım Projelerinde Doğru Eleman Seçimi ve Yazılım Proje Ekibinin Oluşumu ve Gelişimi Aşamaları
- EK-5 Yazılım Geliştirme Model, Metodoloji ve Teknikleri
- EK-6 Sistem İhtiyaç Özellikleri (SRS) Dokümanı
- EK-7 Yazılım Donanım İletişim Altyapısı Mimarisi
- EK-8 Genetik Algoritmalar ve Geliştirilen Sezgisel (Heuristic) Modeli
- EK-9 Tekrarlı Taşıma Planlama Modelleri
- EK-10 Kavramsal Fiziksel İlişkisel Veri Modeli
- EK-11 Sistem Veri Değişimi
- EK-12 Yazılım Geliştirme Planı
- EK-13 Yazılım Uygulama Planı
- EK-14 Eğitim Planı

## GİRİŞ

Proje yönetimi, iş idaresi ya da yönetim bilimleri başlığı altında da incelenen, insan gücü, sermaye, zaman, mekân, üretim süreçleri ve örgütlenme gibi birçok bileşenin de birlikte ele alındığı, planlama ve uygulamaya yönelik bir disiplindir. Proje yönetim teknikleri, belirli bir amaca ulaşmak için üretilecek mal ve hizmetlere ait girdilerin zaman ve maliyet etkinliği yönüyle en verimli şekilde kullanımını amaçlar.

"Proje yönetimi, planlama anlamında geçen yüzyılın sonlarından itibaren bilinmekle birlikte, bir iş süreci olarak II. Dünya Savaşı sırasında geliştirildiği kabul edilmektedir. Savaşlar, taşıdıkları kritik önemler açısından büyük projelerin başarıya ulaşabilmesi için çeşitli yönetim tekniklerinin geliştirilmesinde önemli bir rol oynamışlardır. Atom bombasının yapılması için oluşturulan çok büyük çaplı "Manhattan Projesi"nin başarısı için yapılan eşgüdüm çalışmaları ile bütçe, zamanlama ve kaynak yönetimi çabaları da bugünkü anlamda proje yönetimi tekniklerinin oluşturulmasını sağlamıştır.

Proje yönetimi 1950'li yıllardan itibaren gelişmekte olan birçok ülkede olduğu gibi ülkemizde de önem kazanmış, özellikle dış kaynaklı ve yabancı ortaklı büyük projelerin yürütülmesinde, proje yönetimi örgütleri oluşturulmuştur. Ülkemizdeki duruma bakıldığında, kamu kuruluşları da dâhil olmak üzere projeler için gerekli kaynakların seferber edildiğini ve uygun teknolojilerin seçildiğini söylemek mümkündür. Ayrıca ihtiyaç duyulan yetişmiş insan gücü son yıllarda gerek nitelik, gerek nicelik olarak yeterlidir. Ancak, projelere çağdaş bir yönetim anlayışı ile yaklaşılmadığında; gerekli kaynaklar tahsis edilmiş olsa bile, projelerde hedeflenen başarı yakalanamamakta, hatta başarılı olması beklenen birçok proje basarsızlıkla sonuçlanabilmektedir.

Gelişen sanal ortamın sağladığı faydalar 2000'lere doğru iş hayatında giderek artan oranda hissedilmeye başladığından çok sayıda bilişim projesi birbirini takiben hayata geçirilmeye başlamıştır. Bu tip projelerin iş ve trend gereği sınırlı işgücü ile aynı anda başlamaları ve sonuçta artan ücretler ve primler de dikkate alındığında bu durumdan firmalar ve müşterilerden çok proje çalışanlarının memnun kaldığı söylenebilir. (Bingi, Sharma ve Godla 1999:7) Bilişim (bilgi ve iletişim) teknolojilerinin



günlük hayatta payını arttırdığı bu dönemde ülkemizde de dünyaya paralel olarak bilişim teknolojileri ile ilgili büyük yatırımlar yapılmaya başlanmıştır (Örn. ERP (Kurumsal Kaynak Planlama) projeleri vb.). Bu yatırımlarla birlikte iş hayatında birçok görev tanımı ve yeni pozisyonlar türemiştir. Ülkemizdeki bilişim projelerinde, bu yeni görev tanımları yeni ekonominin doğduğu Amerikan metodolojilerinden alınmıştır. Ancak bilişim projeleri çoğu zaman yanlış görevlendirme ve eksik kaynaklardan dolayı hatalarla son bulmuştur.

Bu durum sürekli masraf merkezi haline dönen projelerde, işleri kolaylaştırmak yerine zorlaştırmış, hatta durma noktasına getirmiştir. Buna yetkin olmayan proje yöneticilerinin de eklenmesiyle bu alandaki proje yönetim bilgi ve tecrübe eksikliği uzayan daha kötüsü bitmeyen projelere neden olmuştur (Karakanian 1999:1, Kumar ve Hillegersberg 2000:23-26).

Son yıllarda kamu kuruluşlarınca sürdürülmekte olan bilişim projeleri de uygulama süreleri ve maliyetler yönüyle büyük projeler halini almıştır. Bunların başarıyla sonuçlandırılmamalarındaki en önemli neden, bütçe ve insan gücü kısıtlarından çok etkin bir proje yönetiminin sağlanamamış olmasıdır.

Günümüz iş dünyasında her şey süreç odaklıdır, hatta dinamik paralel süreçler kullanılmak zorundadır. Bu tip iş uygulamalarının elektronik hayata geçirilmesi teknolojik bir aşama olmadığından, bu tip projelere değişim projeleri olarak bakılmalı ve stratejik bir plan dâhilinde gerçekleşmesi sağlanmalıdır.

Bilişim projeleri genellikle geç sonuçlandırılma, bütçe aşımı ve istenilen hedefe ulaşamama konularında ünlü projelerdir. Sanayi ve inşaat sektöründeki proje deneyimlerinin yüzyılı aşkın olduğu göz önünde bulundurulduğunda, bilişim projelerinin kısa geçmişinde, proje yönetimine özgü araç ve tekniklerin çok fazla denenmediği, hatta oluşumlarını bile tam olarak tamamlayamadıkları söylenebilir. Bilişim projelerine bu gözle bakıldığında, başarı oranının yükselmesinin, Proje Yönetimi tekniklerinin iyileştirilmesine ve kullanımının yaygınlaştırılmasına bağlı olduğu görülecektir.

## **BİRİNCİ BÖLÜM**

### **PROJE VE PROJE YÖNETİMİ**

#### **1.1. PROJE VE PROJE YÖNETİMİ KAVRAMLARI**

##### **1.1.1. Proje Kavramı ve Özellikleri**

Belirli kaynaklarla belirli bir zaman içerisinde tamamlanması gereken ve tekrarlanmayan özel faaliyetler topluluğu olması yanında proje;

- Süreç kapsamında "Tasarım ve mühendislik uygulamaları gerektiren, belirli bir müşteri ihtiyacını ya da kullanıcı talebini karşılamak amacıyla üstlenilen ve bir ürün ya da hizmetin ortaya konulmasını konu alan sözleşme veya iş programı",
- Sonuç kapsamında "Emek, sermaye, zaman ve bilgi gibi, görünen ya da görünmeyen kaynakların kullanılmasıyla yaratılan ve müşteri ihtiyaçlarını karşılayan ürün veya hizmet" olarak tanımlanabilir.

Proje Yönetimi Enstitüsü (Project Management Institute-PMI) tarafından "Proje, özgün bir ürün veya hizmeti meydana getirmek için yürütülen geçici bir faaliyettir" (PMI, 2000:4) şeklinde yapılan tanım, projenin dünyaca kabul gören en geçerli tanımıdır. Kesin sınırlarla tanımlanmış belirli bir amaca hizmet eden, belirli bir süre içinde ve belirli bütçelerle sürdürülen ve insan, para, malzeme gibi kaynakları tüketen her faaliyet proje olarak tanımlanabilir (Kerzner, 1998:2). Tüm projeler başlama ve tamamlanma gibi süreçleri yaşadıklarından dolayı geçici, üretilecek hâsılanın öncekilerden veya (ilk kez yapıyorsa) diğerlerinden farklı olmasından dolayı da özgün yapıdadırlar. Gray ve Larson (2000) proje özelliklerini; amaç, süreli ömür, profesyonel katılımcılar, emsalsiz olma, özgün zaman, maliyet ve performans özelliklerine sahip olma şeklinde belirlemiştir.

Projeler, aşağıda yer alan belirleyici özellikleri sayesinde daha iyi tanımlanabilir (Barutçugil 2008:14):

- Projelerin açık tanımlanmış belirli amaçları vardır.
- Önceden belirlenmiş başlangıç ve bitiş noktaları vardır. Amaçların elde edildiği kesin bir son noktası bulunmaktadır.
- Olay, kendine özgü, tek ve tekrarlanmayan niteliktedir.

- Proje, belirli bir ürünü ortaya çıkarmak için zaman ve parasal kaynaklar kullanılarak yürütülen karmaşık bir çabadır. Birbirini izleyen ve paralel giden faaliyetlerden oluşan bir süreçtir.

- Bir proje çeşitli örgütsel yapıların kurulmasını ve değişik işlevsel ilişkilerin geliştirilmesini gerektirir,

- Bir proje ürünü, birçok bakımdan kendine özgüdür. Tek ve benzersizdir.

- Proje, çok çeşitli alanlardan, farklı niteliklerdeki kişi ve kuruluşların birbirini izleyen aşamalarda ortak çabalarını gerekli kılan karmaşık bir süreçtir.

- Proje uzun süreli bir çaba gerektirir. Endüstriyel birçok ürün, saatler, hatta dakikalarla ifade edilen süreçler içerisinde üretilirken bir projenin tanımlanması için aylar ve yıllar süren çabalar harcanır.

- Proje pahalı bir üründür. Birçok ürünle karşılaştırıldığında ölçeği ve nitelikleri nedeniyle proje ürününün ortaya çıkarılması büyük parasal kaynakların kullanımını gerektirir.

Yeni bir ürün (araç, yazılım, ekipman, vb.) geliştirilmesi, büyük ve karmaşık bir yapının (super-structure) inşa edilmesi, yeni bir askeri sistemin geliştirilmesi veya modernizasyonu, bir ulaştırma sisteminin kurulması, bir fabrikanın taşınması, bir kurum ya da organizasyonun yapısal değişikliğe gitmesi, bir ulaştırma altyapısı sinyalizasyon sisteminin değiştirilmesi gibi birçok faaliyet projeye örnek olarak verilebilir.

Projelerin başarıyla sonlandırılabilmesi için belirlenen süre ve bütçe dâhilinde ve % 100 müşteri/kullanıcı memnuniyetini sağlayacak nitelikte tamamlanması zorunludur. Kerzner (1998:6), günümüzde başarı için projenin; kapsamının sabitlenmesi veya kapsam bakımından küçük değişikliklerle devam edilmiş olması, organizasyonların ana iş akışını olumsuz etkilememesi, şirket kültürünü değiştirmemesi gibi hususlar da dikkatli davranılması gerektiğini ifade etmiştir. Projelerin, yukarıda açıklanan karmaşık ve kendine özgü yapıları da dikkate alındığında başarılı olabilmeleri etkin şekilde yönetilmelerine bağlıdır.

Karmaşıklıklarına karşın projeler; belirlenmiş hedeflere, öngörülen bir zaman içinde ve üzerinde uzlaşma sağlanmış bir bütçe dâhilinde ulaşmayı amaçlar. Proje başlatmak, yapısal organizasyon değişikliklerini ya da yeni organizasyon oluşumunu

zorunlu kılar. Her bir projenin içinde yer alan faaliyetler tekrarsız ve kendine özgü yapıdadır.

### **1.1.2. Proje Yönetimi Kavramı ve Özellikleri**

Proje yönetimi, kaynakların belirlenen proje amaçlarına ulaşılması için planlanması, teşkilatlanması, yönetilmesi ve kontrolüdür (Kerzner, 1998:4). Proje hedeflerini gerçekleştirmek için bilgi, beceri, teknik, araç, malzeme ve diğer imkânların etkin şekilde kullanılmasıdır.

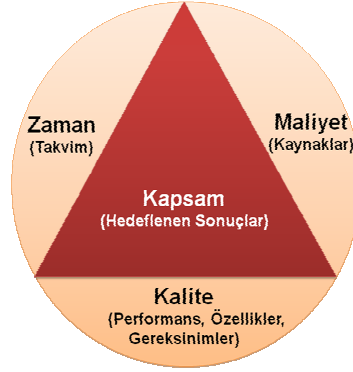
Proje yönetimi; üretim, ticaret ya da hizmet sektöründe çalışan bir organizasyonun ele aldığı bir projenin amaçlarını (belirli bir sürede, belirli bir bütçe ile öngörülen teknik sonuçlara ulaşmak) gerçekleştirebilmek için kullandığı parasal ve parasal olmayan kaynakların faaliyetlerini planlama, organizasyon, yönetme, koordinasyon ve kontrol fonksiyonlarının bütünüdür (Barutçugil 2008:15).

Tekrarlanmayan özel bir faaliyetin başlı başına bir proje olarak ele alınıp yönetilmesi olarak ifade edilen proje yönetimi yaklaşımına ilişkin sonuçları daha iyi anlayabilmek için aşağıdaki ilkelerin bilinmesi gereklidir (Barutçugil 2008:17):

- Proje yönetimi, belirlenmiş bir sonuca götüren araçtır. Kendisi bir sonuç ya da amaç değildir. Belirli bir amacı elde etmek için iletişim, çalışmaya özendirme/isteklendirme, koordinasyon, organizasyon ve görevlerin yerine getirilme biçimidir.
- Proje yönetimi, girişimcinin danışmanların, mühendislerin, tasarımcıların, yapımcıların, yüklenicilerin, sigortacıların ve diğerlerinin amaçlarını birleştiren bir süreci ifade eder. Bunların tümünün ortak amacı, projenin başarıyla tamamlanmasıdır.
- Proje yönetimi, elde bulunan ekonomik kaynakların ve sahip olunan bilgi ve becerinin en iyi biçimde kullanılmasını gerektiren bir çabadır. Sonuçta sağlanan kâr, çalışmayı özendirici bir güdü ve çalışma başarısının bir ölçüsü ve ödülüdür.
- Proje yönetimi, ne yeni riskler yaratan ne de mevcut riskleri üstlenen bir süreçtir.

Proje yönetimi başlangıç, planlama, yürütme, takip ve kontrol, sonlandırma süreçlerinden oluşur (PMI, 2000:4). Bu süreçler gerçekleştirilirken proje faaliyetlerinin koordinasyonu, yeterli ve doğru proje kapsam tanımı yapılması, proje

yönetim politikalarının belirlenmesi (süre, maliyet, kalite, risk, insan kaynakları, iletişim, tedarik) proje ömür devri boyunca etkin şekilde uygulanması gerekmektedir (Şekil-1.1).



**Şekil-1.1:** Proje Yönetimi Bilgi Alanları

Proje yönetimi, karmaşık faaliyetlerin bir plan ve sıra dâhilinde kontrollü olarak gerçekleştirilmesi ve proje organizasyonlarının etkinliğinin sağlanması amacıyla yönetim bilimine bir uzmanlık alanı olarak dâhil edilmiştir. Zaman içinde proje ölçeklerinin büyümesi, teknolojinin ilerlemesi, talebin çeşitlenmesi ve büyük sistemlerin kurulmasında etkinlik sağlama ihtiyacının ortaya çıkması sonucu hızlı bir gelişme göstermiştir.

Projelerde, proje faaliyetlerinin belli bir sıra ve düzen içinde gerçekleştirilmesi ve etkin kaynak kullanımı başarıyı getirmektedir. Bu çerçevede; etkin zaman, malzeme, insan, para, bilgi ve benzeri kaynak yönetimi maliyet, süre ve teknik başarı bağlamındaki proje hedeflerine ulaşılmasında önkoşul niteliği taşımaktadır.

Projelerin kendine özgü yapıları sonuçlarının net olarak tahminlenmesini önlemektedir. Özellikle, büyük ve yüksek teknoloji gerektiren projelerde belirsizlik zaman, maliyet, güvenlik ve teknik başarı konularında ortaya çıktığından önemli riskler söz konusu olmaktadır.

Planlama ve maliyet kontrolü ileri teknoloji ya da teknolojinin geliştirilmesi kapsamında gerçekleştirilen projelerde son derece zordur. Karmaşık yapılar, belirsiz durumlar ve engeller/zorluklar, farklı organizasyon yapılarının oluşturulmasını, farklı yetki - sorumluluk ilişkilerinin düzenlenmesini ve özel planlama - kontrol tekniklerinin kullanımını gerekli kılar.

Proje yönetimi, içinde genel yönetim esaslarını bulundurmakla birlikte olağan işletme faaliyetleri yönetiminden oldukça farklı yapıdadır. Özel organizasyon biçimlerinin teşkil edilmesini, farklı planlama ve kontrol yöntemlerinin kullanılmasını ve özel eğitim alınmasını gerektirir.

Projeler yürütülürken muhtemel problem ya da riskler göz ardı edilmeden, belirlenen hedeflere ulaşma konusunda sarf edilen tüm gayretler proje yönetimi içinde değerlendirilmektedir.

Büyük projelerde, görev alan proje paydaşlarının birlikte çalışma ve ortak amaç konusunda isteklendirilmesi, uzun zaman aralıklarının öngörülmesi, belirsizliklere ve muhtemel risklere karşı planlama yapılması ve çok büyük kaynakların tahsisi ve kullanımı söz konusu olmaktadır. Bu kapsamda, ürün tasarım planlaması, insan yönetimi, risk yönetimi, finansal yönetim, iletişim yönetimi, konfigürasyon yönetimi gibi özel yönetim teknik ve uygulamalarının muhtemel sorunların önlenmesi bakımından bir arada ve etkinlikle kullanılması zorunluluğu ortaya çıkmaktadır. Proje yönetim tekniklerinin büyük projelerde yeterince etkin bir biçimde kullanılmaması maliyet artışı, projede zaman aşımı, yanlış kaynak kullanımı/kaynak aşımı ve teknik hedeflere ulaşılamaması gibi sorunları beraberinde getirmektedir.

Projelerin etkin şekilde yönetilebilmesi, örgütsel yapı ve davranışlar ile nicel araç ve tekniklerin en üst seviyede anlaşılması/kavranması ile mümkündür. Önem dereceleri proje ve sanayi bazında farklılık göstermekle birlikte proje yönetimi prensipleri, evrensel nitelikte olup her çeşit projeye ve her tür sanayiye uygulanabilmektedir. Sanayi/proje türlerine göre proje özellikleri karşılaştırması Tablo-1.1 'de sunulmuştur.

**Tablo-1.1:** Proje Karakteristiklerinin Projelere Göre Sınıflandırılması

| Proje Sanayi Türü                        | ARGE  | Küçük Çaplı İnşa | Büyük Çaplı İnşa | Havacılık/Savunma | Yönetim Bilişim Sistemleri | Mühendislik |
|--|-------|------------------|------------------|-------------------|----------------------------|-------------|
| Kişiler Arası İletişim Yeteneği İhtiyacı | Düşük | Düşük            | Yüksek           | Yüksek            | Yüksek                     | Düşük       |
| Organizasyon Yapısının Önemi             | Düşük | Düşük            | Düşük            | Düşük             | Yüksek                     | Düşük       |

| Proje Sanayi Türü              | ARGE                | Küçük Çaplı İnşa   | Büyük Çaplı İnşa   | Havacılık/ Savunma | Yönetim Bilişim Sistemleri | Mühendislik         |
|--------------------------------|---------------------|--------------------|--------------------|--------------------|----------------------------|---------------------|
| Zaman Yönetimi Zorlukları      | Düşük               | Düşük              | Yüksek             | Yüksek             | Yüksek                     | Düşük               |
| Toplantı Miktarı               | Çok Fazla           | Düşük              | Çok Fazla          | Çok Fazla          | Yüksek                     | Orta                |
| Proje Yöneticisinin Seviyesi   | Orta Seviye Yönetim | Üst Seviye Yönetim | Üst Seviye Yönetim | Üst Seviye Yönetim | Orta Seviye Yönetim        | Orta Seviye Yönetim |
| Çatışma Yoğunluğu              | Düşük               | Düşük              | Yüksek             | Yüksek             | Yüksek                     | Düşük               |
| Maliyet Kontrol Seviyesi       | Düşük               | Düşük              | Yüksek             | Yüksek             | Düşük                      | Düşük               |
| Proje Destekleyicisi İhtiyacı  | Var                 | Yok                | Var                | Var                | Yok                        | Yok                 |
| Planlama/ Çizelgeleme Seviyesi | Sadece Km. Taşları  | Sadece Km. Taşları | Detaylı Plan       | Detaylı Plan       | Sadece Km. Taşları         | Sadece Km. Taşları  |

(Kerzner, 1998:43)

Yüksek proje maliyetlerinden dolayı proje merkezli çalışan alanlarda (savunma/havacılık, büyük çaplı inşa, vb.) proje yönetimi yaklaşımı daha da önem kazanmaktadır. Proje merkezli olmayan alanlarda proje yönetimi daha standart dışı ve gelişmiş güzel gerçekleştirilebilmektedir.

### 1.1.3. Proje Ömür Devri

Proje ömür devri, projelerin özgün yapısını tanımlamanın bir başka yoludur. Ömür devri ile projelerin başlangıcından bitişine kadar olan safhalar tanımlanır. Bu safhaların her birinde farklı yönetim ve işletim süreçleri bulunduğundan, her bir safhada farklı faaliyetlerin ve bu faaliyetlerin gerçekleştirilmesini sağlayacak farklı becerilerin kullanılması zorunluluğu ortaya çıkmaktadır. Bunu kullanmak kimi yöneticilere göre proje yönetiminin temel unsurudur. Proje Ömür Devri sayesinde, projelerin sınırlı bir sürede bitirilmelerinin gerektiği ve bu süre içinde projeye sarf edilen gayret ve odaklanmaların öngörülebilir değişiklikler yaratabileceği projeye ilgilenelelere sürekli olarak hatırlatılmış olur.

Bir projenin ömür devri üç evreden oluşur. Birinci evrede proje fikri doğar ve geliştirilir. İkinci evrede proje gerçekleştirilir. Üçüncü evrede ise proje bitirilerek sonuçları hayata geçirilir ve projeyi gerçekleştirmek için sağlanan kaynaklar (insan

gücü, takım vb.) dağıtılır. Proje Yönetimi, ağırlıklı olarak ikinci evrede rol oynar. (Şekil-1.2).



Şekil-1.2: Proje Yaşam Döngüsü (TBD:25)

Ömür devrinin safhalarını belirleyen kıstaslar projenin yapısına ve amacına göre farklılık göstermektedir. Projelerin türüne ve uygulandığı sektöre göre değişiklik göstermekle birlikte, proje yönetimi kapsamında geliştirilmiş çok sayıda ömür devri modeli örneği bulunmaktadır.

Genel bir yazılım geliştirme projesinin ömür devri modeli; tanımlama, tasarım, kodlama, bütünleştirme/test etme ve idame olmak üzere beş safhadan meydana gelmektedir (Gray ve Larson, 2000:5-6). ABD Savunma Bakanlığı (DoD) projeleri için Kavram (Konsept) ve teknoloji geliştirme, Sistem Geliştirme ve Sergileme, Üretim ve Plana Göre Yerleştirme ve Destek olmak üzere dört safha tanımlanmıştır (PMI, 2000:14). Morris (1988:21) ise inşaat projelerinin safhalarını, "Yapılabilirlik, Planlama ve Tasarım, Üretim, Sermaye Devri ve Başlangıç" şeklinde belirlemiştir. Proje ömür devri safhaları genel manada "Kavrama, Geliştirme, Uygulama ve Sonlandırma" şeklinde sıralanabilir.

Her bir proje yönetimi safhası yapısal özelliklerine bağlı olarak alt aşamalara bölünebilmekte, uygulanacakları proje türü ve sanayiye göre sayıları artabilmekte ve içerikleri geliştirilebilmektedir. Her safhanın hedeflerini açıkça ortaya koyan ve bu hedefler kapsamında etkin kaynak kullanımını sağlayan yetkin bir yönetim anlayışının uygulanması önem arz etmektedir. Her safhada karşılaşılan olumsuzlukların tüm projeyi etkileyeceği kaçınılmaz bir gerçektir.

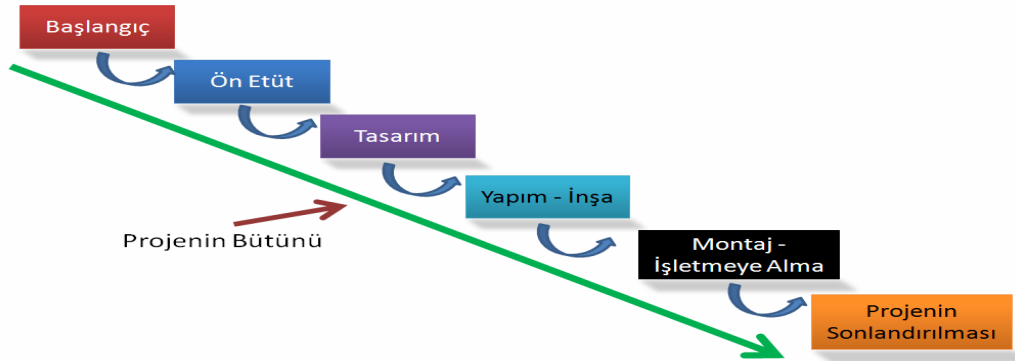
Her projenin bir ömür devri vardır. Proje, açıkça belirlenmiş başlangıç ve bitiş tarihleri arasında bir zaman ölçeği üzerinde gösterilebilecek bir ömre sahiptir. Ömür devri fikrin ortaya konulmasından projenin tamamlanmasına kadar geçen tüm aşamaları kapsar. Proje ilerledikçe aralarındaki başlangıç ve bitiş noktaları genellikle



belirgin bir biçimde görülmeyen ve yalnızca geçici kabul ve devam için resmi yetki verilmesinin aşamaları birbirinden ayırdığı durumlarda açıkça görülebilen çeşitli aşamalardan geçilir.

Ömür devrinin birbirini izleyen her aşamasında yeni ve farklı bir ürün yaratıldığından ve bir aşamanın ürünü bir sonraki aşamanın girdisini oluşturduğundan proje özellikleri peyderpey değişim gösterir (Şekil-1.3).

Proje ömür devri çerçevesinde ilerleme kaydedildikçe maliyet ve bitiş zamanı gitgide netleşmeye, belirsizlikler ve karanlık hususlar aydınlanmaya başlar. Hedeflenen sonuç, sonuç kapsamında hedeflenen bitirme zamanı ve maliyet birbirleriyle yakın ilişkilidir. Başlangıçta oldukça büyük olan belirsizlik alanı, ardışık her aşamada biraz daha azalmakta ve ancak proje sonuçlandığında tamamen ortadan kalkmaktadır. Sonuçları erken ve doğru bir şekilde öngörebilecek etkin ve etkili bir proje yönetimine olan ihtiyaç projenin tamamlanacağı zamana ve toplam maliyetine ilişkin belirsizlikten kaynaklanmaktadır.



**Şekil-1.3:** Proje Ömür Devri Aşamaları (Barutçugil, 2008:20)

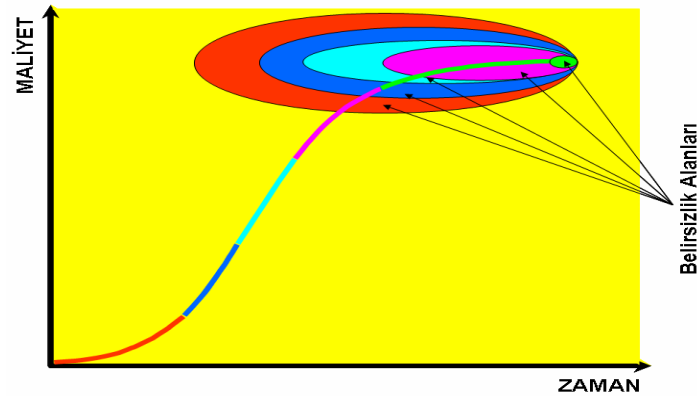
Proje ömür devri dâhilinde herhangi bir aşamanın ya da tüm projenin hızlandırılmasının getireceği maliyet, aşamanın/aşamaların özelliğine de bağlı olarak sonuca yaklaştıkça giderek artar. Bu noktadan hareketle, projede gecikmelerin önlenmesi ve zamanın kısaltılması ilk aşamalarda daha kolay ve daha masrafsız olacağı ifade edilebilir. Kaybedilen zamanı kazanma girişimleri başlangıçtan bitişe doğru gidildikçe artan kümülatif bir maliyet getirmektedir. Bunun önlenmesi de her aşamada uygulanacak etkin bir kontrol sistemiyle mümkündür.

Buraya kadar belirtilen özellikler proje yönetimi kapsamında;

- Projelerin ömür devri esasına göre yönetilmeleri zorunluluğunu,
- Her proje için ayrı, kendi başına, yeni bir organizasyon yapısı oluşturmak yerine organizasyon yapılarının muhtemel değişikliklere kolayca uyum sağlayabilecek esneklikte tesis edilmesi gerekliliğini,
- Başlangıçtaki kararların nihai zaman ve maliyet üzerindeki etkileri, sonraki aşamalarda alınan kararlara göre çok daha büyük olduğundan projenin ilk aşamalarında kararlar alınırken yöneticilerin çok daha dikkatli olmaları gerektiğini,
- Sorumlulukların proje süresince devam ettiğini,
- Başlangıçtan bitime bütünleşik bir yönetim uygulanması gerektiğini vurgulamaktadır.

#### 1.1.4. Projelerin Aşamaları

Projelerin çoğunlukla karmaşık yapıdaki aşama işlem nitelik ve ölçekleri sürekli değişim gösterir.(Şekil-1.4)



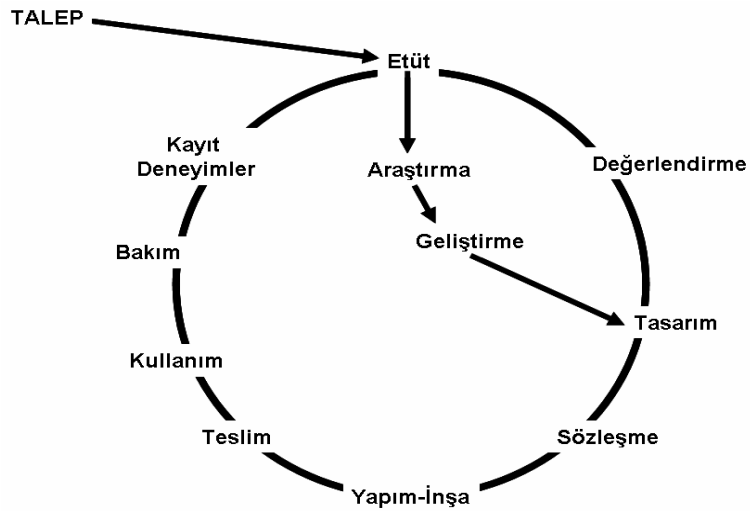
**Şekil-1.4:** Proje aşamaları boyunca zaman ve maliyet belirsizliklerinin değişimi

Proje ömür devri (Şekil-1.5), yeni bir ürün ya da hizmet için bir talebin olması ya da geçmiş projelerden devreden kayıtlar ve deneyimler ile yeni fırsatları dikte eden araştırma sonuçları vb. sayesinde muhtemel bir pazar talebinin tahmini ile başlar. Özellik ve kaynakları açısından farklı ve son derece önemli olan bu tahmin girdileri, önerinin içerdiği yenilik derecesine ve tasarımda ne kadar yenilik veya uygulama gerektireceğine bağlı olarak yeni bir projenin etüt aşamasında bir araya getirilirler.

Proje ömür devrinin müteakip aşamasında girdilerin değerlendirilmesi ve tahminlenen maliyet ve beklenen faydaların kıyaslaması yapılır. Projenin aciliyeti veya önemi söz konusu olduğunda, bu aşamada ihtiyaç duyulan hesaplamalar her halükârda yapılmasına rağmen dikkate alınmayabilir. Alternatif projelerin varlığı, hangi projenin amaçları daha iyi gerçekleştirebileceğini belirleyecek değerlendirme aşamasının önemini artırır. Tahminlere girdi sağlayan ve projenin incelenmesi, araştırılması için ayrılan kaynakların büyüklüğünden etkilenen bilgi niteliklerinin güvenilirliği de göz önüne alınarak, tahmine dayalı hesaplama sonuçlarının kesinliğinden söz edilmesi mümkün olmaz.

Bu aşamanın sonunda projenin kabul veya reddine ilişkin kesin bir karar verilir. Devam kararı alındığında gerçek proje maliyet ve faydalarını belirleyen tasarım ve yeni problem sahalarının tespit edildiği, belirsizliklere ilişkin tahminlerin yeniden değerlendirildiği geliştirme aşamalarına geçilir.

Sözleşme aşaması, izleyen aşamalara geçiş için yetkilendirmenin sağlandığı aşamadır. Yükleniciler ile sözleşmelerin yapımı ve işletme içinde işgücü, malzeme ve fon kullanımına yetki veren iş emirleri sisteminin kullanımı bu aşamada gerçekleştirilir.



**Şekil-1.5:** Proje Ömür Devri (Barutçugil, 2008:24)

İzleyen aşama faaliyetlerin en büyük ölçeğe ulaştığı inşa (yapım - gerçekleştirme) aşamasıdır. Fiziksel gayretlerin belli bir olgunluk seviyesine ulaşması, faaliyetlerin çeşitlilik kazanması ve karmaşık ilişki ve bağımlılıkların

artması daha çok bu aşamada gerçekleşir, olgunlaşır, faaliyetler geniş bir çeşitlilik kazanır ve karmaşık ilişkiler artar.

İzleyen aşamada; çalışmaların sonuçlandırılması, elde edilen ürün veya hizmetin teslimi ve kullanıma sunumu gerçekleştirilerek proje döngüsünde öngörülen hedeflere ulaşılmış olur.

Son olarak, genel bir faaliyet gözden geçirmesi ve değerlendirme, elde edilen birikimin kayıtlanması ve ortaya çıkabilecek teknik veya yönetsel sorunların izlenmesi gerekir.

#### **1.1.5. Proje Yönetimine Olan İhtiyaç ve Proje Yönetiminin Sağlayacağı Faydalar**

Proje yönetimi kapsamında, yeni organizasyon yapıları ve her bir projeye özel ve karmaşık yönetim tekniklerinin geliştirilmesi ve her bir projeden yeni yönetim tecrübeleri elde edilmesinin altında, geleneksel yapı ve tekniklerin proje sorunlarını çözmede yetersiz kalması yatmaktadır.

Proje yönetiminde, farklı yapılanmalara, uzman yönetim bilgi sistemlerine, proje yönetim tekniklerine olduğu kadar, problem çözme konusunda bilgili, becerili ve yetkin yöneticilere de ihtiyaç duyulmaktadır.

Projeler, geçici faaliyetlerdir ve daha uzun süreli istisna araştırma geliştirme projeleri mevcut olsa da ömürleri genellikle altı ay ile beş yıl arasındadır. Her bir proje için üstlenilen görevin etkinlikle icrası maksadıyla yeniden teşkilatlandırıldığı için yönetim, organizasyon ve bilgi sistemlerine yönelik yapılanmalar proje bazında farklılık gösterir. Bunun doğal sonucu olarak da yüklenici işletmeler ya da bunların alt bölümlerinin, birden fazla proje üzerinde çalışıyor olabilmeleri ve her biri de projelerin yaşam dönemlerinin farklı bir aşamasında bulunabilirler. Projede yer alan işletmeler veya bölümler arası sürekli değişen ve gelişen bağımlılıklar bu çerçevede dikkate alındığında oldukça karmaşık bir iş bölümü ve çalışma sistemi yanında, farklı meslekler ve değişik faaliyet alan/bölüm/işletmeleri tarafından görevlendirilmiş personelden oluşan karmaşık bir yapının söz konusu olduğu kolaylıkla görülebilir. Bir projeye katkıda bulunan işlevsel bölümlerin ya da işletmelerin hiçbirinin projenin ömür devrinin bütünü üzerinde en önemli olamayacaklarını gösteriri şekilde; proje başlangıcında ağırlık "etüt-araştırma"da iken, daha sonra sırasıyla tasarım, tedarik ve oradan da inşaya (gerçekleştirme) ve deneme, teslim ve kullanıma sunma ile doğrudan sorumlu bölüme/işletmeye geçer. Bu noktadan bakıldığında, hiçbir

bölüm/işletme yöneticisinin tek başına projenin bütünü için yürütme görevini üstlenemeyeceği söylenebilir.

Projelerin kendine özgü yapıları; iş tanımı, teşkilatlanma, yetkilendirme, planlama, bütçeleme ve kontrolde iletişim ve koordinasyonun sağlanması ve eşzamanlı çalışılabilmesi bakımından önemli sorunlar yaşanmasına neden olabilir. Yönetim işlevleri eksiksiz olarak gerçekleştirildiğinde çözüme kavuşmayacak sorun ve amacına ulaşmayacak proje olmayacaktır.

Proje yapılanmalarının yönetim işlevlerini ihmallerinin ve yönetimi önemsememelerinin altında yatan bazı nedenler özetle aşağıda sunulmuştur (Barutçugil 2008:25):

- Proje organizasyonları genellikle gerekenden az eleman çalıştırmaları, işin fiilen yürütülmesine yönelik teknik fonksiyonlara ağırlık verme eğiliminde olmaları ve önceliklerini buna göre tespit etmeleri nedeniyle, planlama, organizasyon, yöneltme, koordinasyon ve kontrol gibi yönetim işlevlerine gereken önemi vermeyebilirler.

- Proje sonunda üretilmesi beklenen hedef ürün ya da hizmetin kendine özgü yapısı, geleneksel süreçler kullanan organizasyonların bildik yönetim uygulamalarındaki güvenilirliğine ve yeterliliğine darbe vurmaktadır. Çoğu proje organizasyonu tarafından yeterince anlaşılamasa da kârlılık ve başarı, proje organizasyonunun yapısal özelliklerine göre tasarlanmış ve geliştirilmiş özgün bir yönetim sistemiyle sağlanabilir.

- Bazı durumlarda, küçük ölçekli proje organizasyonları, çoğu kez bir firma da uzman ya da yönetici olarak çalışırken parlak bir fikir bilgi beceri ve çabaları sonucu mülkiyete ve yöneticiliğe ulaşmış girişimciler altında faaliyet gösterirler. Bu girişimcilerin yönetimin önemini anlamalarının geç ve güç olmasının nedeni, yönetimi baştan itibaren kendileri için öncelikli ve gerekli görmemeleridir.

Ancak günümüzde ortaya çıkan yeni koşullar, proje organizasyonlarının yönetime giderek daha fazla önem vermelerini zorunlu kılmakta, proje organizasyonlarının ölçeği büyümekte ve projelerin başlatılmasından tamamlanmasına kadar geçen süre ve parasal proje kaynakları da giderek artmaktadır. Zaman ve kaynakların etkin ve esnek kullanımı giderek azalmakta ve daha fazla uzmanlaşmış işgücüne ihtiyaç duyulmaktadır. Bu bağlamda, teknolojik ve

ticari açıdan yoğun rekabet yaşanan pazarlarda başarılı olabilmek için etkili proje yönetimi teknik ve uygulamalarına olan ihtiyaç da artmaktadır.

Birçok projede, yetersiz ve uygunsuz yönetim sistemleri nedeniyle planlanan tamamlanma süresi ve bütçe büyük oranda aşılmaktadır. Proje çalışmalarında etkin iletişim, koordinasyon ve kontrolün sağlanması için geleneksel yönetim - organizasyondaki bilgi sistemlerinden farklı ve projeye özgü bilgi sistemlerine ihtiyaç vardır. Büyüyen projeler yanında, artan katkı sağlayıcı işletme/bölüm sayısı ve karmaşılaşan faaliyetler iletişim ve koordinasyonun önemini artırmaktadır.

Proje içindeki bağıl ilişkisel ve karmaşık yönetimsel yapılanmanın, projeye katkı sağlayan ve farklı amaç ve yöntemler benimsemiş farklı birim ve şahıslar arasında sorun yaratması muhtemeldir. Bu şahıslar ve teşkil ettikleri gruplar arasında sürekli bir çatışma eğilimi bulunması nedeniyle, geleneksel yönetim-organizasyon teknik ve uygulamalarının proje yönetimini destekleyecek şekilde değiştirilmesi ve geliştirilmesi gerekmektedir.

Proje yönetimi kavramı, dar manada projeye katılan farklı grupların işlerinin koordinasyonunu, geniş anlamda ise, proje faaliyetlerine katkıda bulunan tüm grupların etkin işleyen bir organizasyon içinde bütünleştirilmesini ifade ettiğinden, günümüzde proje yönetimi yönetim biliminde ayrı ve önemli bir yere sahiptir.

Proje yönetimi, tüm çabaların yalnızca ulaşılabilir teknik, zaman, maliyet ve güvenlik amaçlarına yönltilmesini ve her projenin özgün yapısına göre yönetimini ve bu sayede hedeflere daha kolay ulaşılmasını sağladığından geleneksel yönetimden farklı özellik ve nitelikler taşımaktadır.

Bütünleştirici bir sorumluluk elemanının tanımlanması (proje yöneticisi atanması); projenin bütün sonuçlarından tek bir kişinin sorumlu olması, kararların projenin genelinin başarısı için alınması, proje paydaşları arasındaki koordinasyonun kolaylaşması ve bütünleşik yönetim sistemi ve kazandırdığı bilgilerin daha kolay ve doğru kullanılması bakımından son derece önemlidir. Bütünleşik yönetimin proje yönetiminde sağlayacağı faydalar aşağıda sunulmuştur:

- Proje paydaşlarının faaliyetlerinin projenin gerçek ihtiyaçlarını karşılayacak şekilde planlanmasını ve yürütülmesini güvence altına alır.
- Projedeki belirli bir işe diğerlerine göre daha fazla önem vermenin etkilerinin ve sonuçlarının doğru bilinmesini sağlar.

- Projenin başarıyla tamamlanmasını engelleyebilecek sorunları erken fark etme ve bu sorunları önleyici ya da çözücü etkin eylemi zamanında gerçekleştirme olanağı sağlar.

Tüm bu yararlarına ve sağladığı üstünlüklere karşın etkin bir proje yönetimi uygulaması projenin tipine, büyüklüğüne, coğrafi konumuna, teknik niteliklerine ve ana işletmenin özelliklerine bağlı olarak büyük değişiklikler gösterebilen ve bir projenin yüzde biri ile beşi arasında değişen ilave bir maliyet getirmektedir. Bu kapsamda bir projenin yönetiminde ortaya çıkan ancak maliyete duyarlı bir yönetim anlayışı ve etkin maliyet kontrol teknikleri ile makul seviyelere çekilebilen maliyet kalemlerinden bazıları önem sırasına göre;

- Proje yöneticisinin ve varsa yardımcı kadrosunun maaş ve ücretleri, sosyal hakları ve seyahat masrafları,
- Merkezi planlama ve kontrol süreçlerinin ve sözleşmelerin yönetiminin neden olduğu maliyetler,
- İletişim ve bilgi teknolojileri ile ilgili yazılım / donanım giderleri şeklinde sıralanabilir.

## **1.2. PROJE ÖRGÜT YAPILARI**

Projeler; şirketler, uluslararası kuruluşlar, devlet kurumları vb. gibi kendilerinden büyük yapılanmaların bir parçasıdır. Her yapılanmanın kendine özgü bir kültürü ve bu kültürü oluşturan değerleri, beklentileri ve yaşam koşulları vardır. Sahip olunan yapılanma kültürü doğrudan projeye yansımaktadır. Proje, kendisi bir örgüt olduğu durumlar da bile (ortak girişim, işbirliği) kendi yapısından ya da projeyi gerçekleştiren organizasyon yapısından etkilenmektedir. Organizasyonların projeler üzerine etkisi Tablo-1.2'de sunulmuştur. Örneğin; sürekli yatay yapıda çalışmış ve yüksek katılımcı bir yönetim yaklaşımını benimsemiş bir proje yöneticisinin hiyerarşik yapıda kurgulanmış bir yapılanma içinde sorunlarla karşılaşması ya da bunun tersi bir durumda problem yaşanması muhtemeldir (PMI, 2000:18).

Proje örgüt yapıları tercih edilirken örgüt kültürü göz ardı edilmemelidir. Bu bölümde organizasyonların proje yönetmek için nasıl yapılandığı ve bu yapılanmalarının projeleri nasıl etkilediği incelenecektir. Proje organizasyon yapıları olan işlevsel örgütlenme, matris örgütlenme (zayıf, dengeli, güçlü) ve proje tabanlı örgütlenme türlerinin özellikleri, kullanıldıkları proje yapısına göre fayda ve sakıncaları açıklanacaktır.

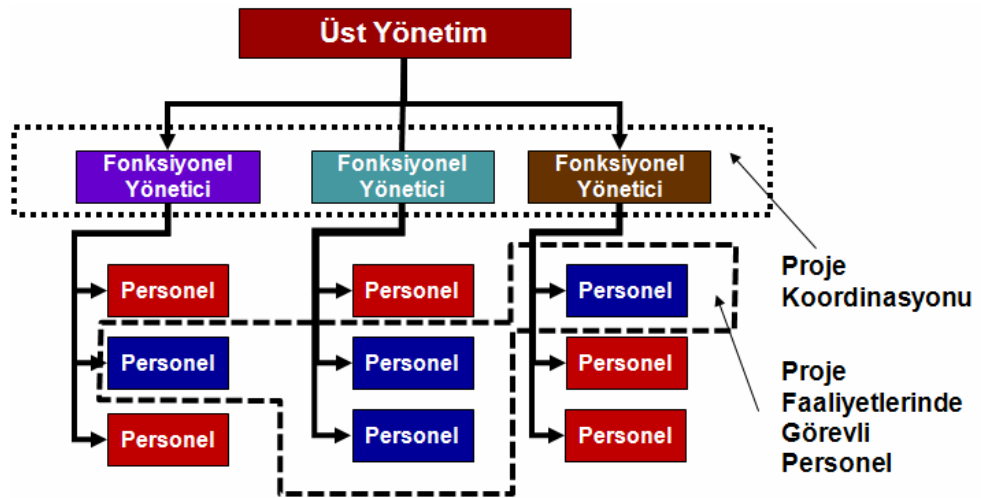
**Tablo-1.2: Teşkilat Yapısının Projelere Etkisi**

| Etki Alanı                                       | Teşkilat Yapısı | İşlevsel                        | Matris                          |                                  |                                     | Proje Tabanlı                       |
|--|-----------------|---------------------------------|---------------------------------|----------------------------------|-------------------------------------|-------------------------------------|
|  |                 |                                 | Zayıf                           | Dengeli                          | Güçlü                               |                                     |
| Proje Yöneticisinin Yetkisi                      |                 | Çok Küçük veya Yok              | Sınırlı                         | Az                               | Orta                                | Yüksek                              |
| Mesaisinin Tamamını Projeye Ayıran Personel % si |                 | Yok                             | 0-%25                           | %15-60                           | %50-95                              | %85-100                             |
| Proje Yöneticisinin Rolü                         |                 | Yarı Zamanlı                    | Yarı Zamanlı                    | Tam Zamanlı                      | Tam Zamanlı                         | Tam Zamanlı                         |
| Proje Yöneticisinin Ünvanı                       |                 | Proje Koordinatörü/Proje Lideri | Proje Koordinatörü/Proje Lideri | Proje Yöneticisi/Proje Sorumlusu | Proje Yöneticisi/Program Yöneticisi | Proje Yöneticisi/Program Yöneticisi |
| Proje Yönetiminde İdari Kadro                    |                 | Yarı Zamanlı                    | Yarı Zamanlı                    | Yarı Zamanlı                     | Tam Zamanlı                         | Tam Zamanlı                         |

(PMI, 2000:19)

### 1.2.1. İşlevsel Örgütlenme

Bir organizasyonda hâlihazırda var olan işlevsel hiyerarşik yapıyı kullanarak proje yönetme aslında bir proje örgütlenmesi yaklaşım tarzıdır. (Şekil-1.6). Bir proje; gerçekleştirilme onayı alındıktan sonra, organizasyon içinde yer alan işlevsel birimler bazında kendi sorumluluk sahalarına göre paylaşılır. Organizasyonlar içinde her bir birim genellikle birden fazla projenin sadece kendi sorumluluklarına düşen kısmı ile ilgilenirken, koordinasyon normal yönetim kanalları vasıtasıyla gerçekleşmektedir (Gray ve Larson, 2000:222).



**Şekil-1.6: İşlevsel Örgütlenme (PMI, 2000:20).**



İşlevsel örgütlenme, proje ana sorumluluğunun bir işlevsel birime verildiği durumlarda da kullanılmaktadır. Koordine sorumluluğu, bu durumda projede baskın olan birimden bir üst yöneticiye verilir. Örneğin; yeni ürün geliştirme ile ilgili bir projede koordinasyon sorumluluğu genellikle Tasarım ve Mühendislik birimlerine verilir.

Proje yönetiminde işlevsel örgütlenmeye gidilmesinin fayda ve sakıncaları mevcuttur. Faydaları (Gray ve Larson, 2000:222);

- Projeler organizasyonun ana teşkilatı içinde tamamlandığından, yapısında kökten bir değişiklik söz konusu olmaz.

- Personel kullanımındaki yüksek esneklik sayesinde proje kapsamında farklı bölümlerdeki uzman personel geçici görevlendirmeye veya aslî görevi uhdesinde kalmak üzere ikiz/üçüz/dördüz... gibi görevlendirmelerle kendi uzmanlık sahasına giren husus(lar) tamamlanıncaya kadar projeye/projelere dâhil edilebilir.

- Dar kapsamlı projelerde sorumluluk, projede faaliyetleri en yoğun olan birime verildiğinde projenin kritik öneme haiz bölümlerine istenilen uzmanlık ayrıntısında çözüm getirilebilir.

- İşlevsel birimlerde çalışan proje görevlilerinin, üstlendikleri her bir proje görevi kendilerini geliştirmeleri, tecrübe kazanmaları ve kariyerlerinde ilerlemeleri anlamında büyük katkı sağlar.

Sakıncaları ise;

- Projede üzerinde çalışılan konuya odaklanma, öncelik her birimin kendi devam eden rutin işlerine ve birim içindeki sorumluluklara verildiğinden zorlaşır ve proje sorumluluğu genelde ikinci plana itilir.

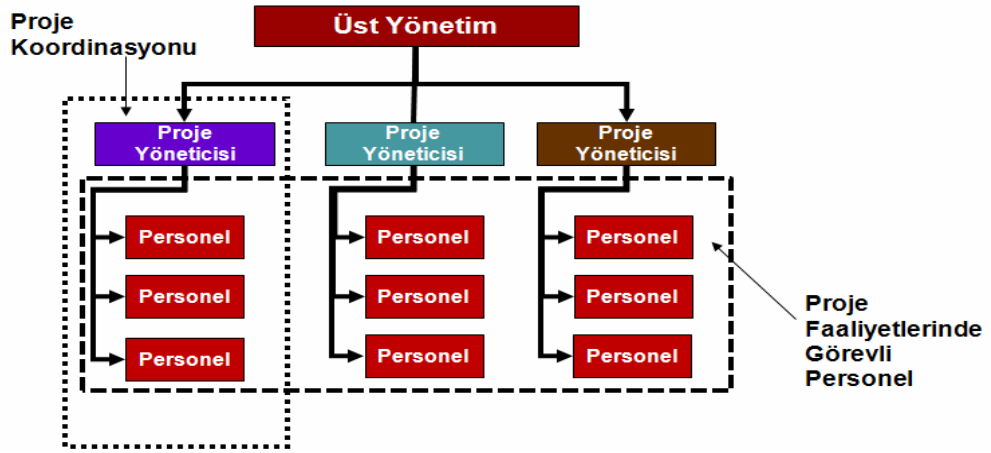
- Birimler arasında proje kapsamında icra edilen faaliyetlerin koordinasyonu ve ekip ruhu içinde bütünleşme zayıflayabilir.

- Projeye ilişkin bilgi akışında normal yönetim kanalları kullanıldığından projelerin tamamlanması daha uzun zaman alabilir.

- Projeye atanan personelin isteklendirmesi (motivasyonu) zayıf olabilir. Proje ek bir yük gibi algılanabilir. Ayrıca personel, projenin bir parçası ile ilgilendiği için benimseyemez ve sahiplenemez.

## 1.2.2. Proje Esaslı Örgütlenme

Proje esaslı örgütlenme, faaliyetlerinin büyük çoğunluğu projeler üzerinden gerçekleşen organizasyonlarda uygulanmaktadır. Proje esaslı teşkilatlanmada bağımsız proje takımları oluşturulur (Şekil-1.7). Proje elemanları çoğunlukla bir aradadır. Proje yöneticisi tam zamanlı görev alır ve büyük oranda yetki ve serbestiye sahiptir. Proje takımı ana teşkilat yapısından fiziksel olarak ayrıdır, öncelikle proje amaçlarını gerçekleştirmekle görevlendirilir. Her proje elemanı sadece bir kişiye, proje yöneticisine rapor vermektedir (Kerzner, 1998:108).



Şekil-1-7: Proje Esaslı Örgütlenme (PMI, 2000:21).

Proje esaslı örgütlenme;

- Projeden örgütün devam eden faaliyetleri doğrudan etkilenmediğinden, proje üzerinde tek, yetkin ve etkili bir otorite kurulabilmektedir.
- Güçlü iletişim kanalları mevcuttur ve takım içi tepki süreleri oldukça kısadır.
- Ortak bir amaç ve yaygın bir sorumluluk ağıyla birbirlerine kaynaşan proje ekibinin projeyi sahiplenmesi, isteklendirilmesi (motivation), morali ve uyumu daha kolay olur.
- Personel, başka yapılması gereken işleri, görevleri olmadığı için tüm dikkatini projeye yoğunlaştırır.
- Projeye uygun kaynaklar atandığı zaman yüksek seviyede çapraz işlevsel bütünleşme gerçekleşir.

- Farklı bölümlerden gelen uzmanlar birlikte yakın bir şekilde çalışırlar ve doğru bir yönlendirme ile kendi uzmanlık alanlarını geliştirmeyi değil projeyi en iyi sonuçla tamamlamayı hedeflerler.

Yukarıda açıklanan faydalar da göz önüne alındığında, çoğunlukla projelerin tamamlanmasında için en iyi yaklaşımın proje esaslı teşkilatlanma olduğu düşünülebilir. Ancak;

- Her proje için ayrı teşkilatlanmaya gitme yüksek maliyet getirmektedir.
- Farklı projelerde aynı faaliyetler için farklı ekiplerin kurulması, gereksiz tekrarlara (duplication) sebebiyet verir ve verimsizlikle sonuçlanabilir.

- Proje ekip personeli proje tamamlandıktan sonra eski bölümlerine döndükleri zaman, görev alanlarındaki yeni gelişmeleri takip etmeleri ve eski görevlerine alışmalarında problemlerle karşılaşmaktadır.

- Asıl görev yerindeki ekip arkadaşları proje tamamlanmasını müteakip aslı görevine geri dönen personeli tekrar içlerine almayarak dışlayabilmektedir.

- Personel için kariyer devamlılığı söz konusu değildir.

- Projeler arasında teknik paylaşım fırsatı azdır (Gray ve Larson, 2000:227; Kerzner, 1998:109).

### **1.2.3. Matris Örgütlenme**

Matris örgütlenme için çok çeşitli uygulamaları ve şekilleri sebebiyle tam ve kesin bir tanım yapmak oldukça zordur. Çoğunlukla, matris teşkilat yapısı, şirketler nasıl olmasını istiyorsa veya araştırmacılar çalışmalarının amacına göre nasıl uygun buluyorlarsa öyle belirlenmektedir (Ford ve Randolph, 1992). Matris yapı, dikey hiyerarşik yapı ile yatay yönetim birimlerinin sorumluluk ve yetkilerini paylaştıkları melez bir örgüt yapısı olarak ifade edilebilir.

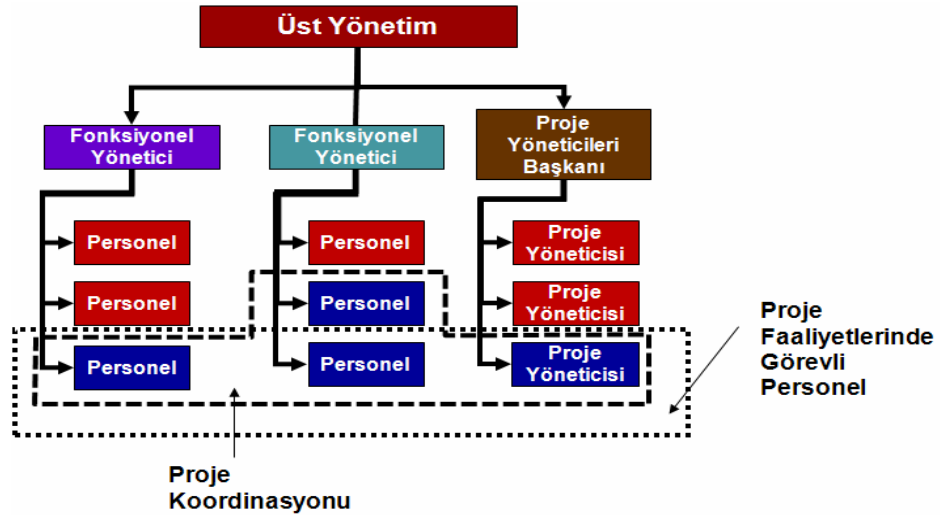
Diğer bir ifade ile proje elemanları ile işlevsel birim elemanlarının, sinerji sağlayacak yeni bir yapılanmayla projeye intibakını sağlayan yapısal bir araçtır (Wall, 1984).

Projede işlevsel yönetici ile proje yöneticisi arasındaki otorite paylaşımına göre "Zayıf (İşlevsel) Matris", "Dengeli Matris" ve "Güçlü Matris" olarak üç tür matris teşkilatlanma vardır.

Otoritenin büyük kısmının işlevsel yöneticide toplandığı matris yapılar için “zayıf matris” ifadesi kullanılmaktadır. Zayıf matriste proje yöneticisi bir yardımcı eleman gibi görev yapar, çizelgelemeleri, kontrol formlarını hazırlar, bilgi toplar ve projenin tamamlanmasına yardımcı olur. Projenin kendi alanına düşen parçasında kimin ne yapacağını ve işlerin ne zaman bitirileceğinin kararı işlevsel yöneticiye aittir.

Dengeli matris yapıda yöneticiler arasında eşit bir otorite paylaşımı vardır. Proje yöneticisi başarı için neler gerekli olduğunu belirlerken işlevsel yönetici bunların nasıl yapılacağıyla ilgilenmektedir. Proje yöneticisi genel planı yapar ve işlevsel birimleri projenin ilgili bölümü ile görevlendirir. İşlevsel yönetici proje yöneticisi tarafından belirlenen standartlar ve program çerçevesinde üzerine düşen sorumluluğu kendi atadığı personelle, kendi yönetiminde yerine getirmektedir.

Proje yöneticisi otoritenin büyük kısmına sahipse güçlü matris olarak adlandırılır (Şekil-1.8). Projede hangi uzmanların ne zaman, nerede, ne yapacaklarının kontrolü ve son kararlar proje yöneticisindedir. İşlevsel yöneticiye gerektiğinde danışılır, personeli üzerinde kısıtlı kontrolü vardır. Güçlü matris yapı, matris organizasyon içinde proje takımı hissi oluşturmaya çabalamaktadır (Gray ve Larson, 2000:232)



Şekil-1.8: Güçlü Matris Teşkilatlanma (PMI, 2000:22).

Belirtildiği üzere, işlevsel teşkilatlanma ile proje esaslı örgütlenmenin farklı faydaları olduğu gibi sakıncaları da vardır. Bu noktada, bir yapı tercih edildiğinde

diğer örgüt yapısının faydalarından yoksun kalınması ikilemi ortaya çıkmaktadır. Organizasyon arařtırmacıları bu ikileme bir çözüm olarak matris yapıyı önermişlerdir (Wright, 1979, 1980). Fakat matris yapının da kesin çözüm olmadığı, proje yönetiminde faydalı olarak kullanılabileceđi gibi, örgüte ve uygulandıđı ortama göre sakıncaları ve dezavantajları olduđu çeřitli kaynaklarda belirtilmiştir (Denis, 1986; Ford ve Randolph, 1992; Joyce, 1986; Kerzner, 1998; Posner, 1986).

Matris organizasyonların güçlü yönleri;

- Örgüt politikalarıyla çakışmamak kaydıyla her proje için bağımsız süreç ve politikalar oluşturulabilir.
- Tepki sürelerinin kısa olması deđişikliklere, çatışma çözümlemelerine ve proje ihtiyaçlarına çabuk cevap imkânı sağlar.
- Resmi olarak atanmış, koordine ve farklı birimleri bütünleřtirmeden sorumlu bir proje yöneticisinin olması projeye güçlü bir odaklanma sağlar. Bu, işlevsel örgütlerde gözden kaçan problemlerin çözümlenmesine daha kapsamlı bir yaklaşım sağlar.
- Yetki ve sorumluluk paylaşımı nedeniyle stres takımın tamamına dağılmaktadır.
- Zaman, maliyet ve performans arasında daha iyi denge sağlanmaktadır.
- Matris yapılar, kaynakların ve yeteneklerin daha esnek kullanımını sağlar. Bazı durumlarda işlevsel birimler, proje yöneticisi emrine personel görevlendirebilirler. Ancak, işlevsel birimlerin elemanlarının projeye olan katkıları, işlevsel yönetici tarafından takip edilir.

Matris organizasyonun zayıf yönleri;

- Farklı proje yöneticileri arasında kaynakların ortak kullanımını gerektiren durumlar çatışma yaratabilir.
- Her yönetici kendi projesi için en iyi olanı ister. Proje yöneticisi ile işlevsel yönetici arasında sürekli bir denge sağlanmalıdır.
- Matris yapı, yönetimde komuta birliđi prensibini ihlal etmektedir. Örneđin, bir proje elemanı, proje yöneticisinden ve işlevsel birim yöneticisinden ne yapması gerektiđi konusunda farklı talimatlar alabilir.

- Elemanlar ve yöneticiler rol belirsizliğine karşı klasik yapıda olduğundan daha hassastır.

### 1.3. PROJE YÖNETİMİNDE BAŞARISIZLIK VE NEDENLERİ

Projelerin hedeflerine güvenlik ve çevre faktörleri de dikkate alınarak belirlenen maliyetler çerçevesinde ve zamanında ulaşabilmeleri etkin yönetilmeleri ile mümkündür. Projelerin başarılı olması yanında başarısız olmamaları da önemlidir. Bu nedenle başarısızlığı getiren hususların çok iyi etüt edilmesi ve başkalarının yaptığı hataları tekrarlamayacak yapı, özellik, teşkil ve sistemlerin oluşturulması esas alınmalıdır. Proje organizasyonlarında yaygın olarak görülen başarısızlık nedenlerinin önemli bir kısmının işletmecilik ve yönetim ile ilgili olduğunu gösteren ve sık rastlanan başarısızlık durumları en çoktan en aza doğru aşağıda sıralanmıştır (Barutçugil, 2008:28):

- İşletmecilik deneyiminin olmayışı
- Proje planlama ve maliyet kontrol sisteminin yetersizliği
- İşletme sermayesinin yetersizliği
- Teknik, mali ve idari nitelikteki olağan güçlükler
- Aşırı rekabet
- Reklam ve tanıtma yetersizliği
- Teknik konularda yeterli beceri ve deneyimin olmayışı
- Hizmetler için yanlış yer seçimi
- Malzeme yönetimi ve stok kontrol sorunları
- Hırsızlık, yolsuzluk ve benzeri olaylar.

Benzer şekilde;

- Projenin, planlanan maliyetleri ve öngörülen zamanı aştığı ve yönetimde genel maliyetlerde ve işletme etkinliğinde olumsuz sonuçlar yarattığı,
- Ticari bir sözleşme gereği yüklenilen projeden beklenen kârın, artan maliyetler veya gecikmeler kapsamında ödenen tazminatlar sonucunda önemli ölçüde azaldığı ve giderek zarara dönüştüğü,

- Projelerin gecikmeli olarak bitirilmesinin yatırımlardan beklenen faydayı azalttığı ve proje firmasının ya da yöneticisinin saygınlığını tehlikeye düşürdüğü,
- Projenin, doğaya, insan sağlık ve güvenliğine tehdit oluşturacak sonuçlar yarattığı,
- Gecikmeli tamamlanan projenin işletme amaç, plan ve politikalarını olumsuz yönde etkilediği,
- Pazardaki yeni iş olanaklarının değerlendirilememesine; pazarın, müşterilerin, kurumsal imajın ve rekabetçi avantajların yitirilmesine yol açıldığı durumlarda proje yönetimi bakımından bir başarısızlıktan söz edilebilir.

Çok sayıda projeyi bir arada yürüten organizasyonların herhangi bir projede başarısız olmaları, başarıyla tamamlanmış çok sayıda projeden sağlanan kazancın kaybına neden olabilir. Büyük projelerde yaygın yürütme başarısızlıkları genellikle istenmeyen sonuçların önlenmesi ve düzeltici tedbirler alınması için çok geç oluncaya kadar fark edilmediklerinden, büyük projeleri üstlenen her işletmenin bünyesinde etkin proje yönetimi gerçekleştirebilecek yetenek, bilgi ve beceriye sahip uzmanlar bulundurması zorunludur.

Proje yönetimiyle, projede baştan sona bütün sürecin tam manasıyla kontrolü amaçlanmaktadır. Bu sayede, projenin tahsis edilen bütçe ile zamanında bitirilmesi ve istenilen nitelikte sonuçların elde edilmesi sağlanmaktadır.

Proje yönetiminde başarının sağlanması, yönetim görevine uygun yetki ve sorumluluk paylaşımının sağlanması, iş konusunda yetkin niteliklere sahip bir organizasyon yapısının teşkili ve bütünleşik bir planlama ve kontrol sisteminin tesisi ile mümkündür.

Proje yöneticisi, işletmenin en üst düzeydeki genel yöneticisi ya da bir proje için özel olarak atanmış bir yönetici de olabilmesi ve projeyi yönetme ve yönlendirme konusunda tüm yetki ve sorumluluklara sahip olması yanında, bütün çabaları bütünleştirici bir rol üstlenmektedir. Bu manada projenin başarılı bir şekilde yönetilebilmesi proje yöneticisinin bütünleşik yönetim sistemlerini anlaması, benimsemesi, kullanabilmesi ve sonuçlarını değerlendirebilmesi ile sağlanabilir. Proje yöneticisi bu sayede planların doğru ve geçerli olmasını, uygun yönetim kararlarının alındığının ve uygulandığının doğrulanmasını, doğru bir planlama ile

proje ekibinin üyeleri arasındaki ilişkilerin düzenlenmesini, sorun ve çatışmaların çözümünü ve çözilemeyen sorunların yönetim kademelerine iletilmesini sağlar.

Bütünleşik yönetim; amaç ve hedeflerin uzlaşmayla belirlenmesi, işin tanımlanması yanında mevcut kaynaklara göre plan ve programlamasının yapılması ve gelişmelerin düzenli ve sağlıklı olarak ölçülmesi yoluyla sağlanır. Bütünleşik yönetim aynı zamanda; projelerde beklenen teknik sonuçların, zamanın, para ve insan gücü gibi kritik kaynaklarla ifade edilebilecek maliyetlerin ve projenin çevre ve güvenlik boyutlarıyla ilgili önemli bilgi unsurlarının projenin tüm aşamaları için bir araya getirilmesini proje yönetiminde etkinliğin sağlanması bakımından zorunlu kılar.

#### **1.4. PROJE YÖNETİMİ BİLGİ ALANLARI**

Proje yönetimi bilgi alanları 9 ana başlık altında incelenecektir. Her bir bilgi alanı içerik açısından çok zengin ve kapsamlıdır. Bu bölümde bilgi alanları ve süreçlerinin detaylı olarak açıklanması tezin kapsamını aşacağından her bir bilgi alanı ana hatlarıyla belirtilmiş ve özetlenerek açıklanmıştır.

##### **1.4.1. Bütünleştirme (Integration) Yönetimi**

Bütünleştirme yönetimi, çeşitli proje elemanlarının uygun şekilde koordine edilmesini sağlayan süreçler vasıtasıyla gerçekleştirilir. Hedef ve alternatiflerin karşılaştırılması sonucunda, değerlendirmelerin ihtiyaçları ne ölçüde karşılandığı/aşıldığı ölçütüne göre yapılmasını sağlar. Ana süreçler (Dengiz, Erceiş, Karadağ ve Şahmalı, 1998:47, [http://www.yancy.org/research/project\\_management/integration.html](http://www.yancy.org/research/project_management/integration.html));

- **Proje İlgi Alanı Oluşturma:** Diğer planlama süreçlerinden alınan sonuçların tutarlı ve uyumlu bir doküman içinde toplanmasıdır.
- **Proje Planının Yürütülmesi (Uygulama):** Proje planının, içinde yer alan etkinliklerin yerine getirilip getirilmediğinin sağlanmasının yapılması faaliyetidir.
- **Kapsayıcı Değişiklik Kontrolü:** Tüm proje göz önüne alınarak yapılacak değişikliklerin koordinasyonudur.

##### **1.4.2. Kapsam Yönetimi**

Kapsam yönetimi, tasarlanan proje hedeflerinin başarıyla gerçekleştirilmesi için ihtiyaç duyulan bütün faaliyetlerin icra edilmesini sağlayan yönetim yaklaşımıdır. Projenin içinde nelerin yer alıp, nelerin yer almayacağını belirlenmesi ve kontrolü bu yaklaşımın temel ilgi sahasını oluşturmaktadır. Proje kapsamında yapılacak her



türlü deęişiklik proje maliyeti, proje süreçleri ve tamamlanma süresini direkt olarak etkilediğinden, proje kapsamının projenin başlangıcından itibaren sabitlenmesi projenin öngörülen plan dâhilinde tamamlanması bakımından zorunludur. Bu yaklaşım 5 temel süreçten oluşur (Dengiz, Erceiş, Karadağ ve Şahmalı, 1998:57).

- **Başlama:** Projenin belirlenen ilk kapsamı dâhilinde başlatılması konusunda örgütün taahhüdüdür.
- **Kapsam Planlama:** Gelecekteki proje kararlarına temel teşkil edecek yazılı kapsam beyanıdır (Scope Statement).
- **Kapsam Tanımlama:** Proje sonucu orta çıkacak ürün ya da hizmetin kolay yönetilebilecek küçük parçalara bölünmesidir.
- **Kapsam Doğrulama:** Proje kapsamının proje paydaşlarınca resmî kabulüdür.
- **Kapsam Deęişiklik Kontrolü:** Proje kapsamındaki deęişiklerin kontrolüdür.

#### 1.4.3. Zaman Yönetimi

Zaman yönetimi, projenin zamanında tamamlanması için gerçekleştirilen faaliyetleri içeren süreçleri öngörür. Bunlar (Kerzner, 2003);

- **Etkinlik Tanımlama:** İş dağılım ağacında belirtilen çeşitli proje çıktılarını oluşturmak üzere gerekli olan iş paketlerinin (etkinliklerin) tespit edilmesidir.
- **Etkinlik İlişkilendirme:** Etkinlikler arası ilişkilerin belirlenmesi ve yazılı hale getirilmesidir.
- **Etkinlik Süre Tahminleri:** Her etkinliğin tamamlanması için gerekli iş süresi tahminidir.
- **Program Geliştirme:** Proje programını oluşturmak üzere etkinliklerin birbirleriyle olan ilişkilerinin, sürelerinin ve kaynak gereksinimlerinin çözümlenmesidir.
- **Program Kontrolü:** Proje programından sapmaların izlenmesi ve düzeltici önlemlerin alınmasıdır.

#### 1.4.4. Maliyet Yönetimi

Maliyet yönetimi, projenin onaylanmış bütçe sınırları içerisinde tamamlanmasını sağlayan gerekli süreçleri içerir (Rad, 2002; Pilcher, 1973).

- **Kaynak Dağılımı:** Proje faaliyetlerinin yürütülmesi için gerekli kaynakların (insan, teçhizat, malzeme b.) ve bunların ne miktarda kullanılacağına belirlenmesidir.
- **Maliyet Tahmini:** Proje faaliyetlerinin tamamlanması için ihtiyaç duyulan kaynakların maliyetinin yaklaşık bir tahmininin yapılmasıdır.
- **Maliyet Bütçeleme:** Genel maliyet tahmininin münferit iş kalemlerine dağıtılmasıdır.
- **Maliyet Kontrol:** Proje bütçesine yapılacak değişikliklerin kontrol edilmesidir.

#### 1.4.5. Kalite Yönetimi

Kalite Yönetimi, projenin yüklenilen ihtiyaçları karşılama temin edici süreçleri kapsar. Kalite politikası, hedefler ve sorumlulukları belirleyen genel yönetim işlevinin tüm etkinliklerini ve bunları kalite sistemi içinde kalite planlaması, kontrol, kalite güvence ve kalite iyileştirme gibi araçlarla uygulamayı kapsar (Kerzner, 2003; TBD, 1999:83).

- **Kalite Planlama:** Projeye ilgili kalite standartlarının ve nasıl karşılanacağına belirlenmesidir.
- **Kalite Güvence:** Tüm proje performansının projenin ilgili kalite standartlarını karşılaması kapsamında düzenli olarak değerlendirilmesi faaliyetidir.
- **Kalite Kontrol:** İlgili kalite standartlarının karşılanma durumunun görülmesi amacıyla proje sonuçlarının takip ve kontrolü yanında yetersiz performansla neden olan hususları ortadan kaldıracı yöntemlerin seçilmesini ve gerekli önlemlerin alınmasını sağlar.

#### 1.4.6. İnsan Kaynakları Yönetimi

İnsan Kaynakları Yönetimi, proje paydaşlarının proje içindeki etkinliklerini değerlendirilerek, proje ekibi içinde etkinliği artırıcı yöntemlerin belirlenmesi, yeterli sayıda personelin istihdamını, fazla sayıda personelin (varsa) projeden tasfiyesi konularıyla ilgilendirir. Proje paydaşlarını; proje yöneticisi, müşteri/kullanıcı, proje

yürüten örgüt, proje takım elemanları ve yüklenici oluşturmaktadır (Dengiz, Erceiş, Karadağ ve Şahmalı, 1998:121-129).

- **Örgütsel Planlama:** Proje görev ve sorumlulukların ve raporlama ilişkilerinin belirlenmesi, yazılı hale getirilmesi ve örgüt içindeki proje paydaşlarına tahsis edilmesidir.

- **Personel Tedarik:** İnsan kaynakları yönetimi sayesinde projeye tahsis edilen insan kaynağının projeye atandırılması ve projenin uzman bilgi ve tecrübe gerektiren bölümlerinde yeteri kadar personelin istihdam edilmesidir.

- **Takım Oluşturma:** Proje performansının yükseltilmesi için münferit şahıs ve grup becerilerini bir araya getirecek şekilde etkin ve yetkin bir teşkilatlanmaya gidilmesidir. Projenin ilgi sahası bakımından yeteri kadar tecrübe ve bilgi seviyesine sahip olmayan personelin yeteneklerinin geliştirilmesi kapsamında uygulanacak/aldırılacak eğitim de bu çerçevede değerlendirilmektedir.

#### 1.4.7. İletişim Yönetimi

İletişim Yönetimi, zamanlı ve uygun seviyede proje bilgilerinin toplanması, elleçlenmesi/harmanlanması, düzenlenmesi, yayınlanması ve dosyalanması/saklanması maksadıyla icra edilen faaliyetleri/süreçleri kapsar. Başarı bilginin, uygun zamanda (belirli zaman aralıklarında periyodik olarak), uygun birimler/makamlarca, bilmesi gereken prensibine paralel olarak kesintisiz olarak aktarılması sayesinde elde edilebilir. Proje içinde yer alan tüm proje paydaşlarının "proje dilinde" konuşmasını sağlamak üzere herkesin her şeyi aynı şekilde anlamasını sağlayacak bir jargon oluşturulmalı ve benimsetilmelidir (PTD 1999:45-52).

- **İletişim Planlama:** Projeye dâhil olanlar arasındaki bilgi ve iletişim ihtiyacının tanımlanması ve etkileşimli bilgi transferinin nasıl gerçekleştirileceğinin kesin hatlarıyla ortaya konması için uygulanan planlama süreçleridir.

- **Bilgi Dağıtım:** Gerekli bilgilerin zamanında oluşturularak kullanıma sunulmasıdır.

- **Performans Raporlama:** Performans bilgilerinin toplanması ve dağıtılmasıdır. Durum raporları ve gelişim ölçüm ve tahminleri bunlardan bazılarıdır.

- **İdari Kapanış:** Safha ya da projenin tamamlanmasını resmileştirmek üzere bilgilerin toplanması, elleçlenmesi/harmanlanması ve dağıtılmasıdır.

#### 1.4.8. Tedarik Yönetimi

Tedarik yönetimi, malzeme ve hizmetlerin proje örgütü dışından sağlanan/sağlanması planlanmış kısmı ile ilgilenmektedir. Konuyu basitleştirmek için malzeme ya da hizmet bir veya birçok olsa da "ürün" olarak adlandırılacaktır. Ana süreçler (Dengiz, Erceiş, Karadağ ve Şahmalı, 1998:161-166);

- **Tedarik Planlama:** Projenin planlanmış safhaları doğrultusunda; projenin hangi aşamasında nelerin, ne zaman ve nasıl tedarik edileceğinin planlanmasıdır.
- **İhale Planlama:** Ürün ihtiyaçlarının yazılı hale getirilmesi ve muhtemel temin kaynaklarının belirlenmesidir.
- **İhale:** Teklif ya da proje önerilerinin alınmasıdır
- **Kaynak Seçimi:** Muhtemel yükleniciler arasında seçim yapılmasıdır.
- **Sözleşme İdaresi:** İlişkilerin sözleşme ile yürütülmesi demektir.
- **Sözleşme Kapanış:** Sözleşmenin sonlandırılması kapsamında icra edilecek faaliyetleri içerir.

#### 1.4.9. Risk Yönetimi

Projede risk denilince, projeyi sekteye uğratabilecek, proje takvimi, belirlenmiş proje süresi ve onaylanmış proje kapsamında ön alınmadığı takdirde olumsuz değişikliklere sebebiyet verebilecek hayati öneme haiz olay ya da belirsizlikler akla gelmelidir. Bu tehlike her aşamada etkin bir risk yönetimi ile bertaraf edilebilir. Muhtemel problemler dikkate alınarak yapılan planlamaların değerlendirilmesi ve bu problemleri yok edici veya etkisiz kılıcı stratejiler belirlenmesi, projenin başarımına katkı ve şans sağlayacaktır (Ritchie ve Marshall 1993, Skipper 1998, Barkley 2004, Cervone 2006:256-262, Royer 2002).

Etkili bir risk yönetimi için tavsiye edilen aşamalar aşağıda sunulmuştur.

- Risklerin belirlenmesi,
- Risklerin niceliklerinin (boyutunun ve muhtemel etkisinin) belirlenmesi,
- Riskleri önleyici planların hazırlanması ve risklere müdahale esaslarının bu çerçevede belirlenmesi,
- Risklerin yönetimi, takip ve kontrolü.

Risklerin yeter miktarda analizi, ayrıntılı bir plana ihtiyaç gösterir. Bu nedenle de risk analizi projeye başlamadan önce yapılmalıdır. Risk analizi bir kez icra edilip bitirilen bir faaliyet olmayıp, aksine proje tamamlanıncaya kadar hemen her aşamada, uygulamadan ya da yanlış planlama ve öngörülerden kaynaklanan büyük sapmalar dâhilinde gerçekleştirilmesi gerekli süreklilik arz eden bir süreçtir. Üzerinde kafa yorularak proje risklerinin zamanında belirlenmesi ve önlem alınmasıyla projede meydana gelebilecek büyük çaplı olumsuzluklar engellenebilmektedir.

Yukarıda bahsi geçen tüm süreçler, mensubu oldukları bilgi alanı içinde olduğu gibi diğer bilgi alanlarındaki süreçlerle de sürekli bir etkileşim içindedirler. Her bir süreç, projenin ihtiyaçlarının karşılanması bakımından birden fazla bireyin ya da grubun gayret birliği yapmalarını zorunlu kılar.

## İKİNCİ BÖLÜM

### PROJE YÖNETİM TEKNİKLERİ

#### 2.1. GENEL

Proje yönetim teknikleri, proje yönetimine daha taktik düzeyde yaklaşım sağlamakta ve alınan stratejik kararların etkin bir biçimde gerçekleştirilmesi ile ilgilenmektedir. Kendilerine özgü planlama, yönetim, koordinasyon, organizasyon ve kontrol sorunları nedeniyle başarılı bir proje yönetimi, endüstriyel işletme yönetiminden farklı yaklaşımlar içinde olmak ve farklı bilgi ve becerilerden yararlanmak zorundadır. Bu da projelerin planlanması ve kontrolü amacıyla özel yönetim tekniklerinin geliştirilmesini zorunlu kılmaktadır.

Günümüzde, her birinin kendine özgü üstünlükleri ve sakıncaları bulunan birçok proje yönetim tekniği geliştirilmiştir. Herhangi bir tekniğin her yerde ve her koşulda diğerlerine göre daha üstün veya daha kullanışlı olduğu iddia edilemez ve proje yönetiminin farklı aşamalarında farklı tekniklerin uygulanması ile daha başarılı sonuçlar elde edilebilir. İlk planlama aşamasında belirsizliklerin yoğun olması nedeniyle ileri düzeyde karmaşık ve geliştirilmiş teknikler projenin ayrıntıları açıklık kazandıkça kullanılır. Örneğin, bir inşaat projesinin başlangıcında basit bir GANTT şeması, planlama ve kontrol açısından yeterli olurken, yürütülecek görevlerin, bunların sürelerinin ve maliyetlerinin daha belirgin olarak ortaya çıktığı ileri aşamalarda PERT ve CPM türü ağ analizlerine geçilebilecektir. Ayrıca, projelerin özel gereklerine uygun olduğu sürece çeşitli tekniklerin bir arada kullanılması da yönetim etkinliğini artıracaktır.

Hangi teknik kullanılırsa kullanılsın, planlanan ve kontrol edilen bir projenin üç temel amacının (ya da tek amacın üç ayrı yönünün) aynı derecede önem taşıdığı bilmesi gerekir. Bunlardan herhangi birinin gerçekleştirilmesi için diğerleri göz ardı edilemeyecektir. Bir ürünün üstün teknik özelliklerde üretilmesi, daha uzun bir sürenin harcanmasını ve daha yüksek maliyetlere katlanılmasını gerektirir. Zamanın kısaltılmak istenmesi ise maliyetleri artırır. Maliyetleri azaltmak ve daha kısa sürede projeyi sonuçlandırmak istendiğinde de teknik başarı düzeyi düşecektir. Başarılı bir proje yönetiminde, bu üç boyut arasında denge sağlayıcı bir yapı kurgulanır.

PERT ve CPM, faaliyetlerin bağımlılıklarını ve birbirlerine göre öncelik durumlarını göz önüne almaktadır. Ancak karmaşık projelerde de genellikle yazılımlara aktarılmış olan PERT ve CPM programlama tekniklerinin bir ayağını basitleştirilmiş GANTT şemaları oluşturmaktadır.

Bu bölümde proje yönetim teknikleri üzerinde durulacak ve çok detaya inilmeden genel mahiyette bilgi verilecektir.

## **2.2. PROJE YÖNETİM TEKNİKLERİ**

### **2.2.1. Proje Kontrol Tabloları**

Proje yöneticileri, kullanacakları tekniklerin öncelikle şu üç konuda kendilerine bilgi sağlamasını beklerler:

- Yerine getirilecek her görevin gerektireceği zaman ve maliyetler,
- Öngörülen programın gerisinde kalan görevler ve bunların projenin genel gelişimi üzerindeki muhtemel etkileri,
- Planlanan maliyet ve öngörülen bitiş süresi ile karşılaştırmalı olarak belirlenen projenin gelişme düzeyi.

Uygun bir yapı oluşturulduğu takdirde ilk iki maddeyle ilgili sorun yaşanması beklenmez. Yönetici, üçüncü madde gereği olanla olması gereken arasında fark tespit ettiğinde, çözüm için kaynakların yeniden dağıtılmasını, yeni kaynakların sağlanmasını veya son bir çözüm olarak planın yeniden gözden geçirilmesini isteyebilir.

Zaman içinde belirli bir noktadaki genel teknik gelişme düzeyi ve toplam harcama rakamlarının planla karşılaştırılması sayesinde;

- Projenin hangi bölümlerinde planlananın ilerisinde, hangi bölümlerinde gerisinde kaldığı,
- Hangi bölümünün önceden tahmin edilen maliyetten çok daha fazlasına, hangi bölümünün çok daha düşüğüne mal edildiği kolaylıkla görülebilir. Ancak, projenin nasıl geliştiğinin görülmesi bakımından belirli bir dönem boyunca bu değerlendirmelerin yapılması çok daha yararlıdır. Bu veriler, bir yöneticiye en uygun biçimde;
- Zaman ve toplam maliyetler,
- Maliyetler ve teknik gelişme,

- Zaman ve teknik gelişmenin birlikte gösterildiği “Proje Kontrol Tabloları” yardımıyla sunulabilir.

Bu üç tablonun, bir projenin uygulanmasına farklı bakış açıları getirmesi nedeniyle birlikte çizilmesi doğru olacaktır. Bu sayede maliyet artışları ile yüksek teknik gelişmenin birbirlerine göre değerlendirilmesi (yaşanan teknik bir sorun nedeniyle gelişme yavaşladığında maliyetlerin yükselmesi gibi) ve bu değerlendirmenin planlanan zaman açısından irdelenmesi (maliyet artışı olmasına rağmen proje zamanında tamamlanabilme durumunun ortaya konması) mümkün olmaktadır. Bu bağlamda, projenin başarısının, maliyetlerden çok tamamlanma süresine karşı duyarlı olması durumunda proje için yakın zamanda bir tehlike bulunmadığı ifade edilebilir.

Proje kontrol tablolarında bir projenin, maliyet ve teknik gelişme durumunu birikimli (cumulative) değerlerle ifade ederek bir özet proje raporu çıkarma olanağı bulunmaktadır.

### **2.2.2. İş Dağılım Ağacı (Work Breakdown Structure-WBS)**

İş dağılım ağacı proje kapsamının; yönetilebilir, öngörülebilir, sorumlu personele veya birime planlanabilir nitelikte iş paketlerine ayrılmasını amaçlamaktadır. WBS, 1960'lı yıllarda proje tanımlamak için geliştirilmiş daha sonra planlama ve kontrol sisteminin ana unsuru haline gelmiştir. WBS, projedeki karmaşıklığı basit, yönetilebilir bileşenlere ayırmaya yardım eden hiyerarşik bir bölümlendirmedir. WBS'nin kullanılma nedenleri aşağıda açıklanmıştır.

- Proje ömür devri için önceden geliştirilen bir iş dağılım ağacı proje kapsamını kesin tanımlanabilir bir hale getirerek proje paydaşlarının projeyi açıkça anlamalarına yardımcı olmaktadır.

- WBS’de tanımlanan her bir iş paketi, maliyetlerin, çizelgelerin ve üretim bilgilerinin öngörülmesinde ve izlenmesinde, projeyi yürüten personele, gerçekleştirdiği faaliyetleri karşılaştırmak için bir kontrol noktası sağlamaktadır.

- Personelin değişmesi ve çok sayıda personelin katılmasıyla ortaya çıkabilecek kavram karmaşasını ortadan kaldırmakta, her terimin personel için aynı anlamı ifade etmesini sağlamaktadır.



- WBS, bilgi sistemleri arasındaki bilgi akışını tesis edici bir temel yapı oluşturur. Birimler arası raporlama ve takip faaliyetlerini kolaylaştırır (Lavold, 1988: 322).

### **2.2.3. GANTT Şemaları ve Denge Hattı Tekniği**

Proje planlama ve kontrol amacıyla yaygın olarak kullanılan en basit teknik GANTT şemalarıdır. Doğrudan işin parçalanmış yapısına bağlı olan bu şema, yatay ekseninde zaman birimlerine (gün, hafta veya ay gibi) ayrılmış bir ölçeğin bulunduğu ve dikey ekseninde de proje unsurlarının sıralandığı iki boyutlu bir grafikdir. Her proje unsurunun süresini, durumunu, çubuklar, çizgiler ve diğer semboller yardımıyla, zaman ölçeği ile ilişkili olarak planlama ve izleme olanağını sağlar. Bu şemalar, genellikle mevcut zamanı ve toplam işi gösteren bir özet raporlama tekniği ile birlikte, projenin tamamlanması için gerekli basamakların yönetici tarafından izlenmesini kolaylaştırır. GANTT şemalarında zaman çizgisi boyunca her proje aktivitesi için yatay barlar çizilir.

GANTT şemalarının anlaşılması kolaydır. Bu şemalar herhangi bir zamanda hangi aktivitenin gerçekleştirilmesi gerektiğini, daha da önemlisi projenin ilerleyişini günlük olarak gösterebilir ve böylelikle gerektiğinde düzeltici önlemlerin alınabilmesini sağlar.

Düşük maliyetli olan GANTT şemaları yöneticilere, bütün aktivitelerin planlanması, bunların performans sıralarının dikkate alınması, aktivite zamanlarının tahmin edilip kaydedilmesi, bütün proje için gereken zamanın ortaya çıkarılması konularında yol göstermektedir.

GANTT şemasının avantajı çok çabuk hazırlanabilmesidir. Karmaşık olmayan projelerde bu tip programlama şemaları yalnız başlarına kullanılabilir. Bu şemalar yöneticilere her bir aktivitenin ilerlemesini izlemelerine, problemleri alanları açığa çıkarmalarına ve üstesinden gelmelerine izin verir. Ancak GANTT şemaları aktiviteler ve kaynaklar arası ilişkileri gereği gibi gösteremez (Burke, 1999).

1900'lerde Henry L. GANTT tarafından geliştirilen ilk şemalarda faaliyetleri ve görevleri tamamlamak için önem verilen faktör, işçilerin ve makinelerin kapasiteleriydi. "Gelişme Şeması", "İnsan ve Makine Kayıt Şeması" ve "Yük Şeması" gibi isimler alan bu ilk şemalar üzerindeki yatay hatlar, belirli bir zaman aralığında tamamlanan işin miktarı ile programlanan miktar arasındaki ilişkiyi göstermekteydi.

Bu tekniklerin uygulamasının yaygınlaşması üzerine, özellikle bazı projelerde, zamanın kaynaklardan daha önemli bir faktör olduğu ortaya çıkmıştır. Böyle durumlarda, projenin olabildiğince kısa bir zamanda tamamlanabilmesi için kaynaklar belirli sınırlar içerisinde sağlanabiliyordu. Bu nedenle, projenin planlama ve kontrolünde kullanılacak GANTT şemalarında faaliyetlerin zaman içindeki gelişme düzeylerinin izlenmesi, daha fazla önem ve ağırlık kazanan bir amaç olmuştur.

GANTT şemaları, yönetim için önemli yararlar sağlamakla birlikte projenin etkin planlanması, programlanması ve kontrol edilmesi açısından ek ve daha sağlıklı bazı temel bilgileri göstermemektedir. Örneğin, tasarım, mühendislik, tedarik, inşaat, tesis ve teslim gibi faaliyetler arasındaki ilişkiler gösterilmemekte ve;

- Bu işlerin hangi kısımları aynı zamanda, paralel olarak yapılabileceği,
- Her işin hangi kısımlarının diğer işlerin başlamasından önce bitirilmesi gerektiği,
- Projenin zamanında bitirilmesini sağlamak için hangi işlere veya işlerin belirli kısımlarına öncelik verilmesi gerektiği,
- Bazı işlerin veya bu işlerin bazı kısımlarının başlangıç ve sonuçlanma tarihleri isteğe bağlı olarak değiştirilebilme durumlarına ilişkin bilgi verilmemektedir.

GANTT şemaları, günümüze kadar yapılan çeşitli geliştirme ve iyileştirmelerle önemini ve yararlılığını korumuştur. Yönetime projenin durumunu açıklıkla özetleyebildiği için son derece yararlı bir iletişim aracı olmuş ve proje kontrolünde etkinliği artırmıştır. Ancak, projelerin ölçeği büyüdükçe ve karmaşıklığı arttıkça GANTT şemalarının yetersiz kaldığı görülmektedir. Bu şemaların gözden geçirilmesi ve güncelleştirilmesi son derece güç ve zaman alan bir faaliyettir.

Bu eksiklikleri gidermek üzere, GANTT şemalarının ve projenin çeşitli aşamaları arasındaki bağılıkları daha açık bir biçimde sıralayan “Denge Hattı Tekniği” geliştirilmiştir. Bu teknik, projenin çeşitli aşamaları arasındaki ilişkilerin belirlenmesini, uygulamada denge ve uyumun sağlanmasını ve dolayısıyla daha uygun sonuçlara ulaşılmasını amaçlamaktadır.

Temelde imalat süreçlerinde, örneğin montaj hattı dengelenmesinde kullanılan bu teknik, projelerde de kullanılabilen yararlı bir araçtır. Burada öncelikle, bir proje sürecinin zaman içindeki aşama/işlem sırasının bilinmesi, daha sonra da öngörülen aşama/işlem çıktılarının zaman ve miktar olarak teslim tarihlerinin karşılaştırılması ve belirli dönemlerde durumun gözden geçirilerek değerlendirilmesi gerekir. Bu değerlendirme sırasında geciken ya da plan hedeflerinden önde olan üretim aşamaları fark edildiğinde gerekli düzeltici ve dengeleyici önlemler alınacaktır.

Denge hattı tekniğinin projelerde daha başarılı olarak kullanımı, bir taraftan yalnızca bir kez yapılan faaliyetleri ve diğer taraftan tekrarlanarak yapılan işlemleri birlikte gösterebilmesine bağlıdır. Bunu sağlayan yöntem, denge hattı tekniği ile ağ analizini bir araya getiren PERT-LOB sistemidir. Burada çizilen ağ diyagramı bilinen PERT diyagramından farklı değildir. Yalnızca, tekrarlanan faaliyetleri belirten oklar kesikli çizgilerle, tekrarlanan olaylar ise üzerlerine (x) işareti konularak belirtilir. Bu diyagramda tekrarlanan olaylar sistemin temel kontrol noktalarını oluştururlar. Faaliyetler, en az tekrarlandıktan en fazla tekrarlanana doğru sıralandıktan ve kodlandıktan sonra PERT tekniğine uygun olarak çizilen bir ağ diyagramı (yörünge ağı) üzerinde gösterilirler. Zaman hesaplamaları PERT'de olduğu gibidir. Belirli tek fark, tekrarlanan faaliyetlerin en erken ve en geç başlama ve bitirme zamanları ile boş zamanların her durum için ayrı hesaplanmasıdır.

#### **2.2.4. Ağ Analizi (Network Analysis) Teknikleri**

Ortak bir amaca yönelik, birbirleriyle bağlantılı bir dizi faaliyetten oluşan bir projede en yaygın kullanılan proje yönetim tekniği ağ analizidir. Özellikle, iyi tanımlanmış bir başlangıç noktasına ve açık bir amaca sahip bulunan projelerde bu yöntem çok yararlı ve başarılı olmaktadır.

Ağ analizi, bir projenin amacına ulaşabilmesi, diğer bir ifadeyle, projenin en kısa sürede ve en az maliyetle gerçekleştirilmesi için yapılması gereken faaliyet ve olaylardan oluşan, bu faaliyetler ve olaylar arasındaki sıra ve mantık ilişkilerini gösteren grafikler yardımıyla sürdürülen bir proje yönetim tekniğidir (Battersby 1970).

1960'lardan başlayarak geliştirilen CPM ve PERT teknikleri, bilgi sistemlerinin de gelişmesi sonucu çeşitli büyüklük ve karmaşıklıkta imalat ve inşaat projelerinde bugün yaygın olarak kullanılmaktadır. Bu tekniklerin, çeşitli

ve karmaşık sorunların çözümüne yönelik çok sayıda uygulama alanı bulunmaktadır.

Ağ diyagramı yöntemleri, uygulamada değişik kullanım amaçlarına ve kapsamlarına göre çeşitli isimler almaktadırlar. CPM ve PERT dışında, belki yüzlerce isim altında geliştirilen ve kullanılan ağ diyagramı yöntemleri bulunmaktadır.

Ağ analizi tekniklerinin çoğu temelde dört aşamalı bir süreç öngören bir yaklaşım sergilemektedir. Bu aşamalar:

- Projedeki her işin mantıksal bir ağ üzerinde yerinin belirlenmesi
- Kritik önemi olan işlerin, yani projenin, tamamlanmasını etkileyecek nitelikteki işlerin belirlenmesi
- Kaynakların (insan, makine, malzeme v.b.) her işe bir bütün olarak, sistemin maliyetler ve süre açısından uygunluğunu sağlayacak biçimde dağıtılması
- Gelişmenin izlenmesi ve belirlenen amaçlara ulaşmak için eğer gerek duyuluyorsa kaynak dağıtımının yeniden düzenlenmesidir.

Ağ analizi, karmaşık projelerde faaliyetleri birbirinden ayrı olarak ele alır. Her faaliyet başlamadan önce hangi işlerin yapılmış olması gerektiği, o faaliyet bitirildiğinde hangi işlere hemen başlanabileceği ve faaliyet sürdürülürken hangi işlerin de aynı zamanda yürütülebileceği belirlenir. Buradan hareketle, faaliyetler arasındaki ilişkileri gösteren bir diyagram çıkarılır. Böyle bir diyagrama sahip olan proje yöneticisi, toplam proje zamanını ve kaynaklarını en iyi kullanacak planlama ve kontrol imkânına da ulaşmış olacaktır.

#### **2.2.4.1. Kritik Yol Metodu (Critical Path Method, CPM)**

Kritik Yol Metodu projelerin etkili şekilde planlanması ve kontrolünü sağlayan bir tekniktir. Kullanıcıları karmaşık ve büyük ölçüdeki proje bilgilerini hızlı ve düzenli bir şekilde yapılandırmasına yardımcı olmaktadır.

CPM, ağ diyagramı oluşturulması esasına dayanır. Projedeki faaliyetleri göstermek için ok diyagramları kullanılır. Oklar faaliyetler arasındaki ilişkileri belirleyecek şekilde birbirine bağlanır, bu şekilde proje başından sonuna kadar olan iş akışı gösterilmiş olur. Ağ diyagramının doğru bir şekilde hazırlanabilmesi için;

birbirinden bağımsız, öncelikli ve eş zamanlı gerçekleştirilebilecek faaliyetler belirlenmelidir. Ağ analizi sadece faaliyetler arasındaki ilişkiyi değil aynı zamanda her faaliyetin zaman ve maliyet açısından tamamlanma miktarını da ortaya koymaktadır (Wolf ve Hauck, 1985).

CPM uygulamak için çizilen ağ diyagramındaki tüm faaliyetlere süreleri ve maliyetleri atanır, iş takvimi oluşturulur. Daha sonra kritik faaliyetler belirlenir. Bir faaliyetin başlama tarihindeki gecikme, tüm projenin tamamlanma tarihinde gecikmeye sebep oluyorsa o faaliyet kritik faaliyet olarak adlandırılır ve kritik yola dahil edilir.

Kritik yol, ağ diyagramının başlama ve bitiş olaylarını birleştiren faaliyetler zinciri olarak tanımlanmaktadır (Halaç, 1991 :189).

#### **2.2.4.2. Program Değerlendirme ve Gözden Geçirme Tekniği (Program Evaluation and Review Technique, PERT)**

PERT, projelerin planlanması, çizelgelenmesi ve kontrolüne yardımcı olmak üzere tasarlanmış ağ esaslı bir modeldir. PERT'in uygulamasındaki temel beklenti, zaman ve kaynak harcanan birbirleriyle ilişkili çizelgeleme faaliyetlerine analitik anlamlar kazandırmaktır. İşletmelerde yöneticiye uygun ve kesin karar vermede yardım eden en önemli metotlardan biri PERT'dir.

PERT ağı ile yönetici ve işletmeci planlama, programlama ve kontrol fonksiyonlarını yerine getirirken hangi faaliyetlerin hangileri ile ilişkili olduğunu saptayarak alacağı kararların hangi faaliyetleri etkileyeceğini ve hangi faaliyetler için ayrıca karar alması gerektiğini bilmektedir.

PERT tekniği ile toplam faaliyet içinde herhangi bir alt faaliyetin aksaması ve gecikmesi etkilerinin neler olacağını zaman kaybına uğramaksızın veya minimum bir zaman kaybı ile öğrenmek mümkündür. Çünkü PERT metodu, dinamik bir programlamaya olanak vererek yöneticinin düzeltici tedbirleri derhal almasını sağlamaktadır.

PERT yalnız sonuç üzerinde doğrudan doğruya etkili olan kritik faaliyetlere ait bilgileri vermekle kalmaz aynı zamanda kritik olmayan faaliyetlerdeki kullanılabilir artık zamanları da belirterek bu faaliyetlerde kullanılan işgücü ve diğer kaynakların kritik faaliyetlere kaydırılması yolu ile daha verimli bir şekilde kullanılmasını sağlamaktadır (Burke, 1999).

Bunlar arasında, başlıca üç konuda farklılıklar ortaya çıkmaktadır. Bunlardan ilki, CPM'in geleneksel olarak faaliyetlere, PERT'in ise olaylara yönelik olarak planlanmasıdır. İkinci önemli fark, faaliyet zamanlarının tahmini ile ilgilidir. PERT'de her faaliyet süresinin hesaplanmasında belirsizliğin etkilerini giderebilmek amacıyla iyimser, kötümser ve en olası olmak üzere üç zaman tahmini yapılmakta ve buradan bir ortalama zamanın "te = a+4m+b/6" formülü yardımıyla hesaplanmasına gidilmektedir. Burada, a; en iyimser zaman tahminini, b; en kötümser zaman tahminini ve m; en olası zaman tahmini ifade etmektedir. Buna karşılık, CPM'de normal zaman ve sıkışık zaman olmak üzere iki zaman hesaplaması yapılmaktadır. Diğer bir ifade ile PERT, faaliyet süreleri olasılıklı olduğu zaman, CPM ise faaliyet süreleri kesin belirli olduğu zaman kullanılmaktadır (Wolf ve Hauck, 1985). CPM ile PERT arasındaki farklılıklardan üçüncüsü ise, CPM'de planlama ve kontrol sürecine maliyet kavramının da alınması, zaman ve maliyet kavramları arasında bağlantı kurulmasıdır.

CPM, proje yönetiminde en kısa tamamlanma süresinin belirlenmesinde kullanılan ve belirli bir faaliyeti hızlandırmanın katlanılacak maliyete değer olup olmadığını gösteren bir yöntemdir. Faaliyetlerin çoğu, ek eleman veya ek zaman kullanılması yoluyla hızlandırılabilir. Ancak, eğer projenin bütünü için gereken ölçüde toplam zaman kısaltılamıyorsa bunun için ek bir harcama yapmak akılcı olmayacaktır. Kritik yolda en düşük hızlandırma maliyetine sahip olduğu görülen faaliyetin kısaltılması yönünde karar alınır.

PERT tekniğinde maliyete ilişkin bilgilerin analize girmemesine karşın, bu tekniğin geliştirilmiş bir türü olan PERT /MALİYET tekniğinde faaliyetlerin birbirini izleyen parçalara bölünmesi gibi, tüm proje maliyeti de parçalara ayrılarak ilgili faaliyetlere dağıtılır. Böylelikle, maliyet kontrolü daha etkin bir biçimde uygulanabilir.

Ağ analizi yöntemleri, bir projenin mevcut kaynaklarla ve öngörülen zamanda tamamlanıp tamamlanamayacağını ve eğer tamamlanamıyorsa, daha fazla kaynağın nerelerde kullanılmasının projeye en büyük katkısı sağlayacağını belirlemeyi önemli ölçüde kolaylaştırır.

Bu yöntemle, hangi faaliyetlerin eş zamanlı olarak yürütülebileceği ve her faaliyetin ne zaman başlaması gerektiği açıkça görülebilir. Bu, gerek

insan gücünün ve gerekse makinelerin daha iyi kullanılmasına olanak sağlar. Ayrıca, projenin devam ettiği süre içinde herhangi bir sapmanın belirlenmesini ve nelerin yapılması gerektiğinin yönetime rapor edilmesini de kolaylaştırır.

#### **2.2.4.3.Ağ Analizi Kullanılma Esasları**

Projelerin genelde, geleceğe dönük ve benzeri olmayan çabalar olması risk ve belirsizlik faktörünü ön plana çıkarmaktadır. Bu nedenle, CPM ve PERT gibi ağ analizi tekniklerinin kullanılması bazı sorunlar yaratmaktadır. Bu sorunları ortadan kaldırmak ve başarılı bir uygulama olanağı sağlamak üzere temel bazı ilkelerin göz önünde bulundurulması gerekir:

- Proje ağ diyagramında, ana çizelgenin özelliklerini ve ağırlığını koruması sağlanmalıdır.
- Tüm ilişkiler ve dönüm noktası oluşturan olaylar kapsanmalıdır.
- Ağ diyagramı projenin tüm yapısal unsurlarını yansıtacak şekilde çizilmelidir.
- Tüm özel görevlerin ağda gösterilmesi sağlanmalıdır.
- Görev çizelgelerinde veya kısa ara çizelgelerde belirtilmesi daha uygun olan ayrıntılardan kaçınılmalıdır.
- Müşteriden veya başka dış kaynaklardan gelen yükümlülükler ve sınırlamalar plan kapsamına alınmalıdır.
- Proje ağ diyagramı; ana proje çizelgesini geçerli kılmak, önemini artırmak ve iyileştirme yöntemlerini belirlemek amacıyla kullanılmalıdır. Ana çizelge, yürürlükteki planı yansıtacak şekilde otomatik olarak değil, yalnızca uygun yönetim kararları ile değiştirilmelidir
- Bilgi işlem sistemleri desteği gerekli olduğunda, programlar dikkatle seçilmelidir. Esnek ve uygun olmayan programlar, projenin gecikmesine ve maliyetlerin artmasına neden olurlar.
- Proje ağ planının analizi ile elde edilen sonuçlar, üst yönetimin incelemesi için sabit zaman ölçeği üzerinde gösterilmelidir. Bilgi işlem kullanımı, yalnızca ekonomik ve yapılabilir olduğunda tercih edilmelidir.

Tüm ilgililerin ağ analizini anlamalarını, benimsemelerini ve etkin olarak kullanabilmelerini sağlamak üzere uygun eğitimler verilmelidir.

#### 2.2.4.4.Ağ Analizi ile Sağlanabilecek Faydalar

Ağ diyagramı kullanan planlama ve kritik yol analizi tekniklerinin proje yönetiminde kullanılması önemli yararlar sağlar. Doğru uygulanması durumunda, bunlardan beklenebilecek yararların başlıcaları şunlardır (Battersby 1970):

- Bir projenin temel unsurlarına ayrılması, projeye yeni bir bakış açısı getirir ve projenin daha iyi anlaşılmasına olanak sağlar. Bir ağ diyagramının çizilebilmesi için öncelikle çeşitli görevlerin ve faaliyetlerin mantıksal olarak birbirleriyle nasıl etkileşim içinde oldukları ya da olmaları gerektiği ortaya konulmalıdır. Bu da projenin tam ve doğru olarak anlaşılmasını kolaylaştırır.
- Bir ağ üzerinde kritik yolun belirlenmesi, yürütülecek faaliyetlerin zaman içerisinde programlanmasında önem ve öncelik verilecek olanları ortaya çıkarır. Proje hazırlamanın önemli bir yönünü oluşturan bu çalışma, projenin tamamlanması için gerekli süre hakkında bilgi sağlar. Ayrıca, gecikmeleri projenin tamamlanma süresini etkileyemeyen yani kritik olmayan faaliyetlerin de belirlenmesine yardımcı olur.
- PM, projeyi tamamlamak için gerekli toplam sürenin azaltılması olanakları ile bulunabilecek alternatif yolların beraberinde getireceği ek maliyetleri birlikte inceleme olanağını sağlar.
- Programlamada, basit olasılık dağılımları yardımıyla belirsizlik faktörünün de dikkate alınması sağlanabilir. Böylelikle, projenin yaklaşık olarak ne zaman tamamlanabileceğine ilişkin sorular cevaplandırılabilir.
- Ağ diyagramı, sınırlı miktarlarda bulunan belirli kaynakların en uygun biçimde nasıl dağıtılacağına planlanmasına yardımcı olur.
- Diyagramdan aynı zamanda, kaynakların tam ihtiyaç duyulduğu zamanda - ne çok erken ne de çok geç - bulundurulmasını sağlamak amacıyla da yararlanılır. Ağ analizinin en sık kullanıldığı alanlardan biri de budur.
- Projenin gelişmesinin izlenmesi amacıyla faaliyetlerin beklenen zamanlar ile gerçekleşme için fiilen geçen zamanlar karşılaştırılabilir. Bu, ağ diyagramı ile bütünleştirilen GANTT şemaları yardımıyla yapılır ve oluşabilecek darboğazların daha başlangıçta belirlenmesini kolaylaştırır.



- Ağ analizi kullanımı, finansal planlamaya da büyük ölçüde yardımcı olur. Zaman içerisinde farklı noktalarda ve alternatif programlarda nakit ihtiyacının ne kadar olacağı ağ yardımıyla incelenebilir.

- Teknik, statik değildir ve değişiklikler olduğu anda bu diyagrama yansıtılabilir. Gerçekten, ağ analizini, proje hazırlamanın belirli bir aşamasında çizilen basit bir diyagram ya da programın ne kadar gerisinde kaldığını izlemek için ara sıra bakılan bir tablo olarak görmemek gerekir. Bu teknik, projenin hazırlanmasında, değerlendirilmesinde ve yürütülmesinde yardımcı olacak bilgileri sağlayan, projenin planlanmasının ve kontrolünün tüm yönlerini kolaylaştıran bir tekniktir.

### **2.2.5. Proje Planlama Diyagramları (PPD)**

Proje yönetiminde ağ analiz tekniklerinin bazı durumlarda uygun olmamasının ve benimsenememesinin başlıca iki nedeni bulunmaktadır. Bunlardan birincisi, sembollerin yeterince esnek olmaması ve kolay kullanılamamasıdır. İkinci neden ise planlama sürecinin güçlüğü ve yüksek maliyeti olmaktadır. Özellikle, belirsizlik ve risk unsurları ağır basan, yeni teknoloji veya ürün geliştirme projelerinde, başarı olasılığını yükseltecek bir planlama isteniyorsa çok sayıda alternatif yolun dikkate alınması gerekecektir. Bir taraftan, alternatiflerin çokluğu ve birbirlerine yakınlığı, diğer taraftan planlamaya harcanacak zaman ve çabanın sınırlılığı, bu sürecin verimliliğini artıracak basit, açık ve hızlı işleyen bir planlama tekniğine gerek doğurmaktadır. Aynı şekilde, kullanılan sayısal tekniklerin de doğru kullanılabilir, basit teknikler olması gerekmektedir. Bu açıdan "Proje Planlama Diyagramlarının" bu ihtiyaçlara cevap veren açık ve basit bir proje tanımlama ve analiz yöntemi olduğu söylenebilir (Barutçugil, 2008:191).

Bilgi işlem sistemleri ile programlanabilen bir akış diyagrama dayanan PPD, yazılması ve okunması basit bir tekniktir. Diyagram, dikey olması nedeniyle daha az yer gerektirir ve açıklıkla görülür (Aslında bu yönüyle şekilsel bir algoritmadır). Yöneticinin planının gerçekte ne olduğunu daha net görmesine ve açık düşünmesine yardımcı olur. Proje yöneticisi ile fonksiyonel yöneticiler arasındaki iletişimi ve proje ekibinin üyelerinin kendi aralarındaki işbirliğini kolaylaştırır. Yöneticinin, projenin malzeme, teçhizat ve insan gücü ihtiyaçlarını genel hatlarıyla görmesine de olanak sağlar.

Bu yöntemin kullanıldığı projelerde başarılı bir planlama ve kontrol tekniği olduğunu kanıtladığı öne sürülmektedir. Bir proje planlama diyagramı daha önceleri en az birkaç sayfa tanımlama gerektiren karmaşık planları tek bir sayfada toplayabilmekte ve özellikle karmaşıklığın ve belirsizliğin egemen olduğu projelerin aşamalarını yöneticilere daha basit ve açık bir biçimde gösterebilmektedir.

#### **2.2.6. Kayma Şemaları**

İşletmelerde ele alınan projeler, genellikle öngörülen maliyetlerde ve planlanan zamanda gerçekleştirilememektedir. Bir yandan maliyetlerin yükselmesi, diğer yandan projenin zamanında bitirilememesi nedeniyle faydaların azalması, işletmeler açısından olumsuz sonuçları olan ortak bir sorundur. Bütün projelerde maliyetlerin, zamanın ve sağlanacak faydaların önceden kesin olarak tahmin edilmesi mümkün değildir. Bu nedenle, değişen belirsizlik koşulları altında, mümkün olan en iyi sonucu verecek bir proje kontrol sisteminin geliştirilmesi büyük önem taşır (Barutçugil, 2008:194-196).

Küçük veya büyük projelerin planlanması ve yürütülmesinde yönetimin yararlanabileceği çok sayıda yöntem bulunmakla beraber, bunların bazıları uygulamada yetersiz kalmaktadır. Örneğin, GANTT Şeması ve Ağ Analizi teknikleri ile belirsizlik koşulları altındaki projelerin gözden geçirilmeleri ve güncelleştirilmeleri önemli güçlükler yaratmaktadır. Ayrıca bu teknikler, genel yönetim amaçları için gerekenden daha ayrıntılı olmaları nedeniyle, projenin belirli bir yönünün planlanması ve izlenmesi bakımından bir değer taşımamakta, genel eylemlerin gerektirdiği bakış açısını sağlamamaktadır.

Gerçekte gerek duyulan, basit, hızlı, günün koşullarına uydurulabilen ve yönetim amaçları açısından uygun ve açık bilgi sağlayan bir sistemdir. Bu düşünce ile geliştirilen Kayma Şemaları, bu özellikleri taşıyan bir yöntemdir. Bu yöntemde proje, sonuçları açıkça tanımlanabilen çeşitli aşamalara bölünür. Örneğin, bir projede bu aşamalar; ilk tasarımların çizilmesi, malzemenin alımı, ürünün yapımı ve teslim aşamaları olabilir. Bu aşamaların sayısı, projenin niteliğine ve izlenmek istenen ayrıntı düzeyine göre değişecektir.

Projenin aşamaları, öngörülen tamamlama tarihlerine göre şemanın üst eksenine yazılır. Düzenli aralıklarla yapılan gözden geçirmelerde her aşamanın gerçekleşme durumu belirlenir. Öngörülen programa uygun süren aşamaların gelişme çizgileri dik hatların üzerinde kalacaktır. Dik hatlardan

öne veya geriye kayan çizgiler, projenin o aşamasının öngörülen zamandan geride veya ilerde olduğunu gösterecektir. İlk tahminlerin yapılması ve gelişmelerin işaretlenerek gösterilmesi, proje yöneticisinin sorumluluğundadır.

Bu yöntemin benzer ve daha basit bir uygulaması, dikey ekseninde planlanan bitiş zamanının veya projenin tamamlanması için gerek duyulan çabaların tahmininin gösterildiği ve yatay ekseninde de geçen zamanın veya harcanan toplam çabaların belirtildiği iki boyutlu basit grafiklerle yapılan proje kontrolleridir.

Proje başlangıcında, harcanan bir zaman ve çaba olmadığından dikey eksen üzerinde bir noktadan grafik çizimi başlar. Proje ilerledikçe gelişmeyi gösteren çizgiler, aşağıya, yatay eksene doğru inmeye başlar. Yatay eksene ulaşıldığında proje (veya projenin belirlenen bir aşaması) tamamlanmış olmaktadır.

Projenin plana uygun bir gelişme göstermesi durumunda, grafik dikey eksenden yatay eksene 45°lik bir doğru ile inecektir. 45°lik doğrudan sapmalar, proje için daha fazla veya daha az çaba harcandığını ya da projenin planlanan zamandan önde veya geride olduğunu gösterecektir. Bir gelişme çizgisinin ısrarla belirli bir yönde sapma göstermesi durumunda, bu çizgiyi 45° lik doğru üzerine getirmeye çalışmak yerine planı (veya tahmini) gözden geçirmek daha doğru olacaktır.

### **2.2.7. Proje Yönetim Tekniklerindeki Son Gelişmeler**

Her projenin kendine has bir doğası vardır. Bu sebeple de her projenin karakteristiği farklılıklar arz etmektedir. Bu durum projelerin standardize etme süreçlerini de büyük oranda engellemekte aynı zamanda da proje planlama ve izleme safhalarını olumsuz etkilemektedir. Bu sebeple proje yönetiminde planlama, programlama, izleme ve kontrol alanında kullanılan teknikler araştırmacılar ve kullanıcılar tarafından sürekli geliştirilmekte ve yenilenmektedir. Bu bölümde proje yönetim teknikleri alanında son yıllarda yapılmış, önemli görülen araştırmalardan ve gelişmelerden bahsedilecektir.

#### **2.2.7.1.CPM/PERT Tekniklerinin Sınırlılıkları**

CPM ve PERT teknikleri proje planlama ve izlemede yaygın olarak kullanılan, en kullanışlı programlama teknikleridir. Ancak bu tekniklerin sınırlılıkları bilinmektedir. Farklı endüstrilerin değişen gereksinimlerinin sonucu olarak bu

tekniklerin hesaplama etkinliklerinin artırılması amacıyla arařtırmalar yapılmakta ve diđer tekniklerin kullanımının artırılması yönünde çalışmalar devam etmektedir. Bunun yanında proje planlamalarında Doğrusal Programlama Teknikleri, Simülasyon Teknikleri, Genetik Algoritmalar gibi yöntemlerin kullanılması için arařtırmalar yapılmaktadır.

Fischer ve Aalami (1996) mevcut programlama tekniklerinin, maliyet tahminleri dinamik olarak proje programı ile ilişkilendirilemediğini bir sınırlılık olarak kabul etmiştir. Bu sınırlılığı ortadan kaldırmak için projelerin tasarım bilgilerini, maliyet ve plan şekline bilgisayar destekli ve dinamik olarak dönüřtüren bütünlüřtirici bir model geliřtirmişlerdir.

Cotrell (1999) PERT yönteminde kullanılan faaliyet süreleri için gerekli 3 tahmini deęeri 2'ye düşüren ve PERT'in basit bir sürümü olan bir teknik geliřtirmiş ve başarıyla test etmiştir. Bu teknikte PERT'ten farklı olarak Beta dağılımı yerine Normal dağılım kullanmıştır. Gerekli olan iki tahmin ise "en muhtemel süre" ve "kötümser süre" tahminidir.

Dawson ve Dawson (1998), PERT gibi standart planlama tekniklerinin görev sürelerinde belirsizlik olan projeler için yeterli olmadığını ifade ederek, mevcut yazılım araçlarının belirsizlikleri yönetecek şekilde geliřtirilmelerini, öğrenimi ve kullanımı kolay hale getirilmelerini gerektiğini önermişlerdir,

#### **2.2.7.2.Proje Programlama**

Üretim ve inřaat projelerinde kullanışlı ve yüksek kaliteli bir plan geliřtirmenin önündeki en önemli engel genellikle proje planının başlangıç taslağını oluřturma safhasıdır. Yine de geçmiş tecrübe ve deneyimlere dayanarak ve risk eğilimleriyle benzer projeleri göz önünde bulundurarak, herhangi bir proje (inřaat, ARGE) için tahmini bir zaman planlaması yapmak mümkündür. Bu plan tam olarak doğru olmayabilir. Ancak toplam proje süresi hakkında tatmin edici bir fikir verebilir (Ahuja ve Thiruvengadam, 2004).

Chavallier ve Russel (2001), sınırlı girdilere baęlı bir proje taslak planı oluřturmak için esnek tabanlı uzman bir sistemin proje yönetim sistemindeki önceden tanımlı, standart bilgi kalıpları ile birleřtirilebileceğini göstermişlerdir.

#### **2.2.7.3.Faaliyet Sürelerindeki Belirsizlikler**

Ben-Haim ve Laufer (1998) faaliyetlerin devam ettięi süreçte yařanan belirsizliklerden olumsuz etkilenen proje planının güvenilirliğini geliřtirmek için yeni

bir kavram (concept) ortaya koymuşlardır. Bu kavramın uygulanması için gerekli olan teknik, kullanılması basit bir teknik olup minimum düzeyde bilgi gerektirmekte, öznel bilgilere ihtiyaç duyulmakta ve arzu edilen güvenilirlik düzeyinde bir proje planının hazırlanmasına yardımcı olmaktadır. Bu tekniğin kullanıldığı belirgin örnekler şu şekildedir,

- Proje planının güvenilirliğini hesaplamak,
- Proje planının güvenilirliğini arttırmak,
- Proje güvenilirliğini azaltmadan proje sürecini kısaltmak,
- Proje sürecindeki faaliyetlerin proje güvenilirliğini nasıl etkilediğini tespit etmek.

Mishra (1989) proje sürecindeki istatistiksel dağılım parametreleri ve bu parametrelerin kabul edilme düzeylerini belirlemek için bir "Bulanık Set Modeli (Fuzzy Set Model)" geliştirmiştir. Bu model faaliyet sürelerinin analizi için gerekli öznel terimleri dikkate almak ve eylemsel ifadeleri sayısal ifadelere dönüştürmeye yardımcı olmak için uygun bir yaklaşım olarak değerlendirilmektedir.

Wang ve Demsetz (2000) proje faaliyetleri arasındaki ilişkiyi gösteren korelasyon düzeylerine bağlı olarak plan ağlarını değerlendiren ve daha gelişmiş bir model olan "İlişkili Belirsizlik Etkisindeki Ağlar (Networks Under Correlated Uncertainty-NETCOR)" modelini ortaya koymuşlar ve bu modelin mevcut bir proje üstünde pratik bir uygulamasını göstermişlerdir. Aynı girdiler kullanılarak, korelasyon ilişkilerinin ele alınmadığı PERT ve Simülasyon Analizleri de yapılmıştır. Sonuçlar karşılaştırıldığında proje analizinde korelasyon ilişkilerini göz önünde bulundurmanın önemli olduğu ortaya çıkmıştır (Chavallier ve Russel, 2001),

#### **2.2.7.4.Zaman Maliyet Optimizasyonu**

Zaman ve maliyet proje kontratlarının iki temel unsurudur. Bu unsurlar aynı zamanda proje yönetiminin iki ana hedefini teşkil etmektedir. Sahoo (1996), proje planlama ve izleme sürecinde kullanılabilir zaman ve maliyet unsurlarını analiz eden dinamik programlama yaklaşımını geliştirmiştir. Lee ve Yi (1999), inşaat projelerinde ki zaman ve maliyet veri setlerini bütünleştiren ve bu iki unsur arasındaki uyumsuzlukları tespit eden sayısal bir yaklaşım geliştirmişlerdir. O'brain ve Fischer (2000), tedarikçi ve alt yüklenicilerin bakış açılarını dikkate alarak ağ planlama yönteminin altındaki varsayımlara dayalı uygulamaya yönelik bir kritik yöntem geliştirmişlerdir. Que (2002), gerçek ve güncel problemleri, zaman-maliyet optimizasyonuna göre çözüme ulaştıran bir Genetik Algoritma yaklaşımı geliştirmiştir.

#### **2.2.7.5.Kaynakların Tahsisi**

İnşaat projeleri benzer faaliyetlerin tekrar edildiği çok sayıda birim içermektedir. Bu durum benzer veya tekrarlanan faaliyetlerde kaynakların kesintisiz olarak kullanımını temin etmeyi gerektirmektedir. CPM'in benzer faaliyetlerdeki kaynak tahsisinin etkili bir şekilde gerçekleştirilmesini sağlayamadığını ifade eden Harris ve Ionnau (1998). CPM'in teknik kısıtlarını dikkate alan ve aynı zamanda kaynak tahsisi kısıtlarının sürekli kullanılmasını sağlayan "Tekrarlı Planlama Metodu (Repetitive Scheduling Method-RSM)"nu geliştirmişlerdir.

#### **2.2.7.6.Yönetim**

Projenin uygulanması safhasında proje yöneticileri mevcut plan, eldeki kaynaklar ve diğer faktörleri göz önünde bulundurarak projeyi etkili bir şekilde yönetmek durumundadırlar. Faniran (1998) proje planının uygulanmasını gerçekleştirebilmek için gerekli stratejilerin belirlenmesine yönelik bir çerçeve yöntem ortaya koymuştur. Proje planı ve anlık çevresel faktörler arasındaki ilişkiyi regresyon analizi ile incelemiştir. Choo (1999) sistematik haftalık iş planlarının oluşturulması amacıyla "iş Planı" adı verilen bir veri tabanı programı geliştirmiştir.

Eldin (1997) proje geliştirme araçlarının sayısını ve bunların faydalarını, başarı faktörlerini ve uygulama sınırlılıklarını tanımlayan bir literatür taraması yapmıştır. Amacı proje maliyetini arttırmadan toplam proje süresini kısaltmaya yarayan araçları bulmaktır. Belirlenen kriteri karşılayan proje süresini kısaltıcı araç olarak "Eşzamanlı Mühendislik" seçilmiştir. Yaptığı analiz sonucunda eşzamanlı

mühendislik yönteminin uygulanması ile proje sürecinin %25 azaltılabileceği görülmüştür. Çalışma, başarı faktörleri, uygulama sınırlılıkları ve alınan dersler açısından proje yönetimi ile ilgili genelleyici, faydalı bilgiler içermektedir.

#### **2.2.7.7.Proje İzleme ve Kontrol**

Projeler, uygulama safhasına geçildiğinde planlanan ile gerçekleştirilen işlemler arasında sürekli olarak bir karşılaştırma yapılır. Önceden öngörülemeyen çeşitli sebeplerden dolayı projeler planlandığı gibi ilerlemeyebilir. Bu nedenle yöneticiler proje işlemlerinin işleyişini sürekli ve doğru bir şekilde takip etmek ve gelecekte ortaya çıkabilecek ihtiyaçları önceden tahmin etmek durumundadırlar.

Bu alanda en geçerli yöntemlerden birisi Kazanılmış Değer Analizi-KOA (Earned Value Analysis)'dir. KDA, yapılan faaliyetleri, maliyetlerini dikkate alarak parasal anlamda ifade etmektedir. Bu sebeple, diğer tekniklere kıyasla performans değerlendirmeleri daha açıklayıcı, gerçekçi ve anlaşılır olmaktadır. (Ayrıntılı bilgi için bkz. EK-5)

Shi (2001) faaliyetlerdeki gecikmeleri ve bu gecikmelerin projeye etkilerini hesaplayan bir yöntem ortaya koymuştur. Shi (2001 )'nin önerdiği yöntem temel olarak planlanan ve gerçekleştirilen faaliyetlerin karşılaştırılmasından ibarettir.

## ÜÇÜNCÜ BÖLÜM

### YAZILIM PROJE YÖNETİMİ

#### 3.1. BİLGİ SİSTEMLERİ

Günümüzde bir sisteme giren verilerin sistem elemanları tarafından anlaşılabilir şekilde derlenmesi, elleçlenmesi ve uyarlanması faaliyetleri, her biri değişik işlevler üstlenmiş birbirine girift (birinin çıktısı diğerinin girdisi olan) münferit yazılımlar ya da birbirlerine bir yönetim modülü ile bağlantılı modüler yapıdaki yazılımlardan oluşan sistemler tarafından gerçekleştirildiğinden bilgi sistemlerinin yazılım tabanlı olarak çalıştığı söylenebilir.

Bilgi sistemleri; belli bir amaç için bir araya gelmiş yapılarda kontrol ve grup içi koordinasyonun sağlanması ve karar verici için karara esas teşkil edecek verilerin/bilgilerin toplanması, işlenmesi, depolanması ve dağıtımı için oluşturulan (genellikle yazılım) birbirine girift birimsel/ilişkisel elemanlar (yazılım modülleri) kümesidir. Bütün bu işlevlerinin yanında işletmelerde bilgi sistem araçları (yazılımları) mevcut/muhtemel problem/hata analizi, yeni ürün geliştirilmesi ve karmaşık işlevlerin gerçekleştirilmesi gibi konularda hem çalışanlara hem de yöneticilere destek sağlamaktadır. Bilgi sistem işlevlerini özetleyen ve Bacon ve Fitzgerald (Bacon ve Fitzgerald, 2001:46-67) tarafından da üzerinde durulan bilgi sistemi alanları için sistematik bir altyapı modeli mevcuttur. İşletmelerin elde ettikleri verilerin işletme açısından anlamlı hale gelebilmesi için işletmenin ihtiyaç, beklenti ve değerlendirmelerine göre belli çözümlenme/süzme/ değerlendirme aşamalarından geçmeleri gerekmektedir. Bunu da günümüzde en iyi sağlayabilecek sistemler kuşkusuz bilgi sistemleridir. Başka bir deyişle; işletmelerin çevresinde gerçekleşen olguları temsil eden/sisteme giren bilgiler, ancak bilgi sisteminde işlendikten sonra anlamlı yani kullanılabilir bilgilere dönüşmektedir (Kalıpsız, Buharalı ve Biricik, 2006; 25). Soyuer (2000) ise işletmelerde bilgi sistemlerini, yönetimin ihtiyaç duyduğu veri ve bilgileri işletmenin makro ve mikro çevresinden toplayan, organize eden, özetleyen ve ihtiyaç duyan alt sistemlere ve yöneticilere raporlayan sistemlerdir şeklinde tanımlamıştır. Bu amaçlar doğrultusunda işletmelerde bilgi saklama, işleme ve dağıtımı için kullanılan donanım, yazılım, iletişim ve veritabanı yönetimi gibi bilgi işleme dayalı teknolojiler bütününe de bilgi teknolojileri denir (Post ve Anderso, 2003; 507-543).

İşletmelerde bilgi sistemlerinin üstlendiği pek çok görev vardır ve en temel olarak işletmelerin faaliyetleri için bilgi bütünleştirilmesi (Gulledge, 2006:5-20)



sağlayan bu sistemlerin görevleri/kullanım maksatları;

- Maliyetlerin düşürülmesi ve etkinliğin artırılması sayesinde verimliliğin artırılması,
- Kalitenin artırılması,
- Rekabet üstünlüğünün sağlanması,
- Firma stratejilerinin uygulanması,
- İş etüdü,
- Daha iyi ve doğru kararlar alınabilmesi,
- Müşteri ihtiyaçlarına daha kısa zamanda cevap verebilmesi ve gerekli geliştirmelerin yapılabilmesi,
- Değerli bilgiye hızlı erişimin sağlanması,
- Yaratıcılığın ve yeniliklerin geliştirilmesi şeklinde özetlenebilir.

İşletmeler yukarıda ifade edilen hedeflere ulaşmak için farklı seviyelerde bilgi sistemlerine [İşlemsel Bilgi Sistemleri (Transaction Processing Systems, TPS), Yönetim Bilgi Sistemleri (Management Information Systems, MIS), Ofis Otomasyon Sistemleri (Office Systems, OS), Karar Destek Sistemleri (Decision Support Systems, DSS), Üst Yönetici Bilgi Sistemleri (Executive Information Systems, EIS) ve Bilgi Tabanlı İş Sistemleri (Knowledge-based Work Systems, KWS)] ihtiyaç duyarlar (Kalıpsız, Buharalı ve Biricik, 2006; 25-26). Organizasyonda kullanıldıkları seviye bakımından bilgi sistemlerine ilişkin farklılıklar Şekil- 3.1'de gösterilmektedir.

| Stratejik Seviye Sistemleri                   |                               |                        |                    |                     | A |   |
|---|-------------------------------|------------------------|--------------------|---------------------|---|---|
| 5 Yıllık Satış                                | 5 Yıllık Plan                 | 5 Yıllık Bütçe         | Kâr Planlama       | İşgücü Planlaması   |   |   |
| Yönetim Seviyesi Sistemleri                   |                               |                        |                    |                     | B |   |
| Satış Yönetimi                                | Envanter Kontrolü             | Yıllık Maliyet Analizi | Yatırım Analizi    | Fiyat / Kar Analizi |   |   |
| Satış Bölge Analizi                           | Üretim Programlama            |                        |                    |                     |   |   |
| Bilgi Seviyesi Sistemleri                     |                               |                        |                    |                     | C | D |
| Bilgisayar Destekli Simülasyon Tasarım/Üretim | Simülasyon                    |                        | Sanal Gerçeklik    |                     |   |   |
| Kelime İşlem                                  | Doküman Yönetimi              |                        | Elektronik Takvim  |                     |   |   |
| İşlemsel Seviye Sistemleri                    |                               |                        |                    |                     | E | F |
| Sipariş                                       | Makine Kontrolü               |                        | Bordro Yönetimi    | Tazminat            |   |   |
|   | Tesis/Fabrika Programlama     |                        | Borçlar Muhasebesi | Eğitim& Geliştirme  |   |   |
| Sipariş                                       | Hammadde Hareketleri Kontrolü | Nakit Yönetimi         | Alacak Muhasebesi  | İşçi Kaydı Tutma    |   |   |
| Satış ve Pazarlama                            | Üretim                        | Finans                 | Muhasebe           | İnsan Kaynakları    |   |   |

A: Üst yönetici Bilgi Sistemi(ÜBS); B: Yönetim Bilgi Sistemi(YBS); C: Karar Destek Sistemleri(KDS); D: Bilgi Tabanlı İş Sistemleri(BTİS); E: Ofis Otomasyon Sistemleri(OOS); F: İşlemsel Bilgi Sistemleri(İBS)

**Şekil-3.1:** Bilgi sistemi türleri ve organizasyonlardaki bilgi sistemleri (Loudon, 2004)

Yukarıda açıklanan bilgi sistem işlevlerinin zaman içinde tek bir sistem altında toplanması gündeme gelmiş ve günümüzde Kurumsal Kaynak Planlaması (KKP) olarak bilinen sistemler ortaya çıkmıştır. İşletmelerde KKP denildiğinde, tüm işletme birimlerinde karşılıklı etkileşim sayesinde bilgi toplama, işleme, analiz, raporlama ve değerlendirmesini sağlayan yazılım uygulamaları akla gelmektedir. Şekil-3.1'de ifade edilen alt bilgi sistemlerine ilişkin özet bilgi müteakip maddelerde sunulmuştur.

### **3.1.1. İşlemsel Bilgi Sistemleri (Yazılımları) (İBS)**

İBS, genel olarak organizasyon içerisindeki günlük işlemler sonucunda elde edilecek verileri ve tutulacak kayıtları bir araya getirmek, elleçlemek, ve her kademedeki yönetici istek ve ihtiyaçlarını dikkate alarak kullanıma hazır duruma getirmek amacıyla geliştirilmiştir. İşletmede Satış ve Pazarlama, Üretim, Finans, Muhasebe ve İnsan Kaynakları olmak üzere bölümler içinde sürekli olarak gerçekleşen işlemler işlemsel bilgi sistemi tarafından kaydedilir ve anlamlı bilgilere dönüştürülür. İBS ile; organizasyonla ilgili maaş, sigorta, vergi, çek, tahsilat ve stok işlemlerinin kayıtlanması yanında manuel, uzun zaman ve yoğun çaba gerektiren hesaplamaların bilgisayar yazılımları ile sağlanacak bir otomasyon sayesinde daha kısa sürede, daha doğru ve en az maliyetle yapılması amaçlanmıştır. (Anameriç, 2005; 22-23).

İBS'nin faaliyet kayıtlaması (siparişler, alındılar, ödemeler, makbuzlar, faturalar, sipariş kayıtları, materyal istek fişleri vb. işlemler), raporların değerlendirmesi (belirlenmiş aralıklara göre hazırlanan satın alma, ödünç verme, personel faaliyetleri vb. işlemler) ve sorgulama (stok kontrol ve tarama sonuçları vb. işlemler) gibi üç ana bölümü bulunmaktadır. (Lucey, 1987; 185-186).

Özetle İBS sürecinde önceden belirlenmiş bilgi, yine önceden belirlenmiş kurallar çerçevesinde uygun standartlarda üretilir ve organizasyon içerisinde ilgili birimlere gönderilir (Ferguson ve Whitelaw, 1992;185). İBS, organizasyon içerisinde verilerin daha hızlı işlenmesi ve iletilmesinin yanında hataların azaltılması, saklanan veri ve bilginin güvenliğinin sağlanması, terminaller aracılığıyla veri ve bilgiye ulaşılması, doğruluk ve güvenilirlik oranının artırılması, işlemlerin yapıldığı anda kayıtlara geçirilmesi ve analizlerin gerçekleştirilmesinde önemli avantajlar sağlamaktadır.

### **3.1.2. Ofis Otomasyon Sistemleri (Yazılımları) (OOS)**

OOS, kelime işleme, tablolama, grafik çizimi paket programları, elektronik posta, faks, telekonferans gibi bilgisayar tabanlı iletişim sistemleri, kişisel

bilgisayarlar ve büyük yer kaplayan belgelerin saklanması ve kolay geri çağırılması için mikro grafik birimlerini kapsamaktadır (Kalıpsız, Buharalı ve Biricik, 2006; 25).

### **3.1.3. Bilgi Tabanlı İş Sistemleri (Yazılımları) (BTİS)**

Bilgi tabanlı iş sistemleri, belirli alanlarda uzmanlaşmış kişilere hitap eden, bilimsel tasarım, bilgi kazanma ve bu yeni bilgiyi iş akışı ile bütünleştiren sistemlerdir. Bu sisteme verilebilecek en iyi örneklerden biri olan bilgisayar destekli tasarım (Computer Aided Desing-CAD), araştırmacının yerini almadan sadece araştırmacının sonuçlarını bilgi sistemiyle birleştirerek daha verimli kullanılmasını sağlar (Kalıpsız, Buharalı ve Biricik, 2006; 30).

### **3.1.4. Karar Destek Sistemleri (Yazılımları) (KDS)**

Karar destek sistemleri, karar verme sürecinde ihtiyaç duyulan verilere ulaşılması, verilerin değerlendirilmesi ve analizi, karara esas olacak veri süzme faaliyetlerinin ve hesaplamalarının yapılması, karar vericiye verilecek kararda yeteri kadar alternatifi içerecek öngörü ve değerlendirmelerin sunulmasını sağlarlar. Model odaklı veya veri odaklı olarak var olabilen bu sistemler tüm kurumu çevreleyen geniş kullanıcı gruplarını destekleyen ağ bağlantılı veri bankaları şeklinde düzenlenmiş olabileceği gibi, sınırlı sayıda kullanıcının kullanım konusunda yetkilendirildiği karmaşık ve eğitim gerektiren özel yazılımlar da olabilirler (Christopher ve Philip, 1991).

İşletmelerin bu bilgi sistemlerini kullanmasıyla daha analitik kararlar vermesi sonucu kârlılıklarının yükselmesi, kârlılıklarının daha kararlı olması, karar süresinin kısılması ve daha fazla alternatifin değerlendirilmesi gibi önemli yararlar sağlanmaktadır (Chou, Tripuramallu ve Chou, 2005).

Kullanıcıların etkili karar almalarında yüksek oranda başarı sağlayan karar destek sistem yazılım araçları, finansal analizden ne eğer (what if) analizine, optimizasyondan olasılık değerlendirmelerine, grafik gösterimden tahminlemeye, her türlü istatistikî analizden hedef arama ve rapor üretimine kadar değişik bileşenlere sahip olabilmektedir.

### **3.1.5. Yönetim Bilgi Sistemleri (Yazılımları) (YBS)**

YBS, işletmelerde farklı yönetim seviyelerinde bulunan yöneticilerin yönetsel faaliyetler kapsamında ihtiyaç duydukları bilgilere istenilen format ya da şekilde ulaşabilmelerine imkân tanıyan genellikle otomasyona dayalı olarak çalışan bir komuta kontrol ve destek sistemidir.

YBS, yönetim konularında destek sağlayan bir sistem olması yanında,

işletmelerin mevcut faaliyetlerinin planlanması, takibi ve kontrolü sayesinde gelecekteki performanslarının tahminine imkân veren standart, rutin ve özlü raporların hazırlanmasını ve sunulmasını sağlamaktadır. Çevre ve işletme dışı faaliyetlerden daha çok işletme içi faaliyetler üzerinde odaklanan bu sistemler, yönetim düzeyindeki planlama, kontrol ve karar verme fonksiyonlarını desteklemekte ve ihtiyaç duydukları verileri diğer ticari işlem sistemlerinden almaktadırlar (Elibol, 2005; 158).

Teknolojik gelişmelere paralel olarak uygulama alanlarını artırmış olan YBS'nin organizasyonlara uygulanması aslında sadece 30-35 yıl öncesine dayanmaktadır. Sürekli güncellemeler sayesinde bilgilerin elde edilmesi, işlenmesi, depolanması ve iletiminde sağladığı kolaylıklarla bilgiden en üst düzeyde yararlanılmasını sağlar (Anameriç, 2005; 24).

YBS'nin aslî işlevi, karara esas teşkil edecek nitelik ve özellikteki bilgilerin istenilen ve arzu edilen doğruluk derecesinde organizasyon yönetim ekibine sunulması olduğundan, üretilen bilgilerin özetlenmesi, tablolanması, daha işe yarar bir şekilde sunulması ve kolay ulaşılabilir olması önem arz etmektedir.

### **3.1.6. Üst Yönetim Bilgi Sistemleri (Yazılımları) (ÜBS)**

Yapılandırılmamış sorunlarda üst yönetimin ihtiyacı olan bilgi üzerinde yoğunlaşan bu sistemler, aslında üst yönetimin işletmenin stratejik amaçlarının gerçekleştirilmesi kapsamında ihtiyaç duydukları bilgilere kolay ve hızlı ulaşmalarını sağlamaktadırlar.

Bu sistemler, işlemsel bilgi sistemleri ve yönetim bilgi sistemlerince üretilen sonuç veriler, finans verileri, ofis sistemleri, modelleme analiz verileri, borsa, uzun vadeli satış tahminleri, planlar, bütçe planları, kâr planları ve işgücü planları gibi stratejik kararları etkileyecek grafik, ara yüz, rapor ve ekran seçenekleri sunmaktadır. (Haag, Cummings ve Dawkins 1998:197)

İşletme içi ve dışı verilerin filtrelendiği, özetlendiği, sınıflandırıldığı ve formülize edildiği işlemler sunan bu sistemler, ürettikleri nihaî veriler sayesinde yöneticilere bilgisayar uzmanlığı gereksinimi duymadan, üst yönetim bilgi sistemini kolayca kullanarak raporlarda bulamadıkları ayrıntılara anında ve çizelge veya grafik şeklinde ulaşmalarını sağlar (McLeod ve Schell, 2001).

## **3.2. YAZILIM PROJELERİ**

Problem çözme ve bilgi edinme tarihin başlangıcından beri insanoğlunun önem verdiği bir olgu olmuş, bilgiyi saklamak, işlemek ve nakletmek amacıyla pek çok yöntem ve araç üretilmiştir. 20'nci Yüzyılın ortalarında insanoğlu yarı iletken

teknolojilerin sağladığı imkânlarla her türlü bilgiyi 0 ve 1'ler cinsinden ifade edebileceğini fark etmiştir. Bu dönüşümle önce bilginin saklanabileceği, işlenebileceği ve iletebileceği ortak bir elektronik ortamı, müteakiben bir bilim dalı ve nihayetinde de geniş bir yelpazeye yayılan bilişim teknolojilerini (Information Technologies, IT) oluşturmayı başarmıştır. IT, asıl gelişimini izleyen 20 yıl içinde gerçekleştirmiş ve toplumun her kesimince kullanılan vazgeçilmez bir araç haline dönüşmüştür. Günümüzde bu teknolojiler, değişik özelliklerde tasarlanabilmekte; bilimsel araştırmalarda, üretim ve servis sektöründe, eğitimde, haberleşmede, sağlıkta, savunmada, eğlencede, ekonomik ve sosyal planlamada, kısaca; hayatın her kesitinde, oldukça yaygın olarak kullanılmaktadır.

IT çerçevesinde "YAZILIM" olarak adlandırdığımız olgu ise, bilgisayara ne yapması gerektiğini bildiren kurallar manzumesi olarak tanımlanabilir. Yazılım olmadan kullandığınız donanımın hiçbir anlamı olmayacaktır. Günlük hayatta kullandığımız hesap makinelerinde bile 1000 satır kod olduğunu düşündüğümüzde, yazılımın neleri kapsadığı ve ne anlama geldiği çok daha kolay anlaşılacaktır.

Yazılımların büyüklüğünün ve karmaşıklığının giderek artması, yazılım geliştirilmesi ve tedarikinde süreçlerin kontrolünün her zamankinden daha yoğun gerçekleştirilmesi gereğini ortaya koymaktadır.

Bu da yazılım geliştirme faaliyetlerinin sistematik ve maliyet etkin olarak gerçekleştirilmesi gerekliliğini vurgulamaktadır. Hemen her alana girmiş olan yazılım ve yazılım sistemlerinin geliştirilmesi, projelendirilmesi ve gerçekleştirilmesi özelliklerine binaen detaylandırılmış süreçler içermektedir. Bu kapsamda tezin izleyen bölümlerinde önce yazılım proje ve özellikleri üzerinde durulacak, yazılım projelerinde yapılan hatalar vurgulanacak, yazılım projelerinde eleman seçimine ilişkin önemli hususlar irdelenecek ve yazılım proje yönetimine ilişkin kapsamlı bilgiler sunulacaktır.

### **3.2.1. Yazılım Projeleri ve Özellikleri**

Yazılım projeleri, bilişim teknolojisinin sağladığı imkânlar sayesinde kurum ve işletmelerin gerçekleştirdikleri faaliyetlerin daha hızlı, etkin ve verimli hale getirilmesini amaçlamaktadır. Örneğin, bir bankanın şube işlemlerini otomasyona geçirmesi bir yazılım projesidir. Burada bankanın amacı, elle (manuel) yürüttüğü işleri, bilişim teknolojisi kullanarak gerçekleştirmek ve bunun getireceği üstünlüklerden yararlanmaktır.

Yirminci Yüzyılın ikinci yarısında gündeme gelen yazılım projeleri, iş süreçleri ile bilişim teknolojisinin bütünleştirilmesini amaçlamaktadır. Özellikle 80'lerin ikinci yarısından itibaren bilişim sektörünün gelişmesiyle proje yönetimi ile ilgili en etkili ve yetkili kuruluş olan PMI'nın (Project Management Institute, www.pmi.org) yıllık büyüme hızı %20-%30 arasında olduğu bilinmektedir. PMI, bugün dünya çapında 250 şubesi (PMI Chapter) ve 200.000 üyesi bulunan ve proje yönetimi konusunda tüm dünyada tanınan PMP (Project Management Professional) sertifikaları veren bir enstitüdür. Bu sertifika, PMI'nın yılların tecrübesi ve birikimiyle oluşturduğu bir metodolojiye dayanmaktadır. PMI projeleri 6 temel başlık altına almakta, ağırlıklandırmakta ve Tablo-3.1'deki gibi detaylandırmaktadır.

**Tablo-3.1: Temel Proje Aşamaları**

**Ağırlıklandırılmış ve Detaylandırılmış Temel Proje Aşamaları**

| Proje Başlatma (%8,5)           | Proje Planlama (%23,5)  | Proje Uygulama (%23,5)                     | Proje Kontrol (%23)                           | Proje Sonlandırma (%7)                      | Profesyonel Sorumluluklar (%14,5)      |
|---------------------------------|---|--|---|---|--|
| Proje hedeflerinin tanımlanması | Proje isteklerinin belirlenmesi                               | Kaynakları görevlendirme                   | Performans Ölçümü                             | Teslimatların Kabulü                        | Güvenilirlik-Doğruluk                  |
| Teslimatların tanımlanması      | İş Dağılım Yapıları (Work Breakdown Structure-WBS), oluşturma | Uygulamalar                                | Kontrol limitlerinin belirlenmesi             | Öğrenilen Derslerin yazılı hale getirilmesi | Bilgi birikimine katkıda bulunmak      |
| Süreç çıktıları                 | Kaynak Yönetim Planı Geliştirme                               | Süreçlerin yönetilmesi                     | Düzeltilici faaliyetler                       | Kapama işlemleri                            | Profesyonel hayata karşı sorumluluklar |
| Kısıtların Dokümantasyonu       | Zaman- Maliyet tahminleri geliştirme                          | İletişim                                   | Düzeltilici faaliyet etkinliğinin incelenmesi | Ürün bilgilerinin kaydedilmesi              | Paydaşların ihtiyaçlarını dengelemek   |
| Varsayımların Dokümantasyonu    | Proje kontrol sisteminin kurulması                            | Kalite güvence prosedürlerinin uygulanması | Planın doğruluğundan emin olma                | Kaynakların dağıtılması                     | Farklılıklara saygı duyma              |
| Stratejinin tanımlanması        | Proje planı geliştirme  |  | Kontrol planlarının yeniden değerlendirilmesi |   |  |
| Performans kriter tanımlama     | Plan onayının alınması  |  | Risklere karşı yanıt oluşturma                |   |  |
| Kaynak ihtiyacı tanımlanması    |   |  | Proje faaliyetlerini takip etme               |   |  |
| Bütçenin tanımlanması           |   |  |   |   |  |
| Resmi Dokümantasyon Süreci      |   |  |   |   |  |

(Davenport, 1998:121-131)

Tablo-3.1'den de görüldüğü üzere, çok fazla finansman gerektiren bilişim/yazılım projelerinde planlama, uygulama ve kontrol en fazla paya sahip ve en önemli olan adımlardır. Projelerin bu detayda başlatılması, uygulanması ve sonlandırılması disiplinler arası çalışabilme becerisi ve süreçlere hâkim olmayı gerektirir, aksi takdirde başarısızlık kaçınılmazdır (Davenport, 1998:121-131).

Yazılım projeleri, diğer projelerle pek çok ortak özelliğe sahip olmakla birlikte, aşağıdaki noktalarda bunlardan farklılıklar göstermektedir:

- Proje yönetimine özgü araç ve teknikler çok fazla denenmemiş, hatta oluşumlarını dahi tamamlamamıştır.
- Gelişen teknolojiye paralel olarak yeni teknik olanaklar ortaya çıktıkça, yazılım projelerinin hedefleri de proje sürecinde hızlı değişimler göstermektedir. Değişen hedeflere kolayca uyum sağlamayı gerektiren proje yönetimi ihtiyacı bilişim/yazılım projelerine özgüdür.
- Proje bitirilip, ürün işler hale geçmeden, projenin başarısı sağlıklı olarak ölçülememektedir. Örneğin, bir bina inşaatının yapımı her aşamasında gözlenip, ilerlemesinin ölçülmesine karşılık, benzer yöntemler kullanılarak iyi yürütüldüğü sanılan bir yazılım projesinin aslında başarısız olduğu son dakikaya kadar anlaşılabilir.
- Yazılım projelerinde ürünün tanımlanması kişilere ve kişisel beklentilere bağlı olduğundan ve kişiden kişiye değiştiğinden projenin tanımlanması ve tanımlandığı şekilde tamamlanması zorlaşır.
- Ortaya koyduğu ürünler ve karşılması beklenen ihtiyaçlar projenin bitiminden sonra da değişebilir. Buna tasarım ve gerçekleştirme aşamasında dikkat edilmelidir.
- Yazılım projeleri sonucunda genellikle ilgili kurumda bir örgüt değişikliği gerekir. Projelerin planlanması ve yürütülmesi aşamasında, bu tür bir örgüt değişikliğinin hayata geçirilmesinde karşılaşılabilecek sorunlar dikkatle irdelenmeli ve çözümler bulunmalıdır.

### **3.2.2. Yazılım Projelerinin Problemleri**

Yazılımlar, günümüz tedarik projelerinin en riskli bölümü olarak kabul edildiğinden, tedarik süreçlerinde en fazla üzerine eğilinilen ve üzerinde çalışılması gereken unsur olarak karşımıza çıkmaktadır.

Washington Post gazetesinde yayınlanan, 55 havacılık ve savunma yüklenicisini kapsayan bir araştırmada, firmalardan %75'inin onaylı proje programlarını aştıklarının belirlenmesi bunu kanıtlar niteliktedir. Benzer şekilde ABD'de; yazılım içermeyen projelerin onaylanmış proje takvimlerini ortalama 4-7 ay

aştığı, yazılım içerikli projelerde ise durumun ortalama 20 ayı bulduğu belirlenmiştir (Bkz. Şekil-3.2).



**Şekil-3.2:** Yazılım içeren ve içermeyen sistemler arasındaki süre aşımı oranları (İşlek, 2000:6, STSC-US, 2000:32)

Bu duruma benzer bir şekilde, DACS (Data & Analyses Center for Software)'ın Roma Laboratuvarı tarafından yayınlanan rapor sonuçları Tablo-3.2'de sunulmuştur.

**Tablo-3.2:** DACS ROMA Rapor Sonuçları

| Başarı Faktörü                      | Sonuç  |
|-------------------------------------|--------|
| Yazılım programlarının tamamlanması | %25    |
| Hataların geç belirlenmesi          | %15    |
| Tekrar işleme için harcanan zaman   | %30-40 |
| Takvim aşımaları                    | %50    |

Bazı yazılım problemleri örnekleri aşağıda sunulmuştur. (Bkz.Tablo-3.3)

**Tablo-3.3:** Yazılım Problemleri Örnekleri

| YIL  | PROJE                                   | SONUÇLAR                            |
|------|---|-------------------------------------|
| 1980 | ITT 4 Anahtar Sistemi                   | 500 milyon dolar kayıp, Vazgeçildi, |
| 1987 | California Department of Motor Vehicles | 30 milyon dolar kayıp, Vazgeçildi,  |



| YIL  | PROJE   | SONUÇLAR   |
|------|---|--|
| 1989 | Washington Eyaleti Otomatik Sosyal Servis Sistemi | Kullanıcı ihtiyaçlarını karşılamaktan uzak, 20 milyon dolar kayıp, Vazgeçildi, |
| 1992 | Amerikan Havayolları Uçuş Sistemi                 | 165 milyon dolar kayıp, Vazgeçildi,  |

Yine bununla ilgili olarak Capers JONES'un yaptığı bir araştırmaya göre;

- Ticari ürünlerin yaklaşık %10' u oldukça başarılı,
- %20' si başarılı,
- %40' ı sınırdadır,
- %30' u ise başarısızlıkla sonuçlanmaktadır.

### 3.2.3. Yazılım Projelerinin Başarısızlıkla Sonuçlanma Nedenleri

#### 3.2.3.1. Yönetimin Sorumluluğu

Yapılan araştırmalar, yazılım geliştirme süreçlerinde karşılaşılan başarısız sonuçların en önemli sebeplerinden birinin yönetim olduğunu göstermektedir. Yönetim problemi, aynı zamanda insan problemi anlamına gelmektedir. Bazı yönetim problemi örnekleri aşağıda sunulmuştur:

- Yazılımın tasarlanması, kodlanması, test edilmesi ya da kalite yönetimi faaliyetlerinde yer alan personelin yeteneklerinin kısıtlılığı veya belirli bir sistemin geliştirilmesi için oluşturulan geliştirme grubunun olması gerektiği gibi organize edilmemesi,
- İhtiyaçların tam olarak anlaşılabilmesi ve dolayısıyla da geliştirme süreçlerinin yanlış tasarlanması..

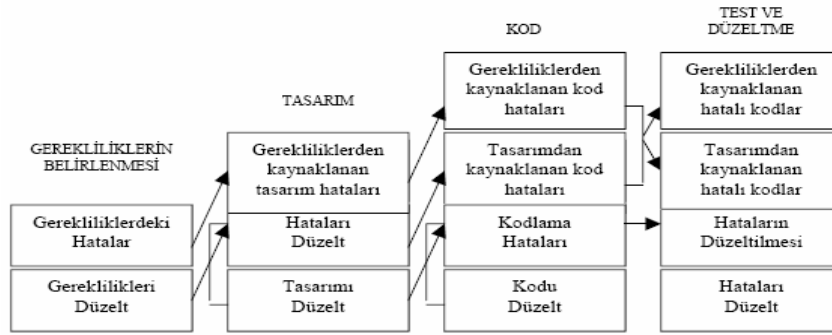
#### 3.2.3.2. İhtiyaçların Değişmesi

Yazılımların geliştirilmesi sırasında karşılaşılan olaylardan en tipik olanı, proje sürecinde, başlangıçta belirlenmiş olan ihtiyaçların, daha sonra değiştirilmesidir. Söz konusu değişiklikler, donanım tedarikinde olduğundan çok daha kötü sonuçlara yol açabilmektedir. Yazılım, geliştirme sürecinde kullanılan yöntemlere bağlı olarak, gerekliliklerin spesifikasyonu "Tasarım-Kod Üretilmesi-Test-Sunum" şeklinde bir süreci izlemektedir. Kod üretilmesi aşamasında müşteri tarafından ihtiyaçlarda yapılacak küçük bir değişiklik, söz konusu sürecin tekrar

baştan başlatılması anlamına gelebilmektedir. Bu durum bazen, bütün sürecin sıfırdan başlatılmasına da yol açabilmektedir.

### 3.2.3.3. Yazılım İhtiyaçlarının Tam Olarak Anlaşılması

Yazılımların geliştirilmesinde de karşılaşılan temel problem, belirlenmiş olan ihtiyaçların tam olarak analiz edilmemesi ve dolayısıyla da yanlış anlaşılmasıdır. İhtiyaçların anlaşılması, sonuçta, daha yüksek maliyetler, proje programının aşılması ya da kalitesiz ürüne yol açmaktadır. Şekil-3.3, bu durumun sonuçlarını göstermektedir:



Şekil-3.3: Hataların birikimli maliyetleri (İşlek, 2000:8)

### 3.2.3.4. Yazılımın Karmaşıklığı

Yazılımlar, donanım üretim sürecinde birçok hususun otomasyonunu sağlarken, kendisi hâlâ elle üretildiğinden insan hatasına en açık üretim sürecini oluşturmaktadır. Bununla birlikte, donanım üretiminde karşılaşılabilecek bir hatanın telafisi mümkün iken ya da problemin çözümü için sınırlı sayıda yöntem mevcut ve bunlar arasında seçim yapmak kolay iken, yazılım için aynı şeyleri söylemek mümkün değildir. Yazılım üretiminde karşılaşılan bir problemin sonsuz sayıda çözüm yöntemi olabilmektedir. Dolayısıyla da en doğru yöntemi bulmak, başlı başına bir problem olmaktadır.

Gelişen teknoloji ve bununla birlikte büyüyen ihtiyaçlar doğrultusunda üretilen ya da üretilmesi gereken yazılımların büyüklüğü ve karmaşıklığı da o ölçüde artmaktadır. Yazılımın kapsamının büyümesi; yanlış anlamaları, hataları, maliyetleri ve başarısızlık olasılığını yükseltmektedir. Tablo-3.4, terminolojideki büyüklük kavramını açıklamaktadır.

**Tablo-3.4:** Proje Büyüklükleri ve Başarı

| Program Büyüklüğü | İnsan sayısı | Programın Süresi | Başarı Olasılığı    |
|-------------------|--------------|------------------|---------------------|
| Küçük             | <10          | 3-6 ay           | Yüksek              |
| Orta              | 20-30        | 1-2 yıl          | Ortalama            |
| Büyük             | 100-300      | 3-5 yıl          | Ortalamanın Altında |
| Çok Büyük         | 1000-2000    | 7-10 yıl         | Çok düşük           |

(İşlek, 2000:9, STSC-US, 2000:50)

Yazılımın büyüklüğü teknik problemler yanında çok ciddi yönetim problemlerine de neden olmaktadır. Yazılımın büyük olması, geliştirme sürecinde yer alacak personel sayısının fazla olması, dolayısıyla da çok daha yoğun yönetim anlamına gelmektedir. Yönetim faaliyetlerinin yoğunlaşması ise personel arası iletişimi zayıflatmakta, bu da geliştirme sürecine çok ciddi darbeler vurabilmektedir.

### **3.2.3.5. Eksik Tahminler**

Yazılım sistemlerinin belirlenmiş maliyetleri ve proje programlarını aşmasının ve sonuçta istenilen ürünün ortaya çıkarılmamasının temel nedenlerinden birisi de, geliştirme süreçleri ile ilgili olarak gerçekçi tahminlerin yapılamamasıdır. Yazılımın sadece kod yazmaktan ibaret olduğunu düşünmek, eksik tahminler yapmanın başlangıcıdır. Geliştirilen yazılımın test ve muayenesi ve hatalarının ayıklanması için haracanan miktarlar proje toplam maliyetinin yaklaşık %60'ını oluşturmaktadır.

Yazılımın doğası (yapısı) gereği, büyüklüğü ve karmaşıklığı konusunda doğru bir tahmin yapmak çok zor olmaktadır. Donanımın aksine yazılım, ne elle tutabilir, ne ağırlığı ölçülebilir, ne de paketlenabilir bir yapıya sahip olduğundan, yazılım tedarikinde maliyetleri ve proje kapatımına ilişkin zamansal tahminleri de doğrudan etkilemektedir.

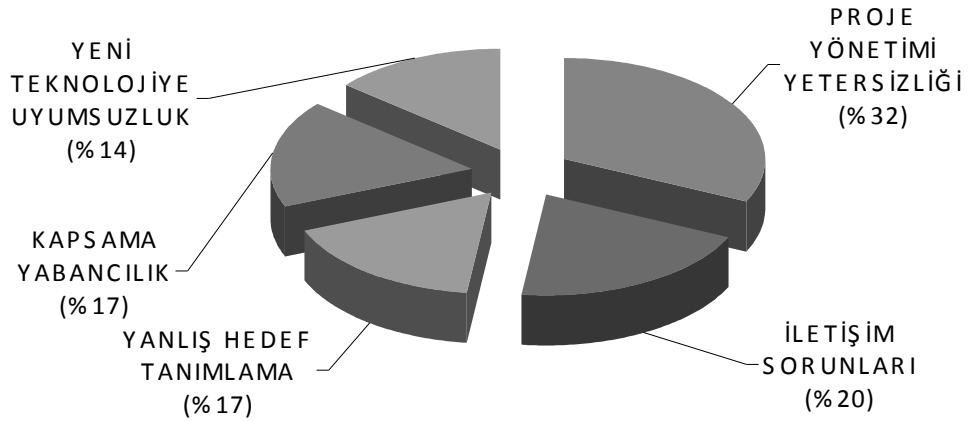
### **3.2.3.6. Yazılım Projelerinde Sıkça Yapılan Hatalar**

Hatalar gruplanacak olunursa;

- **Stratejik Plan Eksikliği:** Hata firma yönetimden kaynaklanmaktadır. Firmanın elinde bir stratejik planı bulunmamakta, bilişim projesine ihtiyaçlardan çok trendi yakalamak için girmektedirler. Stratejik planları olmadığından zamana karşı firmanın yol haritasında olması gereken paftayı takip edemezler. Stratejik plan olmayınca firma taktiksel seviye ile yönetilmeye çalışılır ancak operasyondaki eksikler yüzünden taktik hareketlerde günlük operasyonları kontrol etmeye çalışmaktan öteye gidemez.

- Proje Planlaması: Operasyonlar sorun yumağıdır, süreç haritası belli olmayan firma kara düzen çalışmaktadır.
- Proje Elemanlarının Seçimi: Bu aşamadan sonra danışman firma ve firma içi proje yöneticisi atanır. Bu proje yöneticisi bilişim kökenli ise proje süreç yapılandırma, maliyet düşürme ve karlılığı arttırmaktan öte tamamen teknolojik bir hal almaktadır. Amaç teknolojiyi uygulamaktır, proje yönetiminin performans ölçümü kriterleri bir yana atılıp tamamen teknolojik entegrasyon ön plana çıkmaktadır [Ross, 1999:65- 68].
- Proje Uygulaması: Firmanın asıl odağı kaymış asıl uğraş alanı bu projeyi hayata geçirmek olmuştur. Böyle bir başarısızlığın mali yükü bazen firmanın kapanmasına sebep olabilir (Davenport, 1998:121-131).
- Proje Kontrolü: Projenin istenilen özelliklerde ve arzu edilen seviyede geliştirilebilmesi için proje süresince alınan bütün tedbirler bu kapsamda değerlendirilmelidir.

Şekil-3.4'ten de gördüğümüz üzere başarısızlığın ana faktörü proje yönetimi yetersizliğidir. Proje yönetimi kapsamında; yazılım donanım seçimi, eleman seçimi, yanlış görevlendirme, maliyetleri takip edememe, teknolojiye gömülme vb. önemli hususlardır ve hata sebepleridir. Daha detaylı analizler projeler bazında (ERP, CRM vb.) incelenebilir.



Şekil-3.4: Projelerde Başarısızlık Nedenleri

Yazılım projelerine has durumları başarısızlık nedenlerini sıralayacak olursak; [(Düzakın ve Sevinç, 2002:189-218), (www.microsoft.com/turkiye/dynamics/erp/erp\_secimi.msp, www.netsis.com.tr, 21.06.2009)]

- Kurumun sorunlarının ve gelecekle ilgili beklentilerinin yetersiz analizi,
- Entegrasyonu sağlayan firmanın ve ürünün doğru seçilmemesi,
- Entegrasyonu sağlayan firmanın çözümle ilgi destek, araç ve yöntemlerinin yetersizliği,
- Kullanılacak uygulamanın gerekliliği ve pozitif propagandasının yeterli olarak yapılmaması,
- Projenin analiz, kurulum ve uygulama süreçlerinde doğru görevlendirmelerin yapılmaması,
- Kurum ve personelin eski iş yapma yöntemlerinden vazgeçmekte direnmesi,
- Personelin isteksizliği veya eğitim düzeyinin düşüklüğü, Sisteme veri girişlerinin yetersiz, hatalı, düzensiz olması ve sürekli olmaması.

#### **3.2.4. Yazılım Projelerinde Doğru Eleman Seçimi ve Yazılım Proje Ekibinin Oluşumu ve Gelişimi Aşamaları**

Bu bölüm EK-4'te açıklanmıştır.

#### **3.2.5. Yazılım Proje Geliştirme ve Yönetim Süreci**

Taraflar arasında ikilem yaratabilen yazılım proje geliştirme sürecinde karşılaşılan büyük problemlerin küçük parçalara bölünerek çözümlenmesi esastır. Ancak, çözümlenen bu parçaların yeniden bir araya getirilmesi son derece karmaşık olabilmektedir. Özellikle kaynak planlamasına dayanan sistemler söz konusu olduğunda işletme içerisindeki her birimin etkileşim alanları ile ilişkilendirilmesi ve bilgi akış durumlarının ortaya konulması bahse konu süreç kapsamında son derece zor bir aşamadır. Parçalar farklı birim sorumluluğunda çözümlendiğinden birleşik bir çözümün oluşturulması ve parçacıkların bütünleştirilmesi uzun zaman alabilmektedir. Bu zorlu ve zaman alan sürecin karmaşıklığı çoğu yazılım sistemi geliştirme projesinin başarısızlıkla sonuçlanmasına ya da iptaline neden olmuştur (Bass, Clements ve Kazman, 2003: 96).

Projenin tamamlanmasını müteakip gerçekleştirilen yeni yazılımların uygulanması kritikliğine binaen önem arz eden bir konudur ve işletme içindeki tüm

birimlerin ortak ve koordineli olarak çalışmasını zorunlu kılar. Sistemin uygulanması aşamasında karşılaşılabilecek muhtemel temel sorunların anlaşılmasında durum çalışmaları (case studies) oldukça önemli araçlardır. Bu aşamada işletmeler, yeni uygulamadan vazgeçmelerine neden olabilecek büyük sorunlarla karşılaşabilmektedir. Yazılım projelerinin geliştirilmesi;

- Her zaman aşırı maliyet ve zaman alıcı olarak görülmesi,
- Yanlış kurgulanmış/düzenlenmiş olabilmemesinin veya beklentileri ne ölçüde karşılayabileceğinin tahmin edilememesinin kaygı yaratması,
- Eskiden yeniye geçişin getireceği bocalama döneminin yaratacağı olumsuzluklardan çekinilmesi nedenleriyle pek rağbet görmemektedir. Geliştirme sürecinde karşılaşılan bu zorlukların kontrol edilerek süreci etkin bir şekilde yönetebilecek bazı yöntemler geliştirilmiştir.

### **3.2.6. Yazılım Geliştirme Model, Metodoloji ve Teknikleri**

Bu bölüm EK-5'te sunulmuştur.

### **3.2.7. Yazılım Proje Geliştirme ve Yönetim Süreç, Aşama ve Adımları**

Bu bölüm EK-2'de sunulmuştur.

### **3.2.8. Yazılım Proje Yönetimini Destekleyen Disiplinler**

#### **3.2.8.1. Değişiklik Yönetimi**

Tüm proje geliştirme ve uygulama süreci boyunca proje sahibinden proje ekibinden ya da kontrol edilemeyen dış etkenlerden çeşitli değişiklik isteklerinin gelmesi kaçınılmazdır. Değişiklik yönetimi; projenin gerçekleştirilmesi sırasında oluşabilecek bir değişiklik isteğinin, sistemin tümünün ya da elemanlarının işlevsel ya da fiziksel özelliklerindeki herhangi bir değişimin değerlendirilmesi ve proje kapsamı ile uyumunun gözetilerek yerine getirilmesidir. Proje ile ilgili değişiklikler farklı yöntemlerle kontrol altına alınabilir.

Bu kapsamda en çok kullanılan yöntem bir değişiklik kontrol sisteminin oluşturulmasıdır. Bu sistem çerçevesinde, proje belgelerinde yapılması gereken herhangi bir değişikliğin kural ve işlemleri belirlenmeli ve uygulanmalıdır. Bu plan ve işlemler kurumun politikasıyla da tutarlı olmalıdır. Değişiklik kontrol sisteminin başarılı olabilmesi için su koşulların sağlanması son derece önemlidir (Harrington, Conner ve Horney, 2000):

- Değişiklik isteği açık olarak tanımlanmalı ve dayandığı nedenler belgelenmelidir,
- Söz konusu değişikliğin proje çıktılarına muhtemel etkileri incelenmelidir,
- Değişiklik isteği projeye ilgili tüm tarafların yetkilileri tarafından onaylanmalıdır,
- Onaylanan tüm değişiklikler proje planına eksiksiz olarak aktarılarak projenin bütünlüğü korunmalıdır,
- Eğer projenin en son çıktısı olan ürün üzerinde bir değişiklik yapılmışsa bu değişiklikler projenin tüm süreçlerine de yansıtılmalıdır, .
- Değişikliklerin maliyet, risk, kalite ve personel unsurları üzerindeki etkisi dikkate alınmalıdır. Bu işlevlerin yerine getirilmesi sorumluluğu proje yöneticisine aittir. Ancak, değişiklik isteklerini onaylamak ve reddetmekten sorumlu bir Değişiklik Kontrol Kurulu kurulması önerilir. Çok büyük ve karmaşık projelerde, farklı konulardaki değişiklik isteklerini değerlendirmek üzere farklı Değişiklik Kontrol Kurulları oluşturulabilir. Ancak bu kurulların yalnızca ilgili üyelerin bulunduğu, küçük ve hızlı karar alacak yapıda olmasına dikkat edilmeli, sorunları büyütüp çözümsüzlüğe götüren kurul modelinden kaçınılmalıdır. Acil durumlarda kurul kararına gerek kalmadan işlemlerin otomatik olarak yerine getirilmesi için de gerekli kurallar belirlenmelidir.

### **3.2.8.2. Yazılım Konfigürasyon Yönetimi (Software Configuration Management - SCM)**

Yazılım mühendisliğinde SCM, yazılımda meydana gelen değişikliklerin takip ve kontrolü maksadıyla gerçekleştirilen faaliyetler bütünüdür. Genel tanımı ile Konfigürasyon Yönetimi, bilişim teknolojisine ilişkin bir sistemin yapısı içinde bulunan tüm elemanları kontrol ve koordine etme tekniğidir. Burada sistem ile her tür donanım, yazılım ve bunların bileşenlerinden oluşan bilgi işlem kaynakları kastedilmektedir.

Bir sistem olarak düşünüldüğünde, sistemi oluşturan parçalardan her biri konfigürasyon yönetimine konu olan unsurlardır. Bunlar Konfigürasyon Elemanı olarak adlandırılırlar.

SCM, “Biri bir şey yaptı ama acaba bir başkası bunu nasıl yeniden yapabilir?” sorusuna cevap arar. Problem çoğunlukla yapılan işin/faaliyetin/hizmetin/ürünün

birebir yeniden üretilmesinden çok kontrol edilebilir artırımlı değişikliklerin nasıl gerçekleştirileceğine odaklanır. Bu nedenle soruya cevap vermek bu noktada farklı sonuçların karşılaştırılması ve farklılıkların analizi meselesine dönüşür. Geleneksel konfigürasyon yönetimi, tipik olarak oldukça basit ürünlerin kontrollü üretimi konusuna yoğunlaşmasına rağmen, şimdilerde SCM uygulayıcıları geliştirilmekte olan karmaşık bir sistem dâhilinde oldukça küçük değişikliklerin kendi kontrollerinde gerçekleştirilmesi problemiyle karşı karşıyadır.

SCM terminolojisinin geçmişi konusunda farklı görüşler bulunmaktadır. Roger Pressman, "Yazılım Mühendisliği: Uygulamacı Bir Yaklaşım (Software Engineering: A Practitioner's Approach)" isimli kitabında SCM'yi aşağıdaki şekilde tanımlamaktadır:

*"Değişmesi muhtemel iş ürünlerinin tanımlanması, aralarındaki ilişkilerin belirlenmesi, bu iş ürünlerinin farklı versiyonlarının yönetimi için mekanizmaların tanımlanması, yapılan değişikliklerin kontrol edilmesi, kayıtlanması ve raporlanması kapsamındaki değişikliklerin kontrollü için tasarlanmış faaliyetler dizisidir."*

SCM ile aşağıdaki amaçların gerçekleştirilmesi hedeflenmektedir:

- Konfigürasyonun tanımlanması ve kontrolü
- Bileşenlerin durumlarının kayıtlanması ve raporlanması
- Yazılım bileşenleri arasında bütünlük ve uyumluluk gözden geçirmesi
- Örgütün geliştirme sürecine bağlı kalıp kalmadığının değerlendirilmesi
- Sistem çevre birimlerinin (yazılım ve donanım) yönetimi
- Sürece bağlı olarak proje ekibi içindeki etkileşimin kolaylaştırılması
- Her türlü hataya karşı sistemin gözden geçirilmesi ve hata takibi

Bu kapsamda SCM dört temel işlevden oluşmaktadır.

- Kaynakların Belirlenmesi: Mevcut kaynakların belirlenmesi ve birbirleriyle olan ilişkilerinin tanımlanması,
- Konfigürasyon Değişiklik Kontrolü: Başlangıçta belirlenmiş yapılandırma elemanlarındaki değişikliklerin yalnızca bu yetkiye sahip kişilerce, belli standart ve prosedürlere uygun olarak yapılması,



- Durum Kayıtları: Kaynakların geçmiş, mevcut ve planlanan durumları ile ilgili bilgilerinin kaydedilmesi ve güncellenmesi,
- Doğrulama: Konfigürasyon elemanlarının mevcut durumlarının kayıtlardaki durumları ile tutarlılık içinde olduğunun denetlenmesi.

Konfigürasyon yönetimi, proje yöneticisinin sorumluluğunda olmakla birlikte, gerektiği takdirde projenin büyüklüğü ve kapsamına göre bu işten sorumlu olacak ve proje yöneticisine bağlı çalışacak ayrı bir Konfigürasyon Yöneticisi de belirlenebilir.

Konfigürasyon yönetimi;

- Her bir yazılım, donanım ögesinin sürecin hangi aşamasında hangi fiziksel, fonksiyonel özellikleri sağlaması gerektiği,
- Her bir ögenin konfigürasyonunun hangi noktada dondurulacağı,
- Konfigürasyonu dondurulmuş öğelerde değişiklik yapılması için hangi sürecin izleneceği,
- Bir ana hat (baseline) oluşturmak için hangi denetim ve gözden geçirmelerin yapılması gerektiği,
- Konfigürasyon öğelerinin konfigürasyonlarının doğru olarak izlenebilmesi için hangi araç ve programların hangi yöntemlerle kullanılacağı hususlarına cevap vermelidir.

### **3.2.8.3. Yazılım Risk Yönetimi**

Yazılım proje planı gelecek ile ilgili tahminlerle oluşturulduğundan, her zaman için belirsizlikler içerir. Parasal kaynaklarda, sözleşmede yer alan teslim sürelerinde ya da teknik ihtiyaçlardaki değişiklikler birer belirsizlik örneğidir. Doğal olarak, bu belirsizlikler potansiyel risk taşırlar. Başlangıçta öngörülemediği olan ve proje gidisini aksatabilecek olaylar da birer proje riskidir.

Riskin arttığı yerde pozisyonu korumak ya da iyileştirmek için çaba gerekmektedir. Riskler, proje gidisini ters yönde değiştirebilir, planı aksatabilir, proje ürününü etkileyebilir. Risk yönetimi bu gibi durumların tanımlanmasını, değerlendirilmesini, önlemek ya da etkisini azaltmak yönünde gerekli denetimlerin uygulanmasını ve alternatiflerin planlanmasını içerir.

Belli baslı risk türleri şunlardır:

- Zamanlama riski - yetersiz zaman ayrılması,

- Maliyet risk - maliyetin iyi hesaplanmaması,
- Teknik risk - ürünün oluşturulmasındaki zorluk derecesi,
- Operasyonel risk - Ürünün ihtiyaçları karşılayamaması,
- Tahmin edilemeyen riskler - Düzen değişiklikleri, doğal afetler, çevresel faktörler,
- Tahmin edilebilen (fakat belirsiz) durumlar - Pazar değişiklikleri, enflasyon, döviz kuru değişiklikleri, vergiler, vb.

Risk yönetimi, proje yönetiminin önemli bir parçasıdır. Proje yöneticisi, risk unsuru taşıyan bir durumla karşı karşıya oldukları zaman, bu durumun nasıl ve ne zaman etki yaratacağını görebilmelidir. Risk yönetimi, proje yöneticisinin daha iyi karar alabilmesini sağlayan bir araçtır. Risk yönetiminin iki ana parçası vardır:

- **Riskin Değerlendirilmesi**

Üç adımdan oluşur;

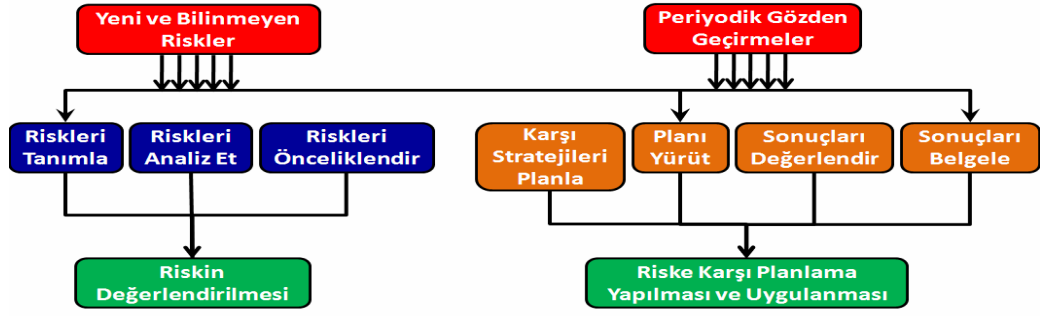
- Risklerin tanımlanması,
- Analiz edilmesi,
- Önceliklerin belirlenmesi.

- **Riske Karşı Planlama Yapılması**

Dört adımdan oluşur;

- Riskleri yok etmeye/azaltmaya yönelik planlama,
- Risk planının yürütülmesi,
- Proje sonuçlarının değerlendirilmesi
- Alınan önlemler ile proje sonuçlarının belgelenmesi.

Risk yönetiminin iki ana sürecinin birleştirilmesi ile oluşturulmuş yedi adımlık süreç Şekil-3.5'de sunulmuştur (Bknz. EK-3).



**Sekil-3.5:** Risk Yönetiminde "Yedi Ana Adım"

- **Risk Yönetimi Araç ve Teknikleri**

Nitelik ve nicelik açısından beş ana bölüme ayrılır:

- Fikir yaratma araçları
  - ◆ Uzmanlarla görüşme
  - ◆ Anket düzenleme
  - ◆ Grup içi fikir jimnastiği (Brainstorming)
- Eşgüdüm araçları
  - ◆ Planların gözden geçirilmesi
  - ◆ Teknik özellikleri okuma
  - ◆ Ekip toplantıları
- İnsan yönetimi araçları
  - ◆ Liderlik
  - ◆ Ekip-oluşturma becerileri
  - ◆ Zaman yönetimi
  - ◆ İlişki kurma becerileri
  - ◆ Yargı
- Karar verme araçları
  - ◆ Olasılık kavramları
  - ◆ Yasam-döngüsü maliyet analizleri
  - ◆ Uzman yazılımlar
  - ◆ Karlılık ölçümleri
- Planlama araçları
  - ◆ GANTT Çizgesi
  - ◆ PERT Çizgesi
  - ◆ Kritik Yol Yöntemi (CPU)
  - ◆ Kilometre Taşı (Milestone) Çizelgesi

## **DÖRDÜNCÜ BÖLÜM**

### **ÇOK MODLU ULAŞTIRMA SİSTEMİ**

#### **4.1. AMAÇ**

Bu çalışmada günümüzün ihtiyaçlarına cevap verebilecek nitelikte; bütün ulaştırma modlarını kullanabilen, gerektiğinde bu modları birleştirebilen (multimodal) veya arka arkaya belli bir sıra dâhilinde birbirlerine aktarılarak kullanılmasına imkân sağlayan (zincirleme ulaştırma, intermodal) ya da uygun durum ve ortamlarda birbiri içinde taşınmasına imkân veren (combined) maliyet etkin bir ulaştırma sistemi geliştirilmesi ve modellenmesi amaçlanmıştır. Modellenen sistem sayesinde kullanıcılara karar desteği verilmesi ve maliyet etkin taşıma planlaması yapılması sağlanacaktır.

#### **4.2. KAPSAM**

Sistem dâhilinde İç Su Yolu (Inland Water Way, IWW) taşımacılığı dâhil tüm ulaştırma modlarına yönelik planlama yapılması, planların programlanması, takip ve kontrolü mümkün olacaktır. Küçük ulaştırma faaliyetlerinden (tek araçlı, parçalı ve/veya kısa süreli kargo taşımacılığı) büyük bir tesisin (bir fabrika gibi) yer değiştirmesi kapsamında yapılacak taşımalara hatta sisteme ileride yapılacak ilavelerle askeri maksatlı taşımalara kadar geniş bir yelpazede planlama yapılması sağlanabilecektir. Sistemin aynı zamanda büyük bir kargo şirketinin bütün ulaştırma faaliyetlerinin planlanmasında ve icrasında da şirketin yapı ve özellikleri dikkate alınarak yapılacak küçük ilavelerle kullanılabileceği değerlendirilmektedir.

#### **4.3. KONSEPT**

Model; ulaştırma ve/veya lojistik hizmet sahalarında faaliyet yürüten, tüm yurt geneli, belli bölgeler ya da belli illerde konuşlanmış, geniş bir ulaştırma sistemi ve alt yapısına sahip ya da yaptığı anlaşmalarla etki ve ilgi alanı dâhilindeki bu sistem ve alt yapılara kolayca ve garantili erişim (assured access) sağlayabilen bir şirketin yaptığı/yapabileceği taşıma faaliyetlerine ilişkin değerlendirme yapılarak oluşturulmuştur.

Sistemin (Model) konsept olarak modüler ve esnek bir yapıda kurgulanması ile her bir modülün diğer modüllerden bağımsız olarak kullanılabilmesi ve ihtiyaç duyulması durumunda sistemin bütünü etkilenmeden yeni modül(ler) ilave edilmesi mümkün kılınmıştır. Modüller bütünlük tek bir veri tabanından beslenecek, bu veri tabanını güncelleyebilecek ve istendiğinde yapılan güncellemeler geri alınabilecektir.

Bu kapsamda sistemin ilk etapta;

- Taşımaya esas malzeme bilgileri ve ulaştırma alt yapısının planlanmasına ve coğrafi verilere yönelik bütünlük ortak bir veri tabanı (VTYM),
- Veri tabanını kullanarak maliyet etkin taşıma/kurye planlamasını, programlamasını, animasyonunu, simülasyonunu ve koordinasyonunu sağlayan, ulaştırma ihtiyaçlarının ve planlamalara ilişkin olası problem alanlarının belirlenmesinde kullanılan planlama modülleri (TPM, KURPM),
- Gerçek/gerçeğe yakın zamanlı olarak araç/malzemelerin yer, konum ve durumlarına ilişkin iz bilgilerini sunan (aktif ve pasif iz takibi) “etkin araç/malzeme iz takip modülü (ETARİTM)” olmak üzere dört ayrı modülden oluşması planlanmıştır. Modüllerin taşıyacağı özellikler EK-6’da (Sistem İhtiyaç Özellikleri Dokümanı) sunulmuştur.

#### **4.4. SİSTEM PLANLAMA ÖZELLİKLERİ**

Sistem işlevsel planlama yapısı bakımından ihtiyaç duydukları verileri (yer/konum/lokasyon verileri, ulaştırma araç verileri, ulaştırma alt yapı verileri, taşınacak birime ilişkin veriler) Veri Tabanı Yönetim Modülünden (VTYM) alan idari planlama ve lojistik (taşıma) planlama olmak üzere iki bölümden oluşmaktadır.

##### **4.4.1. İdari Planlama**

Bu bölümün amacı; şirketlerin operasyonel birimleri ile lojistik birimlerinin aynı dilde konuşmalarının sağlanması için idari bir sistem yapısı (köprü) kurulmasıdır.

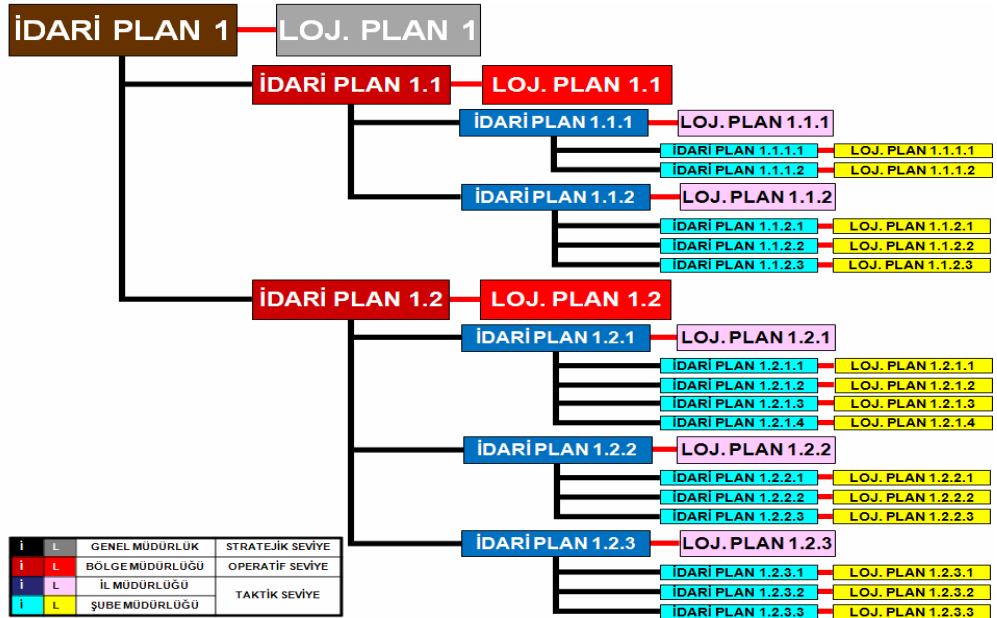
Operasyonel birimler yaptıkları faaliyet planlarında ayrıntıya girmeden icra edilecek görevin gerçekleştirilebilmesi için ihtiyaç duyulan ekip büyüklüklerini, bunların çıkış ve varış noktalarını, müşteri ihtiyacının karşılanması gereken en geç zamanı (Customer’s Required Date, CRD), müşteri ihtiyacına esas taşınacak birim (yolcu, malzeme, yük ekipman vb.) özelliklerini ortaya koyarlar. Lojistikçiler ise, işin ayrıntılandırılması, icra edilecek görevin özelliğine göre taşıma sisteminin yapılandırılması, mevcut imkânlar (sahip olunan taşıma araçları ve garantili erişim dâhilinde sağlanabilecek ulaştırma imkân ve kabiliyetleri) dâhilinde taşımaların gerçekleştirilmesi, bu imkânlar yeterli gelmediği durumlarda hizmet alımı yolu ile müşteri ihtiyacının her halükârda karşılanması hususları ile ilgilenirler.

Şirketlerin çoğunluğunun genel müdürlük, yurtdışı genel müdürlük, bölge müdürlüğü, il müdürlüğü ve şube şeklinde teşkilatlandıkları da dikkate alındığında sistematik ve hiyerarşik bir yapının kurgulanması gerekliliği ortaya çıkmaktadır. Bu kurgu da hiç şüphesiz farklı birimlerde görevli personelin aynı dili veya ortak kabul görebilecek bir dili/yaklaşımı kullanmalarının/izlemelerinin sağlanması ile

mümkündür. Çünkü hiyerarşik bir yapı içinde farklı birimlerce planlama yapılması sistem içinde planlama çatışmalarının ortaya çıkmasına neden olmaktadır.

Bu çalışmada bu noktadan hareketle olası çatışmaların önlenmesi ve ortak dil kullanımının sağlanması amacıyla yapılacak her türlü taşıma planının bir idari plana bağlanması öngörülmüştür. İdari plan operasyonel birimlerce hazırlanacak ve belirlenecek bir referans gün (Bknz. EK-1) dâhilinde hangi taşınacak birimin, nereden, nereye, ne kadar zamanda (CRD) hangi öncelik derecesine göre taşınması gerektiği hususlarını içerecektir. Bunu yaparken zamanın yeterli olduğu durumlarda tabandan tepeye ve zaman kısıtının bulunması durumunda da tepeden tabana doğru planlama yapılmasına imkân sağlayacak bir plan hiyerarşisi oluşturulacaktır. Bu sayede;

- İdari planların yapıldıkları seviyeye (genel müdürlük, yurtdışı genel müdürlük, bölge müdürlüğü, il müdürlüğü, şube) göre bir hiyerarşi ve plan ağacı şeklinde sıralanması sağlanacak,
- Herhangi bir planın hangi seviyede yapıldığı, bağlı olduğu bir üst plan bulunup bulunmadığı görülebilecek,
- Sonrasında idari planlarla ilişkilendirilecek lojistik planların da bu hiyerarşiyi takip etmesi sağlanmış olacaktır (Bknz. Şekil-4.1).



Şekil-4.1: Plan Hiyerarşisi

Her seviyede tek bir idari plan bulunurken aynı seviyede planlama yapılacak birden çok birim bulunabileceğinden bu birimlerin miktarı kadar taşıma planı bulunabilecektir. Bu da tek bir idari planın kendisi ile aynı seviyedeki birden çok taşıma planı ile ilişkilendirilebilmesi anlamına gelmektedir. Bir üst seviyeye çıkıldığında bahse konu planlar bağlı oldukları bu üst birim tarafından birleştirileceğinden ve olası çatışmaları giderileceğinden üst seviyelere çıkıldıkça o seviyedeki idari planla birleştirilecek taşıma planı sayısı da planlamaya katılacak birim sayısının azalmasına bağlı olarak gitgide azalacaktır.

#### **4.4.2. Lojistik (Taşıma) Planlar ile İdari Planların İlişkilendirilmesi**

Lojistik plan oluşturulmadan önce, idari planın lojistik planca da anlaşılabilir bir formatta şekillendirilmesi gereklidir. Bu kapsamda;

- İlk önce yeni bir idari plan oluşturulur ve hiyerarşideki yeri belirlenir.
- Müşteriden gelen talep çerçevesinde; müşterinin taşınmasını istediği bütün envanter bilgileri VTYM modülü vasıtasıyla sisteme dâhil edilir ve taşımaya esas hale getirilir [yolcu, en, boy, yükseklik, tonaj, metrekare, metreküp, TEU, şerit metre (Lane Meter) olacak şekilde paketlenmesi ve taşıma planlamacıları bakımından anlamlı hale getirilmesi].
- Taşınacak birimlerin anlamlı gruplar şeklinde düzenlenmesini sağlayacak taşınacak birim etiketleri oluşturulur.
- Taşıma planı ve ilişkili olduğu idari plan tarafından kullanılacak bir taşınacak birim profili oluşturulur. Bu profilin oluşturulmasından maksat plan dâhilinde taşınacak birimlerin gruplandırılmasının sağlanmasıdır.
- Daha önce oluşturulan taşınacak birim etiketleri bu profil içine atanır ve ilgili taşınacak birim etiketinin temsil ettiği bütün envanter ilgili listeden alınarak bu birim etiketleri altına konur. Bu sayede boyutları, hacmi ve ağırlıkları belirli taşımaya esas bir envanter listesi elde edilmiş olur [Örneğin bir fabrikanın komple taşındığını düşünelim. Fabrika isimli bir profil yaratılsın. Her bir departmanı (montaj, marangozhane, saraçhane vb.) temsil edecek birim etiketleri oluşturulsun ve profil içine atansın. Fabrikanın marangozhanesinde bulunan her türlü şekillendirici ekipman taşımaya esas olacak şekilde paketlenmeyi müteakip “Marangozhane Ekipmanı” isimli taşınacak birim etiketi altında miktar bilgisi de görüntülenecek şekilde gruplandırılabilir].

- Müşterinin talep ve ihtiyaçları doğrultusunda taşıma görevleri [nereden nereye ne kadar sürede (CRD)] ve bu görevlerin birbirlerine göre öncelik durumları müşteriyle koordineli olarak belirlenir [Bu literatürde “ihtiyaçların ortaya konması (Statement of Requirements, SOR)” olarak da ifade edilmektedir].
- Hangi taşınacak birim etiketinin (içerdiği envanter ile birlikte) hangi taşıma göreviyle taşınacağı bu birim etiketlerinin ilgili profil içinden seçilerek ilgili görevlere atanması ile belirlenmiş olur.
- Bütün görevler biraraya getirilerek lojistik (taşıma) planlamada direkt olarak kullanılacak bir taşıma görev listesi oluşturulur (Buna literatürde “Yerdeğiştirme Görev Listesi (Mission List of Disposition, MLD) adı da verilmektedir).
- Sonrasında boş bir taşıma planı oluşturulur ve bu planın oluşturulduğu seviyeye de bağlı olarak hangi MLD’yi nasıl kullanacağı (tamamen veya kısmen) belirlenir ve MLD’nin seçilmesiyle birlikte ilişkilendirme tamamlanmış ve lojistik planın hiyerarşideki yerini alması sağlanmış olur.

Burada dikkat edilmesi gerekli nokta MLD’lerin operasyonu yürüten en üst birim tarafından ve tek elden hazırlanması gerektiğidir.

#### **4.4.3. Lojistik (Taşıma) Planlama**

Bu planlama kendi içinde “bir defaya mahsus taşıma” ve “tekrarlı taşıma” olmak üzere iki gruba ayrılmaktadır. Bir defaya mahsus taşıma; tek bir planla gerçekleştirilen, planın tamamlanması ile sona eren ve devamlılık arz etmeyen taşıma planlamalarıdır. Belli bir envanterin (Yolcu, araç, ekipman vb.) bir noktadan diğerine taşınması kapsamında yapılan planlamalar buna örnektir. Tekrarlı taşıma ise müşteriden gelen talepler değerlendirilerek belli noktalar arasında yıl/ay içinde belli periyotlarla tekrarlanacak şekilde yapılan taşımalar çerçevesindeki planlamalardır. Kurye ya da ring taşımacılığı olarak literatürde yer alan taşımalara ilişkin planlamalar bu kapsamdadır. İkisi arasında tekrarlanma haricindeki en büyük fark, ikincisi için öngörülen planlama periyodu içinde bir kez (alt yapı ve ulaştırma araç kullanımından kaynaklanan olası güncellemeler nedeni ile yılda birkaç kez de olabilir) MLD hazırlanırken, ihtiyaç duyuldukça gerçekleştirilen ilki için her defasında yeni bir MLD oluşturulması zorunluluğudur.

Kullanıcının seçtiği taşıma planlama türüne göre bu aşamada farklı modüllerin kullanımı [bir defaya mahsus taşıma planlaması için Taşıma Planlama Modülü (TPM), tekrarlı taşımaların planlanması için Kurye/Ring Planlama Modülü



(KURPM)] söz konusu olacaktır. Bu modüller kullandıkları taşıma verileri açısından birbirine benzer özellikler sergilese de kullandıkları yöntem ve izledikleri süreçler bakımından birbirlerinden farklı yapıdadırlar.

#### **4.4.3.1. Bir Defaya Mahsus Taşımaların Planlanması**

Plan ilişkilendirmesi sayesinde elde edilen MLD oluşturulan boş taşıma planı içinde açılır. Bu sayede karar destek modeli içine girecek taşınacak birim etiketleri ve içerdikleri envanter taşınmaya hazır hale getirilmiş olur.

Müteakip aşamada EK-7'de sunulan tam sayısal karar destek modelinin bir tam sayısal programlama aracı olan CPLEX paket yazılımı içinde çalıştırılmasından önce, arzu edilen nitelikte çıktıların daha kısa sürede alınması maksadıyla bazı ön değerlendirme ve analizler yapılır. Bunlar;

- **Taşınabilirlik (Transportability) Analizi:**

Plan kapsamında taşınacak envanterin hangi ulaştırma araçlarıyla taşınabileceğine ilişkin olarak sistem araç havuzunda mevcut olan her bir ulaştırma araç cinsi ve miktarı bazında yapılan değerlendirmedir. Bu analiz aslında kullanılacak ulaştırma modelinin Kullanıcı tarafından sisteme tanımlandığı tecrübî/sezgisel değerlere dayalı bir hüristiktir. Taşıdıkları özellikler dikkate alınarak manifestoda yer alan her bir envanter için Kullanıcı tarafından belirlenecek ulaştırma modunda en uygun ulaştırma aracı ve o moda ilişkin olarak ihtiyaç duyulan toplam ulaştırma aracı cins, miktar ve maliyet olarak bu sayede belirlenmiş olur. Maliyetleri ve ekonomiklikleri de dikkate alınarak, bütün ulaştırma modları için elde edilen veriler çerçevesinde kullanıcıdan hangi ulaştırma araçlarının kullanacağını belirlemek beklenir. Bu analiz yapılmadığı takdirde model bunu kendisi belirleme yoluna gidecek ve bu da sistemin problemi çözme süresini büyük oranda artıracaktır. Bu analiz sonucunda Kullanıcı plan kapsamında bir ulaştırma araç havuzu oluşturmuş olacaktır. Buradan elde edilecek veriler aynı zamanda sistem araç havuzundaki araç cins ve miktarlarının yeterli olup olmadığını da gösterecek ve sistem tarafından sisteme kayıtlı belli bir araç tipinin araç kiralama yoluyla havuza dâhil edilmesi gerektiği bildirilmiş olacaktır.

- **Taşıyabilirlik Analizi:**

Bu analizin yapılabilmesi için sisteme önceden tanımlanan her bir araç cinsi için hangi tip ve ebatta malzemeleri/envanteri taşıyabileceğinin belirlenmiş olması gereklidir. Bu da son derece zor bir işlemdir. Ulaştırma aracının

taşıyabileceği en büyük malzeme boyutları ve ağırlığı araçların toplam kapasitesini geçemeyeceğinden hareketle, boyutlar ve yükleme kısıtları dikkate alınarak bir değerlendirme yapılması mümkündür. Bu da manifestodaki her bir envanter için sistemde kayıtlı bütün araç tiplerinin tek tek denenmesi, her bir araç cinsi için belki de sonsuz sayıda işlemin (iteration) gerçekleştirilmesi anlamına geldiğinden az miktardaki envanter için kullanılabilir olmasına rağmen envanter miktarı arttığında uygulanması son derece zor bir analizdir. Örneğin; sistemde kayıtlı araç tipi sayısı 50 ve taşınacak envanter de 50 parçadan oluşuyor olsun. Envanter boyutları dikkate alındığında her bir araç içine her defasında en fazla 5 envanter sığıdığını farz ve kabul edelim. Bu 50'nin 5'li kombinasyonu anlamına gelir ki bu da her bir araç cinsi için 2.118.760 farklı yükleme alternatifine işaret eder. Bu bağlamda sistemi son derece yavaşlatacak bu analizin bir kenara bırakılması ya da işe yarar bir paket program kullanılması uygun olacaktır.

- **Rota (Güzergâh) Analizi:**

Sistem kapsamında Kullanıcıdan;

- VTYM içinde yer alacak bir CBS paket programı (Map Info gibi) sayesinde sistemin en uygun rotayı bulmak için bütün harita sistemi verilerini taramasının önlenmesi maksadıyla çıkış ve varış noktalarını içine alacak şekilde ilgi alanı ve etki alanı belirlemesi,

- Belirlenen bu alanlar içinde yer alan bütün ulaştırma arklarının taranmasını ve ters ulaştırmaya mahal verilmesini engellemek için bazı arkların (çoğunlukla tali olanlar) adım adım iptal edilmesi,

- Bazı lokasyon ya da ulaştırma alt yapılarının kullanımının önlenmesi beklenir. Bunun yapılmasının nedeni sistemin gereksiz belki de milyonlarca işlemi gerçekleştirmesinin önlenmesi ve karar destek sistemi içine mümkün olan en az sayıda arkin girmesinin sağlanmasıdır. Sistem müteakiben gerçekleştirilecek bu ön analiz sonucunda biri asıl (en uygun) diğer dördü nispeten daha az uygun beş alternatifi kullanıcıya sunacaktır. Bunun sonucunda bütün ulaştırma modelleri için kullanılabilir bütün rotaların karar destek sistemine veri olarak girmesi sağlanacaktır.

- **Ulaştırma Modu Analizi:**

Sistem tarafından en uygun ulaştırma modları maliyet etkin olarak kullanıcıya sunulacaktır. Ancak, cevap süresinin kısaltılabilmesi için Kullanıcı

yapacağı seçimle sistemin belirli cins envanter için belirli ulaştırma modellerini zorunlu olarak kullanmasını sağlayabilir. Örneğin; Yolcu araç, konteyner ve dağıntık yüklerin sisteme taşınacak envanter olarak tanımlandığını düşünelim. Gidilecek son noktanın deniz aşırı bir ülke olması durumunda çıkış noktasından çıkış deniz limanına kadar araç ve malzeme taşınması için demir yolu ve çıkış noktasına belirlenen çıkış deniz limanı kadar uzak olan çıkış havalimanına kadar yolcu taşınması için kara yolu ve varış deniz limanı ve varış hava limanlarından son noktaya kadar da kara yolunun kullanılması gerektiği sisteme dikte edilebilir.

- **Ulaştırma Alt Yapı Kullanılabilirlik ve Elleçleme Yeteneği Analizi:**

Bu analiz ile belirlenmiş etki ve ilgi alanı içindeki ulaştırma alt yapılarının;

- Mevcut kapasite ve çalışma zamanı kısıtları dikkate alınarak taşınacak birimlerce planlanan/belirlenen zaman aralığında herhangi bir çatışmaya meydan vermeden aynı anda kullanılabilme durumlarının ortaya konması,

- Taşınacak birimleri elleçleme yetenekleri değerlendirilmesi ve envanterin elleçlenemeyeceği ulaştırma alt yapısına uğramasının karar destek sistemi çalıştırılmadan önce önlenmesi sağlanacaktır. Herhangi bir malzemeyi elleçlemek için uygun hiçbir ulaştırma altyapısının bulunmaması durumunda elleçlemeyi gerçekleştirecek ekipmanın da mümkünse en uygun ulaştırma alt yapısına götürülmesi sağlanacaktır.

Kullanıcı tarafından yukarıda açıklanan analizlerin yapılmasını müteakip malzemelerin bir arada nasıl taşınacağına ya da başka bir ifade ile bir arada taşınması zorunlu görülen ve birbirinden ayrılmadan taşınması gerekli envanterin belirlenmesi işlevi gerçekleştirilir. Bu, aslında taşımanın bir bütün olarak planlanması yerine birlikte taşınması zorunlu küçük parçalara bölünerek ayrı ayrı planlanması anlamına gelmektedir. Bu kullanıcıya taşıma planlamasında farklı ulaştırma modlarının aynı anda kullanılabilmesi açısından esneklik sağlamaktadır. Literatürde buna "Bileşenlere Ayırma (Splitting Components, SC)" adı verilmektedir.

Bundan sonraki aşamada, taşınacak envanter içinde birbiri üzerine/içine yüklenebilecek olanlar belirlenir ve bu yüklemeler gerçekleştirilir. Bu literatürde "Malzeme Kafeslemesi (Item Nesting)" olarak ifade edilir.

Bu noktada artık her türlü veri karar destek modeli tarafından değerlendirilmeye hazırdır. Bu model içinde en uygun tam sayısal değerler

arandığından yukarıda açıklanan ön işlemlerin gerçekleştirilmesi elleçlenmiş verilerin bu kapsamda sağlanması önem arz etmektedir. Aksi takdirde sistemin aylarca ve belki de yıllarca çalışmasına neden olacak sayıda işlemi ve taramayı gerçekleştirilmesi gerekebilir. Zira, herhangi bir problem çözüldükten getirilen kısıt sayısının artması problemi çözecek modellerin o ölçüde üssel olarak artıracak ilave işlem yüküyle karşı karşıya kalması anlamına gelmektedir. Bu gibi durumların önüne geçilmesi için karar destek modeline ilaveten EK-8'de sunulan sezgisel (hüristik) algoritmanın kullanılması önerilmektedir.

#### 4.4.3.2. Tekrarlı Taşımaların (Kurye/Ring Seferi) Planlanması

Tekrarlı taşımalar her ne kadar ilk modelle planlanabilse de her bir periyodik sefer için her defasında bir takım işlemlerin tekrarlanması gerekeceğinden hareketle daha kullanışlı bir model tasarlanmıştır.

Bu kapsamda tekrarlı taşımalar için iki farklı tam sayısal model oluşturulmuştur. Bu modellerin ilkinde gelen taşıma taleplerinin değerlendirilmesi ve planlanması, ikincisinde ise tekrarlı taşımaların bir bütün olarak değerlendirilmesi sayesinde maliyet etkin bir periyodik taşıma sisteminin oluşturulması amaçlanmaktadır.

- **Talep Karşılama Modeli:** Bu modelde, farklı şirket birimleri vasıtasıyla müşterilerden gelen başlangıç ve bitiş lokasyonları belirli taşıma taleplerinin belli bir zaman aralığında taşınmasının planlanması problemi ele alınacaktır. Her ne kadar gerçek hayatta yapılan taleplerin bir an önce karşılanması gerekse de kaynak kısıtı nedeniyle bu her zaman mümkün olamamaktadır. Ayrıca, taşınacak malzemelerin miktarına ve müşterinin belirteceği öncelik derecesine göre gelen taleplerin mümkün en yakın kurye seferine mi yoksa yer durumu müsait en uygun sefere mi planlanacağına açıklığa kavuşturulması gereklidir. Başka bir ifade ile bütün talepler karşılanamadığı durumda herhangi bir önem derecesindeki bir talebin kabul edilebilmesi için, kendisinden düşük önceliğe sahip olan bütün taleplerin reddedilebilmesi ya da ötelenbilmesi gerekecektir. Bunun yanında yaşanan yoğunluk nedeniyle "talep iptal zamanı" adı verilen müşteri tarafından tahammül edilebilir bir taşıma süresinin ve Kullanıcı tarafından kabul edilebilir araç doluluk yüzdesinin belirlenmesi de söz konusudur. Bu modelle, taşıma önceliklerini göz önüne alacak şekilde kabul edilen taleplerin sayısının en üst seviyeye çıkartılması amaçlanmaktadır. Sistemdeki her kurye / ring'in aşılmaması gereken bir istiap haddi, ve gerçekleştirilmesi için her bacağında ulaşılması gereken bir kullanım

oranı vardır. Taşıma önceliklerine göre taşınabilen taleplerin maksimizasyonunun yanı sıra, bu işlemin minimum maliyetle yapılması göz önünde bulundurulmaktadır.

- **Kurye/Ring Sefer Sistemi Optimizasyon Modeli:** Bu modelle, tekrarlı taşıma (kurye/ring sefer) sisteminin maliyet etkin bir biçimde planlanabilmesi amaçlanmaktadır. Belli bir zaman aralığında yapılmış, karşılanmış, reddedilmiş, ötelenmiş talep verileri (başlangıç lokasyonu, bitiş lokasyonu, talep tarihi ve talep iptal zamanı) ışığında ve gerçekleştirilen kurye/ring seferlerinin belli taşıma bacaklarındaki doluluk oranları, yolcu/malzeme indirme/bindirme durumları dikkate alınarak tüm kurye/ring sefer sisteminin bir bütün olarak değerlendirilmesi öngörülmektedir. Bu sayede bazı seferlerin birleştirilmesi, bazılarının iptali, bazı seferlere yeni taşıma bacağı eklenmesi konularında karar desteği sağlanabilecektir.

Her iki modele ilişkin ayrıntılar EK-9'da sunulmuştur.

#### **4.5. ALGORİTMADAN BEKLENEN KARAR DESTEK İŞLEMLERİ**

Optimizasyon, esas itibarı ile bütün faaliyetlerin tek bir modelde birleştirilmesi şeklinde gerçekleştirildiğinde (genel kısıtlar ve kullanıcı tarafından belirlenen kısıtlar da dikkate alınarak planlama yapıldığında) oldukça uzun zaman alabilecektir. Bunun önlenmesi için genel bir kitle yerine işlenmesi/hesaplanması/değerlendirilmesi nispeten daha kolay olan, birinin çıktısı diğerine girdi sağlayan, birbirini takip eden veya aynı anda birden çok mantıksal akış sırası izleyen küçük optimizasyonlar dizisinin/setinin/sisteminin kullanılması daha mantıklı olacaktır. Sistemin mevcut donanım altyapısına da bağlı olarak aynı anda bir veya daha çok optimizasyon yapılabildiği takdirde kullanılan/harcanan hesaplama zamanı daha da kısıllacaktır. Bu küçük optimizasyonlar, aslında birer küçük model olarak tanımlanacaktır.

##### **4.5.1. Maliyet Optimizasyonu**

Yapılacak her türlü taşımanın en maliyet etkin şekilde gerçekleştirilmesi kapsamında kabul edilebilir en uygun maliyetin hesaplanabilmesi esasına dayanmaktadır.

##### **4.5.2. Güzergâh Optimizasyonu**

###### **4.5.2.1. Ulaştırma Altyapı Optimizasyonu**

Elleçleme yeteneklerinin değerlendirilmesi

- Mevcut elleçleme araçları kullanım durumu (Faal, bakımda, arızalı, parça bekliyor, vb.)

- Mevcut elleçleme araçları işletim durumu (3 saat çalışmayı müteakip 2 saat kullanılmama gibi)
- Mevcut elleçleme araçları ile elleçlenebilecek envanterin kontrolü
- Günlük işletim zamanlarının değerlendirilmesi
- Günde toplam işletim süresi
  - Tek vardiya (8 Saat)
  - İki vardiya (16 Saat)
  - Üç vardiya (Tam gün)
- İşletim başlama ve bitiş zamanları
- Dolu/boş olma durumunun değerlendirilmesi
- Mevcut apron/yükleme-boşaltma yeri/elleçleme noktası/aktarma noktası durumunun değerlendirilmesi
- Mevcut bekleme/konaklama/duraklama/park yeri durumunun değerlendirilmesi.
- Elleçleme sonrası ulaştırma alt yapısında müteakip taşımaya kadar bekleyebilme durumunun (Slot definition) değerlendirilmesi.
- Taşıma esnasında kullanım durumunun değerlendirilmesi (seçilecek yolların kullanılabilir olması)
- İlgili sahasına giriş çıkış noktalarının belirlenmesi ve kullanım durumunun değerlendirilmesi.

#### 4.5.2.2. Kullanılacak Güzergâh Optimizasyonu

Güzergâh optimizasyonunda dikkate alınabilecek hususlar aşağıda sıralanmıştır.

- **En Kısa Yolun Bulunması:** Mesafe kısaldıkça maliyet düşeceğinden en kısa yolun bulunması optimizasyonun ilk adımıdır. Sistem ilgili sahası içindeki mevcut ağları değerlendirerek en kısa yolu ve buna alternatif olacak, adedi kullanıcı tarafından belirlenecek (kullanıcı her hangi bir tercih belirtmemişse varsayılan olarak 3 adet alınacaktır) yeteri kadar güzergâh sistem tarafından bulunarak kullanıcıya sunulacaktır. Bu ara yüzde her bir güzergâhın kullanılacağı ulaştırma bacaları, ulaştırma altyapıları, aradaki mesafeler ve toplam mesafe de belirtilecektir. Kullanıcı

istediğinde ilgi sahasındaki ulařtırma ađları ve ulařtırma altyapıları üzerine kısıt getirebilecektir. Bu sayede sistemin daha az alternatifi deđerlendirmesi sađlanabilecektir.

- **En Uygun Ulařtırma Altyapılarının Bulunması:** Sistem, bu yaklařımda öncelikle tercih edilen ilgi sahasını deđerlendirerek bulacađı (ilgi sahasına giriř ve ıkıř noktaları kullanıcı tarafından belirlendiđi takdirde deđerlendirme zamanı kısılacaktır), kullanıcı tarafından belirlenecek veya en kısayollar iinden seilecek bir güzergâh üzerindeki lokasyonların; mevcut tařınacak birimin ya da envanterin zorunlu uđrayıř (sistemin kullanıcı tarafından kullanmaya zorlandıđı lokasyonlar iin), transit geiř, duraklama, yük aktarma, bekleme/park ve ellelenme durumlarını dikkate alarak kullanılabilme durumunu deđerlendirecek ve güzergâhı (ana güzergâh) alternatifleri (tâli güzergâhlar) ile birlikte belirleyebilecektir.

- Genel tercihlerde yer alan/kullanıcı tarafından belirlenecek/plan ara havuzunda yer alan ulařtırma aralarına göre güzergâhın belirlenmesi: Sistem, genel tercihlerde varsayılan olarak kayıtlı olan ya da kullanıcının belirlediđi tercihlere, mevcut envanterin tařınabileceđi aralar listesine veya tamamen kullanıcının belirleyeceđi ara tipi/tiplerine göre bir ön deđerlendirme yapacak ve bahse konu araların kullanabileceđi, uđrayabileceđi, ellelenebileceđi lokasyonları deđerlendirerek güzergâhı (ana güzergâh) alternatifleri (tâli güzergâhlar) ile birlikte belirleyebilecektir.

- Ulařtırma ađı iinde yer alan yol durumlarına göre güzergâhın belirlenmesi: Ulařtırma ađı iinde yer alan yolların özellikleri ve günlük kullanım kapasiteleri dikkate alınarak ilgi sahası iinde kullanılacak yollar tespit edilecek ve bu sayede kullanılacak güzergâh belirlenebilecektir.

- Ulařtırma altyapısının bařka tařımalarda kullanılma durumları ve doluluk veya hizmet verim süreleri de dikkate alınarak zamansal aıdan kullanıma/tahsise yönelik optimizasyonu sađlanacaktır.

- Kullanıcı tarafından gerektiđinde belirli lokasyonların veya güzergâhların kullanımına iliřkin ilave kısıt girilebilecek ve bu kısıt optimizasyonda dikkate alınacaktır.

#### 4.5.3. Ulaştırma Aracı Optimizasyonu

- Taşınabilme kategorilerine yönelik gerekli analiz de yapılarak taşınacak birim envanterini (kargo ve/veya yolcu) taşımak üzere kullanıcı tarafından sisteme tanımlanmış araç havuzu içerisinde en uygun araç/araçların tip ve miktarları belirlenecek (Bu yapılırken sisteme daha önce genel araç tercihleri şeklinde varsayılan olarak atanmış araç tipleri veya kullanıcının üzerinde çalışılan plana göre yaptığı kendi tercihleri de opsiyonel olarak dikkate alınacaktır),
  - Araçların tahsis edileceği her bir birim/unsur/bileşen belirlenecek,
  - Bileşenlerin bir defada taşınması durumunda ihtiyaç duyulacak ilave araç tip ve miktarları kullanıcıya sunulacak,
  - İlave araç sisteme tanımlanmadığı durumlarda mevcut araçlar ile gerektiğinde birden fazla sefer/sorti planlaması yapılabilecek,
  - Havuzdaki araçların başka taşımalarda kullanılma durumlarını da dikkate alarak zamansal açıdan bir kullanım optimizasyonu sağlanacak,
  - Belirlenen araçların daha önce kullanıcı tarafından uygunluğu onaylanan güzergâhta kullanılabilme durumlarına yönelik analiz yapılacak ve uyumsuzluklar kullanıcıya sunularak gerektiğinde değişiklik yapılmasına imkân tanınacak,
  - Ulaştırma araçlarının yüklenme kapasitesini aşmayacak ve gerektiğinde belli yüzdelerle eksik yükleme yapılması (doluluk yüzdesinin bildirilmesi) konusunda kullanıcı tarafından girdi yapılabilecek (kullanıcı herhangi bir değer girmediği takdirde varsayılan değer %90 alınacaktır),
  - Ulaştırma araçları gerektiğinde tam sefer yapabilmek maksadıyla boş olarak hareket ettirilebilecek,
  - Konvoy oluşturulurken aynı ve/veya yakın hız kapasitesindeki araçlardan oluşturulması sağlanarak hızlı/yavaş konvoy oluşturulmasına imkân tanınacak,
  - Araç havuzuna tanımlanmamış olmakla birlikte taşınacak birim envanterinde bulunan ve sisteme taşınabilme kategorisi olarak “yüklenbilir kendi kendini taşıyabilir araç” şeklinde tanımlanmış araçların yüklemede öncelikle yüklenmesi ve bu envanter kategorisindeki araçların gerektiğinde konvoy oluşturularak kara yolundan hareket edebilmesi sağlanacaktır.



#### 4.5.4. Yükleme Optimizasyonu

Sistem tarafından;

- Kullanıcının belirlediği araç havuzundan seçilen her hangi bir araca ilişkin olarak yüklenmesi öngörülen taşımaya esas malzemelerin yüklenmesinde aracın yolcu/yük kapasitesi dikkate alınacak (Örn.: 10 kişi ve 5 ton kapasiteli bir araca 11 ve üstü kişi ve tonajı aşan yük yüklenemeyecek, aracın hiçbir zaman yük kapasitesinin tamamını kullanmayacağı prensibinden hareketle aracın en fazla %90 oranında yüklenmesi sağlanacak,

- Kullanıcının belirlediği araç havuzundan seçilen her hangi bir araca ilişkin olarak yüklenmesi öngörülen taşımaya esas malzemelerin yüklenmesinde aracın yük dengesi dikkate alınacak,

- Kullanıcının belirlediği araç havuzundan seçilen her hangi bir araca ilişkin olarak; taşınacak birime ait envanter yüklenirken, yükleneceği bölümün kapı/kapak genişlik ve yüksekliği dikkate alınacak,

- Kullanıcının belirlediği araç havuzundan seçilen her hangi bir araca ilişkin olarak; taşınacak birime ait envanter yüklenirken, yükler arasında bulundurulması varsayılan değer olarak kullanıcı tarafından belirlenmiş zorunlu boşluk değerlerini dikkate alacak,

- Kullanıcının belirlediği araç havuzundan seçilen her hangi bir araca ilişkin olarak; taşınacak birime ait envanter yüklenirken, araç ve yüklenen malzeme maksimum istif yüksekliğini/adedini dikkate alacak (Örn.: Aracın kargo alanı yüksekliği 2,00. metre ve bu bölüme yüklenebilecek maksimum istif yüksekliği 1,90. metre, yüksekliği 35 cm ve maksimum istif adedi 4 olan ve üzerine malzeme yüklenemeyecek bir malzeme dikkate alındığında bu malzemenin altına, üzerine malzeme yüklenebilir ve 50 cm. ye kadar istif yüksekliğini sağlayabilecek bir malzeme yüklenebilir),

- Her çeşit ulaştırma aracına (kara yolu, deniz yolu, hava yolu, demiryolu) yükleme yapılabilmesi sağlanacaktır ve bu kapsamda;

- Kara yolu araçlarına yükleme yapılırken;

- ◆ Temel yükleme prensiplerine uyulacak,

- ◆ Yüklenen malzemelere ilişkin ağırlık, maksimum istif adedi, kontrollü (hassas) malzeme, gizlilik dereceli malzeme gibi kıstasları dikkate alacak,

- ◆ Farklı güzergâhlardan aynı ulaştırma aracına yükleme yapılabilmesi sağlanacak ve farklı terminallerden yüklenecek envanter için ulaştırma aracının ilgili bölümlerinde önceden yük tahsisi (pre-assigned stow) planlanarak boş alan bırakılabilecek,

- ◆ Farklı terminallerde indirilecek malzemelere, terminalin ve indirme/yükleme aparatlarının durumuna göre yükleme planlaması yapılacak ve LIFO/FILO-Son Giren İlk Çıkar/İlk Giren Son Çıkar, FIFO/LILO-İlk Giren İlk Çıkar, Son Giren Son Çıkar prensiplerine, yükleme/ boşaltma konumlarına (önden, arkadan, üstten) ve kendi kendine yükleme durumlarına göre planlama yapılabilecek,

- ◆ Taşınan malzemenin özelliğine göre konvoy oluşturulmuşsa konvoydaki araç sırası otomatik olarak sistem tarafından düzenlenecek (Örn.: patlayıcı madde taşındığında tehlikeli maddenin bulunduğu araç konvoyun en sonuna planlanmalı ve bir önündeki araçla arasında yeteri kadar mesafe bırakılmalıdır. Güvenlik gerekçesiyle özel emniyet tedbirleri uygulandığında bu aracı yeteri kadar mesafeden takip eden emniyet unsurları bulunmalıdır),

- Demiryolu araçlarına yükleme yapılırken;

- ◆ Temel yükleme prensiplerine uyulacak,

- ◆ Farklı güzergâhlardan aynı ulaştırma aracına yükleme yapılabilmesi sağlanacak ve farklı terminallerden yüklenecek malzemeler için ulaştırma aracının ilgili bölümlerinde önceden yük tahsisi planlanarak boş alan bırakılabilecek,

- ◆ Taşınan malzemenin özelliğine göre katardaki vagon sayısı otomatik olarak belirlenip sistem tarafından otomatik yeni katar oluşturulabilecek (Örn.: patlayıcı madde trenle taşındığında tehlikeli maddenin bulunduğu vagonun önündeki ve arkasındaki vagonlarla arada yeteri kadar boş platform vagon bırakılmalıdır),

- Deniz yolu araçlarına yükleme yapılırken;

- ◆ Temel yükleme prensiplerine uyulacak,

- ◆ Yük yakıt dengesi dikkate alınacak,
- ◆ Balast tanklarının doluluk değerleri dikkate alınarak geminin genel dengesinin (ağırlık merkezinin) ayarlanabilmesi için geminin suya en uygun batma seviyesi (draft) belirlenecek,
- ◆ Trim (öne arkaya eğim bakımından geminin dengede olması- 0 derece eğim gereklidir) dengesi sağlanacak,
- ◆ Yalpa açısının (geminin iskele ve sancağına göre yatma derecesinin gemiyi dengede tutacak şekilde ayarlanmış olması) sıfır olması sağlanacak,
- ◆ Yükleme rampası olarak kullanılan kapağın yüklenecek malzeme dikkate alındığında yük dayanım durumu dikkate alınacak,
- ◆ Rampa yükleme kapaklarının meyil açısı dikkate alınacak,
- ◆ Gemi güvertesinin santimetrekare başına taşıyabileceği ağırlık (santimetrekareye düşen taban dayanım değeri) dikkate alınacak,
- ◆ Yükleme güvertesinin sabitleme ekipmanının gemi üzerindeki konumları dikkate alınacak,
- ◆ Yükleme güvertesinin yüklenebilir bölümünün ve maksimum ebatla yüklenebilecek malzemenin ebatları dikkate alınacak,
- ◆ Geminin kendi kendini yükleme özelliği (kreyn miktarı, kapasitesi ve durumları) dikkate alınacak ve yükleme yapabileceği bölmelere ilişkin özel planlama yapılabilecek,
- ◆ Geminin teknik talimnamesindeki genel gemi yükleme planlaması esasları dikkate alınacak,
- ◆ Farklı güzergâhlardan aynı ulaştırma aracına yükleme yapılabilmesi ve farklı terminallerden yüklenecek malzemeler için ulaştırma aracının ilgili bölümlerinde önceden yük tahsisi planlanarak boş alan bırakılması sağlanacak,
- ◆ Tehlikeli maddelerin yüklenmesine ilişkin esaslar dikkate alınacak ve taşınabileceği bölmelerin yerleri belirlenecek,
  - Hava yolu araçlarına yükleme yapılırken;

- ◆ Uçak temel yükleme prensiplerine uyulacak,
- ◆ Sisteme tanımlı kargo uçaklarına (sivil/askeri), uçakların özelliklerine ve teknik kullanım ve bakım kitapçıklarında belirtilen yükleme esaslarına uyularak yükleme yapılacak (Bu yapılırken araç tipi bazında değil her bir gerçek araç bazında optimizasyon sağlanmalıdır. Çünkü, araçların bazı özellikleri sadece kendisine münhasır olabilmekte ve kapasite/yükleme alanlarında ait olduğu tipe göre farklılıklar olabilmektedir. Örneğin; bazı uçakların belirli noktalarındaki taban alanlarına taban dayanımını artırmak maksadıyla destekleme (threadways) yapılmıştır),
- ◆ Yükleme (kara yolu, demiryolu, deniz yolu ve hava yolu) tamamlandıktan sonra söz konusu yüklemeye ilişkin olarak yükleme planı, yüklemenin yapıldığı bölüm/bölme bilgilerini, malzemelerin yüklendiği yerlerin krokilerini ve istif bilgilerini de içerecek şekilde sistemden rapor ve çıktı alınabilecektir.

#### **4.5.5. Taşıma Optimizasyonu:**

Taşıma planı hazırlarken şu esaslar dikkate alınacaktır:

- Taşıma için ihtiyaç duyulan araç tip ve miktarlarını, taşınacak birimlerin takip edecekleri güzergâhları ve yükleme planlarını kullanıcının onayını müteakip planlamaya alacak,
- Altyapı kullanım durumunu dikkate alacak,
- Taşınacak Birim taşımalarının ve araç hareketlerinin zamanlamasını planlayacak,
- Araçların boş ve dolu olarak takip edeceği güzergâhların kullanıcı tarafından belirlenmesini sağlayacak,
- Planların gerektiğinde bir üst seviyede birleştirilmesi sağlanarak planlamalardaki çatışmalar sistem tarafından gösterilecek ve çözüm yolları ortaya konacak,
- Planların her seviyede farklı detayda hazırlanabilmesine imkân tanınacak,
- Yapılan planlamalar sayısal haritalar üzerinde anime edilecek,

- Yapılan her türlü planlamada sonuca manuel olarak müdahale edilebilecektir.

#### **4.6. DİĞER ALT MODÜLLER**

Yazılıma ilişkin bütün modüller ve işlevleri Sistem İhtiyaç Belirleme (SRS) Dokümanında yer almaktadır (EK-6). Özellik arz eden bazı modüllere ilişkin özet bilgiler aşağıdaki maddelerde sunulmuştur.

##### **4.6.1. VTYM'nin Sahip Olacağı Alt Modüller**

###### **4.6.1.1. Coğrafi Bilgi Sistemi (CBS) Alt Modülü**

Bilindiği gibi Coğrafi Bilgi Sistemi (CBS) mekânsal verilerin değerlendirilmesi, analizi, modeli ve sunumunu içeren bilgisayar tabanlı bir bilgi sistemidir. Bilgisayar teknolojisinin hızlı bir şekilde gelişmesi bilgisayar ortamında bilginin daha verimli olarak kullanılmasını sağlamıştır. Bu durum diğer bilgi sistemlerinde olduğu gibi CBS'de de kendini göstermiş, özellikle karar destek sistemlerinin coğrafik tabanlı veriler üzerindeki uygulanabilirliğini etkin bir şekilde geliştirmiştir. Çünkü CBS karar ve yönetme fonksiyonlarını etkileşimli bir şekilde değerlendiren, birbirleriyle ilişkilerini sağlayan, mekansal verileri işleyen, analizini yapan, modelleyen ve görselleştiren, diğer bilgi sistemlerinden farklı olarak mekansal koordinatlar ile coğrafik veri arasında fonksiyonel etkin bağlantılar kuran bir sistemdir (Güllü, 2003). Karar destek sistemleri, çoklu seçim kriterlerinden ve kriterlerin yönetilmesinde en faydalı çözümü üreten bir yönetim sisteminden oluşur. Bu sistem çerçevesinde değerlendirilebilecek problemler farklı mühendislik uygulamalarını ihtiva eden şehir planlamalarında kendilerine özel olarak yer bulmaktadır. Bu problemlere ait karar destek uygulamaları bir şehir için özellikle endüstri ve inşaat mühendisliği açısından değerlendirildiğinde aşağıda belirtilen bazı problemlerin çözümü için bir ön çalışma olarak değerlendirilebilir. Bunlar; zemin ve deprem durumuna göre riskli bölgelerin belirlenmesi, imar planlarının hazırlanması, doğal gaz, isale ve kanalizasyon hatları için ağ analizleri, otobüs, tren, tramvay ve metro hatları için ağ analizleri, satış merkezleri, okullar, konut vb. için uygun yerlerin seçilmesi olarak sıralanabilir.

Belirtilen bu problemler için standart CBS yazılımlarında çeşitli analizler yapmak mümkün olmakla beraber, çeşitli alternatifler arasından belirli kriterleri sağlayan optimal bir seçim yapmak olanaklı değildir. Fakat bu durum karar destek sistemleri ve çoklu seçim yöntemlerinin CBS yazılımlarına bütünleştirilmesi ile mümkün olabilir. Bu çalışmanın amacı CBS ortamında bu çözüm yöntemlerini

irdelemektir. Bu kapsamda CBS ile birlikte AHP (Analitik Hiyerarşi Süreci) kullanılabilir. Alternatif çözümler içerisinde en uygun olanı bulmak CBS ortamında yazılacak AHP algoritmalarına dayanan bir programlama ile mümkün olacaktır. Bu işlem için uygun bir CBS yazılımı ve programlama dilinin kullanılması gerekmektedir.

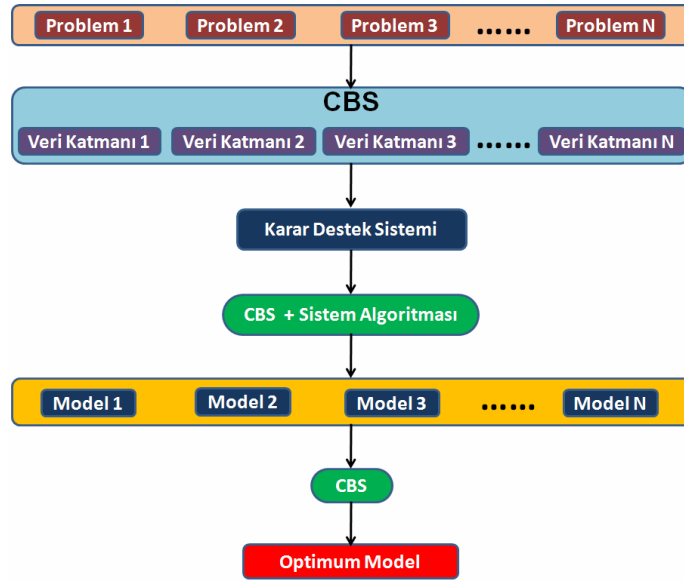
Literatürde çoklu seçim problemlerinin CBS ve karar destek sistemleri ile çözümüne ilişkin entegre çalışma ve uygulamalarına ait yeterli çalışmalara rastlanamamıştır. Gerçekleştirilen birçok uygulama tek seçim kriterli çözüm yöntemlerini içermektedir. Ancak tek seçim kriterli çözüm yöntemleri bile olsa optimizasyon analizlerine dayalı olarak yapılan literatür çalışmalarını incelemek, hem CBS'nin fonksiyonunu daha iyi anlamak hem de karar destek sistemlerinin önemini ortaya koymak açısından önemlidir.

Tom (1995), New Mexico için ulaşım planlaması yapmıştır, bu kapsamda ulaştırma problemleri için CBS'de bir altyapı oluşturmuş ve problemlerin CBS fonksiyonları ile nasıl çözüleceğini göstermiştir. Bu çözümlerde sezgisel algoritmalara dayalı karar destek yöntemlerini de denemiştir. Miller ve Shaw (2001) CBS'nin en fazla uygulandığı ulaştırma modellerinde veri işleme ve değerlendirme problemlerini tartışmışlar ve çözüm yöntemleri geliştirmeye çalışmışlardır. Hwang (2001) en faydalı hizmet açısından kamu binalarının yerleşim noktalarının planlanması problemini CBS ile değerlendirmiş ve CBS'de ileri istatistik yöntemlerini kullanarak optimum bir çözüme ulaşmaya çalışmıştır. Güllü (2001) inşaat mühendisliği uygulaması olarak bir bölgeye ait, depremde hasar görmüş binaları ve bölgenin zemin durumunu ayrı tabakalar olarak CBS' ye işlemiş ve kalitatif yöntemleri kullanarak yerleşim için en uygun bölgeleri belirlemiştir. Easa ve Yupa (2000), Balmohan (2002), Anthony (2002) şehir planlaması ile ilgili olarak ortaya koydukları problemlere yine CBS ile bir çözüm getirmeye çalışmışlardır. Sekar (2002) artan nüfus ve sanayileşme ile ilgili bilgileri CBS' de toplayarak şehir plancılığı perspektifinde karşılaşılabilecek problemleri göz önünde bulundurarak CBS'de çözmeye çalışmıştır. Alterkawi ve Mazyed (2002) Riyad için CBS ile ulaşım planlaması yapmış ve CBS kullanarak farklı hizmet hatları için en uygun ağ çözümlerini ortaya koymuşlardır. Chenfu (2002) farklı ulaştırma modellerini CBS'ye işleyerek CBS ile simülasyonunu gerçekleştirmiş ve problem çözümleri için farklı bir yaklaşım sergilemiştir.

Bu kapsamda planlamada karşılaşılabilecek farklı mekansal problemlerin CBS ve karar destek sistemleri ile çözümü araştırılmıştır. Örneğin, bir şehirdeki

alternatif, metro hatları, belediye otobüs güzergahları, okul, alışveriş ve konut noktaları, zemin deprem durumları gibi çoklu seçenek problemlerinde, metro hattı, otobüs güzergahı, okul ve alışveriş noktası ve zemin durumu açısından değerlendirildiğinde en uygun yerleşim noktasının belirlenmesi karar destek sisteminin bir CBS ile birlikte çalışmasını gerekli kılar. Böyle bir durumda ilk önce, problemler coğrafik koordinatlar ile ilişkilendirilerek CBS'ye işlenir. Sonra her bir problem için ayrı bir veri tabakası oluşturulur. Görsel açıdan önem arz ettiği için her bir tabakanın tematik modellemesi yapılır. Daha sonra her bir veri tabakası SQL sorgulama ile farklı kriterler için değerlendirilir. Son olarak her probleme uygun bir model ortaya konur. Uygun modeller ortaya konulduktan sonra ise CBS optimizasyon analizlerine başlanır. Bu amaçla CBS yazılım dili ile problem optimum modele tanıtılır. Sonunda ise önce kısmi çözüm sonra ise kısmi çözümler arasındaki optimum yaklaşım ile genel çözüm elde edilir. Böylece çoklu seçenekler arasında, en uygun hat, güzergâh, yerleşim noktası veya zemin durumu gibi problemler çözüme kavuşturulmuş olur.

Şekil-4.2'de optimizasyon problemlerinin karar destek sistemleri ve CBS ile çözümünde sergileyeceği genel entegrasyon nasıl olabileceği gösterilmektedir. Bu şekil incelendiğinde karar destek sistemlerinin CBS ile etkili olarak kullanılabilmesi için sistem algoritmasının mutlaka CBS yazılım dili ile CBS'ye tanıtılmasının gerekli olduğu görülmektedir. Daha sonra ise optimizasyon yöntemlerine dayalı coğrafik analizlerin yapılması CBS işlevlerini etkin kılacaktır.



**Şekil-4.2:** Karar destek sistemi ve CBS entegrasyonunun değerlendirilmesi

Yukarıda açıklanan hususlar ışığında sistem kendi altında ve kurulacağı her bir istemci bilgisayar üzerinde sunumcudan veri almadan çalışabilecek raster harita sistemi kullanan basit bir modül yanında sadece sunumcu üzerinde çalışacak ve uzaktan erişimle kullanılacak paket program olarak alınmış ve sistemin kullanımına sunulmuş bir sayısal (dijital) harita sistemi modülüne sahip olacaktır.

#### **4.6.1.2.Raporlanabilir Envanter Kodu (REK) Alt Modülü**

Bu modül sayesinde kullanıcının sisteme yeni kaydedeceği her türlü envanterin belli bir standartta tanımlanmış malzeme özellikleri ile ilişkilendirilmesi ve bu envantere ilişkin raporların kolayca sistemden alınabilmesi mümkün olacaktır.

#### **4.6.1.3.Taşımaya Esas Hale Getirme (Paketleme) Alt Modülü**

Sisteme tanımlanacak her bir malzemenin kullanım/işletim ebatlarının taşıma ebatlarına dönüştürülebilmesi bu modülle sağlanacaktır. Örneğin; aynı cinsten kaç tane malzemenin belli bir paket içine sığabildiği bu modül sayesinde belirlenebilecek ve gerektiğinde tekrar kullanılabilir üzere şablonlanabilecektir (1mX1mX1m ebatlarındaki bir kutuya kaç ayakkabı kutusunun gireceğinin belirlenmesi gibi).

#### **4.6.1.4.Taşıma Talep Alt Modülü**

Müşteriden ilgili taşıma birimince teslim alınan envanterin sisteme müşterinin belirlediği önceliğe ve zaman kısıtına göre talep şeklinde girilmesini sağlayan modüldür. Tekrarlı taşımalar kapsamındaki talepler de sisteme bu modül sayesinde girilecektir.

#### **4.6.1.5.Plan Hiyerarşisi Oluşturma Alt Modülü**

İdari plan hazırlama, taşıma ihtiyacı belirleme ve yer değiştirme listesi oluşturma işlevleri bu alt modülle gerçekleştirilecektir.

#### **4.6.1.6.Yetkilendirme ve Güvenlik Modülü**

Bu modül ayrı bir modül olarak tasarlanmayacak ve VTYM altında bir alt modül olarak kodlanacaktır. Bu modülle her seviyedeki kullanıcıların program üzerindeki yetkilerinin belirlenmesi ve tek merkezden kontrol edilebilmesi amaçlanmıştır. Dinamik menü sistemi sayesinde her kullanıcının yetkileri dâhilinde erişebileceği işlevler ekrana gelebilecektir. Sisteme giriş ve çıkışlar, yapılan değişiklikler, kullanıcı yetki ve kısıtları, Kullanıcıya özel sistem ayarları (Kullanıcıya özel düzenlenebilen dinamik menü sistemi), sistem geri yükleme (reloading), sistem



kayıt kütüğü tutulması, sistem güvenlik duvarı, sistem bakımı vb. özelliklerin kullanılabilirliği bu modül sayesinde gerçekleştirilecektir.

#### **4.6.2. TPM'in Sahip Olacağı Alt Modüller**

##### **4.6.2.1. Plan Görüntüleme/Animasyon Alt Modülü**

Bu modülle; yapılan planların sistem tarafından kullanılacak CBS paket programı ya da sisteme dâhil raster harita sistemi üzerinde nasıl işlediğini ve zamana bağlı olarak yolcu miktar ve yük tonaj değişikliklerini de gösterecek, gerçek zamanlı ya da kullanıcı kontrolünde hızlandırılmış zamanlı animasyon gerçekleştirilecektir. Bu sayede sistemde aksayan veya hatalı planlanan hususların görülebilmesi sağlanacaktır.

##### **4.6.2.2. Plan Birleştirme Alt Modülü**

Bu modülle, aynı seviyedeki planların birleştirilebilmesi veya başka bir ifade ile bir üst seviye plana dönüştürülebilmesi sağlanacaktır.

##### **4.6.2.3. Plan Simülasyon ve İstatistik Değerlendirme Alt Modülü**

Günlük hayatta yapılan planlamalara %100 uyulabilmesi çoğunlukla mümkün olmadığından, Kullanıcı bu modül sayesinde alternatif planlar (simülasyon senaryoları) üretmek sistemin gerçekleştirdiği en uygun plandan olası sapmaları hesaplayabilecek/görebilecektir.

##### **4.6.2.4. Plan Çatışmalarını Görüntüleme ve Giderme Alt Modülü**

Bu modül kullanılarak, birleştirilen planların ya da aynı seviyede farklı iki veya daha fazla planın karşılaştırılması sayesinde; kapasiteyi aşacak şekilde aynı anda aynı rota kullanımı, aynı anda aynı ulaştırma altyapısının kullanımı, aynı anda aynı ulaştırma aracının kullanımı gibi uygulanması mümkün olmayan planlamaların görülebilmesi ve yapılacak müdahalelerle düzeltilmesi sağlanacaktır.

## SONUÇ

İhtiyaçların proje başlangıcında tam olarak belirlenmemiş olması, yazılım geliştirme sürecinin kendine özgü yapısından dolayı, önemli bir risk alanı olarak kabul edilmekte ve geliştirme sürecinin ileriki aşamalarında söz konusu gerekliliklerin değişeceği anlamına gelmektedir. Kod üretilmesi aşamasında müşteri tarafından gerekliliklerde yapılacak küçük bir değişiklik, söz konusu sürecin tekrar baştan başlatılması anlamına gelebilmektedir. Bu durum bazen, bütün sürecin sıfırdan başlatılmasına da yol açabilmektedir. Bunun etkisi ise kendini maliyet artışlarında göstermektedir.

Proje başlangıcında tedarik makamı-yüklenici-kullanıcı arasında kurulu bir iletişimin olmaması ya da yeterince olmaması, bir takım riskleri de beraberinde getirmektedir. İletişimin yokluğu, proje sonunda, beklenen üründen çok farklı özellikleri içeren ürünlerin ortaya çıkmasına neden olabilmektedir. Bu ise, tüm projenin çöpe atılması ile aynı anlamı taşımaktadır.

Bu kapsamda, yukarıda belirtilen benzer problemler ile karşılaşılması ve kaliteli ürünün istenen zaman ve maliyetlerde üretilmesi için aşağıdaki unsurların göz önünde bulundurulması gerektiği değerlendirilmektedir:

- Günümüz kalite anlayışı, sürekli gelişim ve mükemmellik felsefesi çerçevesinde, alınan ürün ya da hizmetin tasarımından, kullanım dışına alınmasına kadar geçen süreçte, müşteri memnuniyetinin sağlanması, tasarım, üretim ve tedarik süreçlerinin bu düşünceye göre organize edilmesini ve yönetilmesini gerektirmektedir. Bu yüzden, hem tedarik hem de üretim süreçlerinin organize edilmesinde ISO/IEC 12207 standardının esas alınması oldukça önem taşımaktadır. Bu şekilde, yazılım geliştirme ve tedarik süreçlerinin standardizasyonu sağlanmış olacaktır. Süreçlerdeki her bir faaliyetin tanımlanmış olması, söz konusu süreçlerin kişilere bağlı olmasını engelleyecek ve bu şekilde süreçler sonunda istenen ürünün sağlanması olasılığı artırılmış olacaktır. Aynı zamanda ömür devri süreçlerinin de değerlendirmeye katılması, ürünün tedarik makamı ve kullanıcı açısından maliyetlerinin tam olarak belirlenmesi açısından büyük önem taşımaktadır. Bununla birlikte ISO/IEC 12207 standardının kullanılması, söz konusu standardın uyarlanması sonucu, kullanılmayacak dokümanlar üretilmemesini ve maliyet/etkin olmayan faaliyetlerin gerçekleştirilmemesini sağlayacak ve bu sebeple zaman ve maliyet açısından oldukça önemli tasarruflar sağlanmış olacaktır. Ayrıca, geliştirme

ve üretim süreçlerinde bu standartların kullanılmasıyla, onları kullanacak firmaların piyasa olanaklarının genişletilmesi de sağlanmış olacaktır.

- Gerekliliklerin proje başlangıcında tam olarak belirlenmesi, üretim süreci sonunda kaliteli ürünlerin kullanıcıya sunulmasını sağlayacaktır. Yazılım içerikli sistemlerin tedarikinde gerekliliklerin tam olarak belirlenmiş olması, geliştirme ve üretim sürecini basit bir mühendislik süreci haline dönüştürmekte, bu ise yazılımın kendine özgü sorunlarının büyük bir kısmının ortadan kalkmasını sağlamaktadır. Bu kapsamda, sisteme ilişkin gerekliliklerin belirlenmesi aşamasında, tedarik makamı-kullanıcı-yüklenici arasındaki iletişimin çok iyi kurulmuş olması ve bununla birlikte kullanıcının kendi ihtiyaçlarını çok iyi biliyor olması, sistemden beklenen performansın sağlanması açısından oldukça önem taşımaktadır.

- Gerekliliklerin tam olarak belirlenemeyecek nitelikte olduğu durumlarda, söz konusu sistemlerin tedarikinde ve geliştirme-üretim sürecinde bir takım özel yöntemlerin uygulanması zorunluluğu ortaya çıkmaktadır. Böyle bir durumda, tedarik ve üretim şekli evrimsel nitelikler içermek durumundadır. Projenin evrimsel özellikler içermesi, ürünün geliştirilmesinin, gerekliliklerin olgunlaşmasıyla birlikte gerçekleştirilmesini sağlayacaktır. Aynı zamanda, süreçlerin esnek olması, belirlenmiş gerekliliklerde meydana gelebilecek değişikliklerin sisteme sorunsuz aktarımını da sağlayacaktır. Bu şekilde, yazılım geliştirme sürecinin kendine özgü olumsuz etkileri azaltılmış olacaktır.

- Yazılımın kendine özgü yapısı, hızlı değişen bir teknoloji içerisinde yer alması, insan unsurunun yoğun olması, çok fazla sayıda kullanıcının var olması gibi nedenler, yazılımı ve bunun geliştirilmesini riskli hale getirmektedir. Risk unsurunun varlığı, yüksek maliyetleri ve zaman aşımını da beraberinde getirmektedir. Bu çerçevede risk unsurlarının elimine edilmesi, projenin belirlenmiş kalite, zaman ve maliyetlerde tamamlanması açısından büyük önem taşımaktadır. Risk unsurlarının ortadan kaldırılması, ancak bu unsurların belirli bir sistematik çerçevesinde yönetilmesi ile mümkün olmaktadır. Bu kapsamda risklerin tanımlanması, analiz edilmesi ve yürütülmesi gerek şart olarak algılanmalıdır. Bu bağlamda, yazılım geliştirme süreçlerinin risk yönetimi odaklı olarak belirlenmesi, istenen ürünün kullanıcıya sunulması olasılığını artıracaktır.

- Günümüzde her türlü silah sisteminin içerisinde yazılım unsurunun yer alması artık bir zorunluluk halini almıştır. Yönlendirme, yön bulma, hedefleme, tanıma gibi her türlü faaliyet bu görevler için geliştirilmiş olan bir takım yazılımlar

sayesinde gerçekleştirilmektedir. Yazılım unsurunun her bir sistemin içinde yer alıyor olması, söz konusu sistemlerin tedariki ile sorumlu organizasyonlarda yazılım ve yazılım geliştirme süreçlerinde uzman personelin yer almasını beraberinde getirmektedir. Bu kapsamda, matris organizasyonların oluşturulma gerekçesine uygun olarak, her bir uzmanın etkin kullanımının sağlanması amacıyla yazılım uzmanlarının ayrı bir birim içerisinde örgütlenmesi gerektiği değerlendirilmektedir. Bu şekilde, her bir projede yer alan yazılım unsurları, temel politika ve stratejiler çerçevesinde standart bir şekilde yönetilmiş olacak ve her bir sistem için büyük bir risk unsuru olan yazılımın, risk unsuru olma etkisinin azaltılması sağlanmış olacaktır.

## KAYNAKLAR

Ahuja, V. ve V. Thiruvengadam. "Project Scheduling and Monitoring: Current Research Status", Construction Innovation, 2004, 4: 19-31.

Alterkawi, Mazyed, Application of GIS in Transportation Planning, The Case of Riyadh, The Kingdom of Saudi Arabia, King Saud University, 2002.

Anameriç, H., "Bilgi Merkezlerinin yönetiminde Bilgi Sistemlerinin Rolü", Bilgi Dünyası 2005, 6(1) ss. 15-35.

Anthony, Yeh, Dijital Urban Planning The Use of GIS in Urban Planning Gar, , On, Centre of Urban Planning and Environmental Management The University of Hong Kong , 2002.

Bacon, C.J. & Fitzgerald, B. (2001) A Systemic Framework for the Field of Information Systems. The Database for Advances in Information Systems, 32 (2) Spring, s. 46-67.

Balmohan N, SP, Enterprise GIS Approach for Urban Transportation Planning) Resources Pte, Ltd, Singapore, 2002.

Barkley, B.T., "Project Risk Management", 1<sup>st</sup> edition, McGraw-Hill Book Company, 2004.

Barutçugil, İsmet. "Proje Yönetimi", Kariyer Yayıncılık İletişim, Eğitim Hiz.Ltd.Şti., 1'inci Baskı, 2008.

Bass, L., Clements, P. ve Kazman, R., "Software Architecture in Practice", Addison Wesley, 2003. s.96.

Battersby, A., "Network Analysis for planning and scheduling", 3rd Ed., Macmillan, 1970.

Benhaim, Y. ve A. Laufer. "Robust Reliability Of Projects With Activity Duration Uncertainty", Journal of Construction Engineering and Management, 1998, 124 (2). ss.125-32.

Bingi, P., Sharma M.K. ve Godla J.K. "Critical Issues Affecting an ERP Implementation," Information Systems Management, 1999, C.16. S.3. s.7.

Burke, Rory. Project Management and Planning & Control Techniques (3th Ed) USA, John Wiley, 1999.

Cervone, H. Frank "Project risk management", OCLC Systems & Services, Vol. 22 (4), 2006, s. 256-262.

Chenfu,Liao, Fusion of Transportation and Traffic Modeling with Urban Design and GIS Morris, Ted, Manager, Intelligent Transportation Systems Laboratory Center For Transportation Studies., 2002.

Chevalier, N.J. ve RUSSELL A.D. "Developing A Draft Schedule Using Templates And Rules", Journal of Construction Engineering and Management; 2001,127 (5), ss. 391-98.

Choo, H.J., Tommelein, I.O. Baliard, G. ve Zabelle T.R. "Workplan: Constraint-Based Database For Work Package Scheduling", Journal of Construction Engineering and Management, 1999, 125 (3), ss.151-160.

Christopher Martin, ve Philip Powell, Information Systems, London: McG raw-Hill , 1991.

Chou, D.C., Tripuramallu, H.B., ve Chou, A.Y. "BI and ERP integration", [www.emeraldinsight.com/0968-5227.htm](http://www.emeraldinsight.com/0968-5227.htm) , 2009.

Cottrell, W. D. "Simplified Program Evaluation and Review Technique (PERT)", Journal of Construction Engineering and Management, 1999, 125 (1), ss. 16-22.

Davenport, T. H. (1998). Putting the Enterprise Into The Enterprise System,.Harvard Business Review, July/ August, ss. 121-131.

Dawson, R.I. ve Dawson C.W. "Practical Proposals for Managing Uncertainty and Risk in Project Planning", International Journal of Project Management, 1998, (5), s.299.

Denis, H. "Is The Matrix Organization A Cumbersome Structure for Engineering Projects", Project Management Journal, 1986, 17 (1), ss. 49-55

Dengiz, G.M., Erceiş, M.A., Karadağ, O. ve Şahmalı, E., "Proje Yönetimi Bilgi Kitabı", Proje Yönetim Derneği (PYD), 1998.

- Düzakın E. ve S. Sevinç (2002) "Kurum Kaynak Planlaması" Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi Cilt XXI, Sayı-1, 2002, ss.189-218,
- Easa, Said and Chan, Yupa, Urban Planning and Development Applications of GIS, 2000.
- Eldin, N.N. "Concurrent Engineering: A Schedule Reduction Tool", Journal of Construction Engineering and Management, 1997, 323 (3), ss.354-62.
- Elibol, H., "Bilişim Teknolojileri Kullanımının İşletmelerin Organizasyon Yapıları Üzerindeki Etkileri" Selçuk Üniv. Sosyal Bil. Dergisi, Sayı 13, 2005, s.158)
- Faniran, O.O., J.O. Oluwoye, ve Lenard. D.J. "Interactions Between Construction Planning And Influence Factors", Journal of Construction Engineering and Management, 1998, 124 (4), ss.245-56.
- Ferguson, S. ve Whitelaw, M., "Computerized management information systems in libraries", Australian Library Journal, 41 (3 August), 1992, S. 184-198
- Fischer, M.A. ve F. Aalami. "Scheduling With Computer-Interpretable Construction Method Models", Journal of Construction Engineering and Management, 1996, 122 (4), ss.337-47
- Ford, C.R., Randolph W.A. "Cross-Functional Structures: A Review and Integration of Matrix Organization and Project Management", Journal of Management, Vol.18, NO.2, 1992, ss. 267-294.
- Gray, F.C., E. Larson. Project Management: The Managing Process, USA, McGraw Hill, 2000.
- Gulledge, "What is integration?", [www.emeraldinsight.com/0263-5577.htm](http://www.emeraldinsight.com/0263-5577.htm), Industrial Management & Data Systems, Vol. 106 No. 1, 2006, ss. 5-20.
- Güllü, H. (2001). "Dinar'ın zemin büyütmelerine göre coğrafik bilgi sistemleri ile mikrobölgelemesi" Doktora Tezi. İTÜ Fen Bilimleri Enstitüsü, 2000.
- Güllü, H., Baykasoğlu, A. ve Dereli, T., GIS ve mühendislikte kullanım alanları. Otomasyon dergisi, Eylül 2003, s.166-172, 2003.
- Haag, S., Cummings, M. ve Dawkins, J., "Management information systems for the information age", Boston: Mc Graw-Hill Co. Inc. 1998, s.197.

Halaç, O. Kantitatif Karar Verme Teknikleri (3ncü baskı), İstanbul, Evrim Basım Yayım Dağıtım, 1991.

Harrington, H.J.,Conner D.R., Horney N.L. "Project Change Management", NY, McGraw Hill, 2000.

Harris, R.B. ve Ioannou, P.G. "Scheduling Projects With Repeating Activities", Journal of Construction Engineering and Management, 1998, 124 (4), ss.269-78

Hwang Heung-Seik, Stochastic Set-Covering Model for Public Facility Location Based On GIS. Department of IE,Dongeeui-University,Busan KOREA, 2001.

İşlek, Kurtuluş. "Yazılım Tedariki ve Yazılım Geliştirme Yöntemleri", MSB Uzmanlık Tezi. Ankara, Kasım 2000.

Joyce, W. F. "Matrix Organization: A Social Experiment", Academy of Management Journal. 1986.29 (3): s. 536-561

Kalıpsız, Buharalı ve Biricik, Sistem Analizi ve Tasarımı: Nesneye Yönelik Modelleme, Papatya Yayıncılık, 2006. s. 25.

Karakanian, M. "Choosing an ERP Implementation Strategy," Year 2000 Practitioner, 1999, C.2. S.7. s.1.

Kerzner, Harold. Project Management (6th Ed.), Canada, John Wiley and Sons Inc., 1998.

Kerzner, Harold. "Project Management: A Systems Approach to Planning, Scheduling, and Controlling" (8th Ed.), John Wiley and Sons Inc., 2003.

Kumar, K., ve Hillegersberg, J. ERP Experience and Evolution Communications of the ACM, 2000, 43(4), s. 23-26.

Lavold, Garry. Project Management Handbook (2nd Ed.), USA, John Wiley and Sons Inc, 1988.

Lee, H.S., Yi K.J. "Application of Mathematical Matrix to Integrate Project Schedule and Cosf,Journal of Construction Engineering and Management, 1999,125(5),s. 339.



Leek, C., "Information systems frameworks and strategy", *Industry Management & Data Systems* 97/3, MCB University Pres 1997, s. 86-89.

Lorenzen, Michael, "Change Management". *Mistakes in library management: Grievous errors and how to avoid them*. Scarecrow Press. 2009, s. 83-94.

Loudon, K.C., Loudon, J.P., "Management Information Systems: managing the digital firm, (8th Edition)", Prentice Hall, 2004.

Lucey, T., "Management information systems", Easleigh: DP Publications, 1987. s. 185-186).

McLeod, R., Schell, G, "Management Information Systems", 8. B., New Jersey, Prentice Hall Inc., 2001

Microsoft web sitesi, [www.microsoft.com/turkiye/dynamics/erp/erp\\_secimi.msp](http://www.microsoft.com/turkiye/dynamics/erp/erp_secimi.msp)  
21.06.2009.

Miller J. and Shaw Shih-Lung, *GIS-T Data Models*, 2001.

Mishra, G.C. *Subjective Evaluation Of Duration Of Construction Activities For Probabilistic Project Management*, Masters Thesis, School of Planning and Architecture, New Delhi. 1989.

Morris, W.G. Peter. "Managing Project Interfaces-Key Points for Success", *Project Management Handbook*, USA, Van Nostrand Reinhold Company, 1988.

Netsis ERP web sitesi, [www.netsis.com.tr](http://www.netsis.com.tr). 21.06.2009 tarihinde alıntılanmıştır.

O'brien, W.J. ve Fischer, M.A. "Importance of Capacity Constraints to Construction Cost and Schedule", *Journal of Construction Engineering and Management*, 2000, 126 (5), s. 366-73.

Pilcher, Roy., *Appraisal and Control of Project Costs*, McGraw Hill, Londra, 1973.

PMI. *A Guide to Project Management Body of Knowledge*, USA, Project Management Institute, 2000.

Posner, B. Z. "What's All The Fighting About? Conflicts In Project Management", *IEEE Transactions on Engineering Management*, 1986, 33 (4) s. 207-211.

Post, G.V. ve Anderson, D.L., "Management Information Systems", USA, McGraw-Hill, 2003; s. 507-543.

Que, B.C. "Incorporating Practicability into Genetic Algorithm-Based TimeCost Optimization", Journal of Construction Engineering and Management, 2002, 128 (2), s. 139-143.

Ritchie B. ve Marshall, D., Business Risk Management, 1st Ed., Chapman & Hall Inc., 1993.

Ross, J.. Surprising Facts about Implementing ERP. IT Professional, July-August 1999, s. 65- 68.

Royer P.S., "Project Risk Management A Proactive Approach", Management Concept Inc. Virginia, 2002

Sahoo, D. "A Dynamic Programming Approach to Time-Cost Trade Off, Masters thesis, School of Planning and Architecture. New Delhi, 1996.

Sekar S.P. GIS Applications for Urban Planning-A case Study of Tinidivanam Town Tamil Nadu ,Institute of Remote Sensing, Anna University Chennai, 2002.

Shi, J.J., Cheung S.O. ve Arditi D. "Construction Delay Computation Method", Journal of Construction Engineering and Management, 2001, 127 (1), s. 60-65.

Skipper H.D. Jr.. "International Risk & Insurance :An Environmental – Managerial Approach", 1<sup>st</sup> edition, McGraw-Hill Book Company, 1998.

Software Technology Support Center (STSC-US), (Yazılım Teknolojileri Destek Merkezi) ABD, "Guidelines for Succesfull Acquisition and Management of Software Intensive Systems Version 3.0", 2000.

Soyuer, H., "İşletmelerde Bilgisayar Destekli Bilgi Sistemi Uygulamaları ve Üretim/İşlemler Yönetiminde Bilgisayara Dayalı Sistemler", Yayınlanmamış Doçentlik Tezi, Ankara, 2000.

Tom, P.E, GIS – Transportation (STEA Manegement Systems Server Net Prototype Pooled Fund Study, Phase A-System Architectures Henderson, R.L.S. Co Principal Investigator, New Mexico State Highway and Transportation Department, 1995.

Türkiye Bilişim Derneği (TBD), "Bilişim Projeleri Yönetimi El Kitabı", Karayel Ajans, 2'nci Baskı, Ağustos 1999.

Wall, W.C. "Integrated Management in Matrix Organization", IEEE Transactions on Engineering Management, 1984, 31 (1), s. 30-36.

Wang, W.C., Demsetz, L.A. "Application Example for Evaluating Networks Considering Correlation", Journal of Construction Engineering and Management, 2000, 126 (6), s. 467-74.

Wolf, L.G. ve W.C. Hauck. "PERTICPM: A Tool For Managing Projects", Industrial Management, 27/1, 1985, s. 22-25.

Wright, N. H. "Matrix Management", Management Review, 1979, 68 (4), s. 58-61.

Wright, N.H. "Matrix Management-Fortifying the Organization Structure", Manufacturing World, 1980.

[http://www.yancy.org/research/project\\_management/integration.html](http://www.yancy.org/research/project_management/integration.html), 12.05.2009.

## **EK-1: Tanımlar, Kısa Adlar ve Kısaltmalar**

### **1. Tanımlar**

**1.1. ARA YÜZ:** Sistemler, sistemin bileşenleri, yazılım modülleri, yazılımlar, donanımlar, donanım bileşenleri, bilgisayarlar, bilgisayar ağları ve hepsinin varsa insan ile arasındaki etkileşimi, iletişimi düzenleyen, mantıksal veya fiziksel, araçlar, ürünler, protokollerdir.

**1.2. AKTİF İZ TAKİBİ:** Araç İz Takip Sistemine dâhil hareket halindeki araçların konumuna (enlem, boylam ve yükseklik) ilişkin izlerinin, bu araçlar üzerinde bulunan iz takip kitleri kullanılarak takibi.

**1.3. ALT SİSTEM:** Bir yazılım sisteminin en üst düzeydeki bileşenlerine alt sistem denir. Sistem kapsamında alt sistem tek başına çalıştığında asgari bir temel işlevi yerine getirebilecek bir uygulama yazılımı bileşeni olarak değerlendirilebilir.

**1.4. AYRINTILI YÜKLEME PLANLAMASI:** Özel olarak belirlenmiş yük/yüklerin (gerektiğinde yolcu ile taşınacak birime), yine özel olarak belirlenmiş olan bir ulaştırma aracına, her bir yükün araç içerisinde yerleştirileceği platform, güverte (deck) vs. üzerindeki yerinin koordinatlarına varıncaya kadar yükleme şeklinin ayrıntılı olarak planlanması ve bindirilecek/yüklenecek yolcu, araç ve malzemelerden, belirlenen ulaştırma aracı ile taşınamaz nitelikte olanların belirlenmesidir.

**1.5. BİLGİ (INFORMATION):** Belli bir işlemde geçirilmiş, kullanılır, yorumlanabilir, sonuç doğuran veridir.

**1.6. BİLGİ SİSTEMİ:** Kurumun verimliliğini ve verdiği hizmetin niteliğini yükseltmek, yönetimine denetim ve karar destek olanakları sunmak üzere yararlanılan, kuruma ilişkin bilginin yaratılması, saklanması, işlenmesi ve iletilmesini konu eden donanım ve yazılım nitelikli sistemlerin oluşturduğu, yönetim kademelerine bilgi ve karar desteği sağlayan yapının tümü.

**1.7. BİRİKİM (KNOWLEDGE):** Bilgi ve kullanım sonuçlarının arasındaki ilişkilerin kurulması neticesinde, belli koşullar altında ve belli varsayımlara dayanarak, belli senaryolar dahilinde öğreti haline gelmiş kurallardır.

**1.8. COĞRAFI BİLGİ SİSTEMİ (CBS):** Coğrafi bilgilerin depolanması, modellenmesi, analiz edilmesi ve raporlanması için gerekli yazılım, coğrafi veriler ve uygulamalar bütünüdür.

**1.9. ÇATIŞMALARIN GİDERİLMESİ:** Aynı seviyedeki planların karşılaştırılması sonucu ortaya çıkan planlama hatalarının yapılacak manuel müdahalelerle giderilmesi işlemidir.

**1.10. DOĞRULAMA (VERIFICATION):** Bir ürünün (üretilmiş mal, hizmetin);

- Ürünü oluşturması öngörölmüş işlemin hedeflediği niteliklere uygunluğunu,
- ürünün bir başka işlemde de kullanılacak olması durumunda, müteakip işlemin gerektirdiği niteliklere sahip olduğunu,
- ürünü oluşturan işlemlerin, daha önce tanımlandığı şekilde icra edildiğini,
- Gerek ürünün gerekse de üretilme sürecinin kendi içinde doğruluk, bütünlük, uyumluluk, tamamlık, belirginlik gibi hususları sağladığını,
- İlgili standart ve kurallarına uyulduğunu denetlemek, incelemek, hatalarını ve eksiklerini tespit etmek işlemidir.

**1.11. DOKÜMANTASYON:** Bilgi sistemleri, sistem analizi, tasarım işlemleri için gerekli, işletim, bakım-onarım, eğitim konularında kullanılmak üzere hazırlanmış her türlü kayıt medyası, disk, disket, CD-ROM, kitap, basılı form, not, resim, şekil, plan, kullanım kılavuzu, broşür vb.dir.

**1.12. DONANIM:** Projenin hedeflediği ürünü, hizmeti oluşturmada kullanılan veya bizzat o ürünün üzerinde çalışacağı, çalışırken yararlanacağı fiziksel araçlardır.

**1.13. EMNİYET:** Bir proje elemanının, ürünün, mahallin, aracın, vb. unsurların, içinde bulunduğu ortamda oluşabilecek olaylar neticesinde maruz kalabileceği fişlevsel, fiziksel zarar ve tehlikelere karşı alınacak tedbirlerdir.

**1.14. ENVANTER:** Taşınması istenen her türlü birimin genel adıdır. Taşınacak malzeme (her türlü alet, araç, gereç, vb.) ve yolcudan oluşur. Bir ağırlık veya hacim olarak belirlenmiş yolcu da ihtiyaç duyulduğunda taşımaya esas malzeme olarak kabul edilir. Yazılım, envanterin daha çok taşımaya esas ağırlık, hacim (en, boy, yükseklik), miktar, kendi kendini veya başka bir malzemeyi taşıyabilirlik, tehlikeli madde olup, olmadığı gibi hususlar ile ilgilenecektir.

**1.15. ENVANTER YÖNETİMİ:** Malzeme ile ilgili planlama, programlama, bütçeleme, stok kontrol, satın alma, ikmal, bakım, onarım vb. faaliyetlerinin yapılmasının ve kontrolünün genel adıdır.

**1.16. G GÜNÜ:** Görevlendirilmiş taşıma birimlerinin taşıma planı içinde taşımaya başla olarak aldıkları gün. (Örn. X biriminin taşıma başlama günü G+5 günü gibi)

**1.17. GAUGE:** Tren yolunda raylar arasındaki açıklık miktarıdır

**1.18. GEÇERLEME (VALIDATION):** Üretilen bir mal veya hizmetin, o mal ve hizmete başlangıçta duyulan ihtiyaçlarını ve kullanım niyetlerini karşılama derecesinin gerçek verilere dayalı olarak tespitidir.

**1.19. GELİŞTİRİCİ (YÜKLENİCİ):** Yazılımın Kullanıcı istek ve beklentileri doğrultusunda geliştirmesi maksadıyla geliştirme projesini üstlenmiş firma/kurum/kuruluş vb.dir.

**1.20. GELİŞTİRME MODELİ:** Bir ürünün geliştirilmesinde izlenecek süreci tanımlayan modeldir.

**1.21. GENEL (KABA) YÜKLEME PLANLAMASI:** Belirli bir görev için taşınacak birim envanterini oluşturan yolcu, araç ve ana malzemelerin (yükleme kapasitesi) belirli bir tipteki kaç adet ulaştırma (örn. Kargo Uçağı veya RORO Gemisi) araç(lar)ına, her bir yükün araç içerisinde yerleştirileceğı kasa, platform, güverte (deck) vs. üzerindeki yeri önemli olmaksızın yüklenebileceğinin/tahsis edilebileceğinin ve/veya belirli bir tipteki ulaştırma araç(lar)ının kaç sorti/seferi ile taşınabileceğinin planlanması ve bu yolcu, araç ve malzemelerinden, seçilmiş olan belirli tipteki ulaştırma araç(lar)ı ile taşınamaz nitelikte olanların belirlenmesidir.

**1.22. GÜVENLİK:** Bir ürünün, o ürünü oluşturan bileşenlerin sahip olduğu özellikleri, o ürünün geliştirilmesinde kullanılan verinin, bilginin, birikimin yazılı, sayısal, fiziksel durumlarına sadece yetkili kişilerce erişilmesini, kopyalanmasını, kullanılmasını, yararlanılmasını sağlamaya yönelik olarak alınacak tedbirlerdir. Personelin yetkilendirilmesi öncesinde erişim hakkına sahip olması için gerekli niteliklerinin/yetkilerinin olup olmadığının araştırılması ve sonucun belgelenmesi faaliyetlerini de kapsar.

**1.23. KULLANICI:** Gerçekleştirilecek projeye konu olan problemi yaşayan, ihtiyacı duyan, geliştirilecek mal ve hizmeti kullanacak veya kullandıracak olan kurum, kuruluş, birim, makam veya kişidir.

**1.24. KURYE/RİNG BACAĞI:** Bir kurye/ring seferinin icra edildiği güzergâhta seferi icra eden ulaştırma aracının uğrayarak yükleme ve boşaltma yaptığı iki durak arasına verilen isimdir. Çıkış noktası ile ilk durak ve son durak ile varış noktası arası da bacak olarak değerlendirilir.

**1.25. KURYE/RİNG ETKİNLİK KRİTERİ:** Bir kurye/ring seferinin belirli bir dönem boyunca etkin kullanıldığıнын ölçülebilmesi amacıyla olması gereken ortalama asgari doluluk oranı.

**1.26. KURYE/RİNG SEFERİ:** Belirli bir zaman dilimi içerisinde, belirli sıklıkta/frekansta tekrar edecek şekilde, aynı kara yolu, deniz yolu, hava yolu ve demir yolu güzergâh/rotasını kullanan, yolcu ve yüke bağımlı olmaksızın mükerrer taşıma faaliyetini icra eden ulaştırma seferleridir. Ulaştırma planlamacısı için kurye/ring seferlerinin anlamı, yolcu/yük taşıma taleplerinin belli tarih ve güzergâhlar için birleştirilmesi ve münferit taşımalar için eldeki araçların tahsisi yanında yeni araç tedariki veya kiralanmasının asgari düzeye indirilmesidir.

**1.27. MODEL:** Bir sistemin, varlığın, olayın veya sürecin fiziksel, matematiksel veya mantıksal temsilidir.

**1.28. MÜNFERİT TAŞIMA İHTİYACI:** Ulaştırma planlamacısının, taşıma faaliyeti için bir ya da birden çok ulaştırma aracının tahsis edilmesine ihtiyaç duymadığı, taşıma planlamasını mevcut tarifeli ulaştırma seferleri (kurye/ring sistemi) ile belli bir güzergâha giden münferit yüklerin bir aracı dolduracak kadar toplanmasını müteakip tahsis edilen araç (ambar sistemi) seferlerinden yararlanması yöntemlerine yönelttiği taşıma ihtiyaçlarıdır (Acil ihtiyaç durumunda ihtiyacın maliyetine bakılmaksızın en seri yöntemle (mod) taşıma gerçekleştirilir)

**1.29. ONAYLAMA (ACCREDITATION):** Doğrulaması ve geçerlemesi yapılmış bir ürünün, bu safhalardan istenilen niteliklere uygun olarak geçmiş olduğunun resmi olarak tescili işlemidir.

**1.30. PLAN HİYERARŞİSİ:** Yapılan planların hazırlandıkları seviyeye göre bir plan ağacı şeklinde birbirleriyle ilintili olarak gösterildikleri hiyerarşik yapıdır.

**1.31. PASİF İZ TAKİBİ:** Fabrika, depo ve yükleme/boşaltma noktalarından taşınmak üzere ulaştırma terminallerine teslim edilen, araç ile taşınmakta olan ya da teslim alınan malzeme cins ve miktar bilgilerinin sabit iz takip kitleri (barkod okuyucu-yazıcıları v.b) kullanılarak takibi yöntemidir.

**1.32. REFERANS GÜN:** Referans günden kasıt faaliyetlerin referans alacağı bir tarihin belirlenmesidir. Bu, müşteri tarafından bizzat bildirilebileceği gibi müşterinin belirttiği CRD'ler esas alınarak da belirlenebilir. Deneme maksatlı planlamalarda bu tarih bilahare sisteme girilebilir. Her hangi bir referans gün girilmediği takdirde sistem varsayılan olarak referansı "G" günü olarak alacak ve bu günden önceki faaliyetleri G-20, ....., G-1 şeklinde, bu günden sonraki faaliyetleri G+1,.....G+20 şeklinde ifade edecektir.

**1.33. SİMÜLASYON :** Bir modelin zaman içerisindeki işleyişinin sanal olarak gösterimidir.

**1.34. SİSTEM:** Bir bilgi sisteminin, belli bir ortam için belirgin işlevleri olacak biçimde tasarlanıp, geliştirilmiş, bu ortamın işleyişinin bilgisayar destekli yürütülmesinde yararlanılan en üst düzeyli bileşendir. Bu bileşen kendi içinde çeşitli alt düzeylerde başka bileşenler içerir.

**1.35. STANDART:** Konuları ve ortaya çıkış nedenleri farklı olsa dahi, projelerin süreç, yönetim, dokümantasyon, organizasyon, görev, sorumluluk, yükümlülük, yetki ve kısıtlar açısından ortak bir temele oturtularak, denetlenmesi ve kontrolünü sağlamak ve eksik ve hataları tespit amacıyla oluşturulan şablonlardır.

**1.36. TABANDAN TEPEYE DOĞRU PLANLAMA:** Bu planlama yönteminde en üst birim gerçekleştirilecek taşıma faaliyetine ilişkin temel prensiplerini ortaya koyar ve ilgili ast birimlerince konuya ilişkin planlarını yapmaları istenir. Alınan temel prensipler ışığında en alt birimden itibaren planlama yapılır ve her bir üst seviyede bu planlar birleştirilerek ve çatışmaları giderilerek en üst birime kadar sistematik bir şekilde gönderilir. En üst birim nihai birleştirmeyi yapar ve sonuçta da ana planın ortaya çıkması sağlanır)

**1.37. TAŞIMA:** Taşınacak birimlere ait yolcu, araç, ve malzemelerin bir coğrafi noktadan diğer bir coğrafi noktaya ihtiyaç sahibinin en son istediği zaman (CRD-Customer Required Date) içerisinde; kara yolu, deniz yolu, hava yolu, demir yolu, iç su yolu ulaşım imkânlarından biri veya bir kaçından istifade edilerek bir bütün halinde ya da gruplara/parçalara/bileşenlere (component) ayrılarak naklidir.

**1.38. TAŞIMA BİLEŞENİ (Movement Component):** Envanterin bir arada taşınması zorunlu parçalarının belirlenerek bölünmeden taşınabilmesi ve farklı ulaştırma modlarının kullanımında esneklik sağlanabilmesi maksadıyla bir araya getirilmiş küçük envanter gruplarıdır. Yolcu ve envanterin; taşınmasının daha etkin



ve ekonomik olarak gerçekleştirilmesi amacıyla, anlamlı parçalar halinde [örn. Öncü (advanced party), deniz yolu ile taşınacak yük grubu (sea cargo party); sürücü ve operatörler grubu (ship meeting party), ana yolcu grubu (main pax party) vs.] değişik zaman ve ulaştırma modelleri ile taşınacak şekilde bölünerek (splitting) oluşturulan parçalarına denir.

**1.39. TAŞIMA BİRİMİ:** Her seviyedeki ulaştırma planlama, icra ve icranın izlenmesinden sorumlu ve taşımayı bizzat gerçekleştirecek ulaştırma aracına sahip olan ya da bu aracı sorumluluğuna almış birimdir.

**1.40. TAŞIMA BİTİŞ NOKTASI (TBN) :** Taşımanın tek bir ulaştırma modu kullanılarak gerçekleştirildiği bölümünün (örn. Kara yolu bacağı, deniz yolu bacağı gibi) bittiği noktaya denir (Örn.: deniz yolu bacağı için bir liman olabilir).

**1.41. TAŞIMA ÇIKIŞ NOKTASI (TÇN) :** Taşımanın tek bir ulaştırma modu kullanılarak gerçekleştirildiği bölümünün (örn. Kara yolu bacağı, deniz yolu bacağı gibi) başladığı noktaya denir (Örn.: Bir hava yolu bacağı için bir havaalanı olabilir).

**1.42. TAŞIMA GÖREVİ:** Her bir taşıma faaliyetidir.

**1.43. TAŞIMA PLANI:** Coğrafi bilgiler ışığında ihtiyaç duyulan taşımaların gerçekleştirilebilmesi için, icra edilmesi gerekli faaliyetlerin ve kullanılacak ulaştırma vasıtaları ve alt yapılarının (kara, deniz, hava, demir ve iç su yolu ulaştırma alt yapıları) belirlenmesi, zamana dayalı olarak önceliklendirilmesi ve uygulanabilir hale getirilmesini sağlayan plandır. Taşınması istenen birimlerin, istenilen yerde, istenilen zamanda (CRD-Customer Required Date) bulundurulması amacıyla, taşıma önceliklerinin belirlenmesi, bölge müdürlüklerince eş zamanda kullanılması planlanmış ulaştırma altyapısı ve ulaştırma araçlarına ilişkin plan çatışmalarının giderilmesi ve Kullanıcının ulaştırma ihtiyacının belirlenmesi amacıyla hazırlanan taşıma planlarıdır.

**1.44. TAŞIMA PLANI ÇATIŞMASI:** Birden çok birimin taşıma planlarında, aynı veya birbirine çok yakın zaman diliminde, aynı ulaştırma araç veya altyapılarının kullanılmasının planlanmış olmaları durumudur.

**1.45. TAŞIMA KOORDİNASYON MERKEZLERİ:** Ulaştırma birimlerinin ulaştırma ve taşıma faaliyetlerinin planlaması, planların icrası, icranın takip ve kontrolünden sorumlu merkezlerdir. Genelde Genel müdürlükler ve bölge müdürlükleri seviyesinde tesis edilir.

**1.46. TAŞIMA MODU:** Kara yolu, deniz yolu, hava yolu, demir yolu ve iç su yolu gibi ulaştırma faaliyetlerinin farklı altyapı, araç vb. kullanılarak yürütüldüğü modellerin her birine verilen genel isimdir.

**1.47. TAŞIMAYA ESAS HALE GETİRME:** Araç/Malzeme/Ekipmanların kullanıldıkları zamanki ebatları ile taşındıkları zamanki ebatları birbirinden farklı olabilmektedir. Bunlar literatürde "İşletim Ebatları (Operational Dimensions, OD) ve "Taşıma Ebatları (Shipping Dimensions, SD) olarak ifade edilmektedir. Envanterin işletim ebatlarının taşıma ebatlarına dönüştürülmesine (envanterin taşınabilecek şekilde paketlenmesi ya da taşınacağı ebatlara indirgenmesine) taşımaya esas hale getirme denir.

**1.48. TAŞINACAK BİRİM:** Taşınmak üzere sisteme tanımlanmış ve bölünmemiş her bir envanter ana grubudur.

**1.49. TEPEDEN TABANA DOĞRU PLANLAMA:** Zaman kısıtının bulunduğu durumlarda taşımadan sorumlu en üst birimce gerçekleştirilen ve ast birimlerden herhangi bir veri alınmadan eldeki mevcut verilerle gerçekleştirilen planlama türüdür. Bu plan türünde tek bir lojistik (taşıma) plan bulunduğundan plan hiyerarşisi oluşturulmamaktadır.

**1.50. TOPLU TAŞIMA İHTİYACI:** Ulaştırma planlama faaliyetinin; yolcu ve yük miktarının bir ya da birden çok ulaştırma aracının tahsis edilmesini gerektirdiği ve ihtiyacın mevcut tarifeli ulaştırma seferleri (kurye/ring sistemi)'nden yararlanılarak karşılanmasının etkin olmayacağı ve/veya birim bütünlüğünü bozacağı durumlara ilişkin olarak belirlenen taşıma ihtiyaçlarıdır.

**1.51. TRIM:** Yüklemeden sonra gemilerin öne ve arkaya yatıklık denge durumu

**1.52 ULAŞTIRMA RESMİ:** Her seviyedeki taşıma biriminin etki alanı içerisindeki herhangi bir taşımaya tahsis edilmemiş ve/veya tahsisi ve yükleme planlaması yapılmış ulaştırma araçlarının boş kapasite durumları ile icra edilmekte olan taşıma faaliyetlerinin konum bilgilerinin, ulaştırma araçları tarafından taşınmakta olan yolcu ve yük bilgilerinin tümünün veya bir kısmının gerçeğe yakın zamanlı veya planlama açısından kıymetten düşmeden önceki zamanda görülen resmidir.

**1.53 VERİ (DATA):** Bir sürecin işletilmesinde, bir fonksiyonun yerine getirilmesinde girdi olarak kullanılan ham bilgidir. İşlemin çıktısı işlenmiş bilgi olabileceği gibi, müteakip fonksiyonun icrası için bir ham bilgi de olabilir.

**1.54. YALPA AÇISI:** Yüklemeden sonra gemilerin sağa ve sola yatıklık denge durumu

**1.55. YAZILIM (SOFTWARE):** Bir bilgi işlem sisteminin işleyişi ile ilgili bilgisayar programlarının, yordamlarının, kurallarının ve gerektiğinde belgelemenin tümüdür.

## 2. Kısa Adlar ve Kısaltmalar

|           |   |
|-----------|---|
| 3PL       | Third Party Logistics (Üçüncü Taraf Lojistiği)  |
| ABM       | Araç, (Taşınacak) Birim, Malzeme  |
| ACWP      | Actual Cost of Work Performed (Gerçekleştirilen İş Gerçek Değeri)                     |
| ANSI      | American National Standards Institute   |
| ASA       | Anahtar Süreç Alanları  |
| ASP.NET   | .NET Active Server Pages (.NET Aktif Hizmet Sayfaları)                                |
| BAC       | Budget At Completion (Tamamlama Bütçesi)  |
| BCWP      | Budgeted Cost for Work Performed (Gerçekleştirilen İş Bütçeleme Değeri)               |
| BCWR      | Work Remaining (Kalan İş)   |
| BCWS      | Budgeted Cost for Work Scheduled (Programlanmış İş Bütçeleme Değeri)                  |
| BDUF      | Big Design Up Front (Büyük Tasarım Önermesi)  |
| BR        | Budget Remaining (Kalan Bütçe)  |
| BSO       | Business System Options (İş Sistemi Seçenekleri)                                      |
| C3 System | Crysler Comprehensive Compensation System (Crysler Kapsamlı Ödeme Sistemi)            |
| CALS      | Continuous Acquisition and Life-Cycle Support (Sürekli Tedarik ve Ömür Devir Desteği) |
| CASE      | Computer Aided Software Engineering (Bilgisayar Destekli Yazılım Mühendisliği-BDYM)   |
| CBS       | Coğrafi Bilgi Sistemi (GIS-Geographical Information System)                           |
| CDR       | Critical Design Review (Kritik Tasarım Gözden Geçirmesi)                              |

|               |  |
|---------------|--|
| CD-ROM        | Compact Disk – Read Only Memory (Bilgi Tekeri-Sadece Okunur Hafıza)                        |
| CLR           | The Common Language Runtime (Ortak Programlama Dili Geliştirme Ortamı)                     |
| CMM           | Capability Maturity Model (Yetenek Olgunluk Modeli)  |
| CMMI          | Capability Maturity Model Integration (Yetenek Olgunluk Modeli Entegrasyonu)               |
| CMMI-ACQ,     | CMMI for Acquisition (Tedarik YOME)  |
| CMMI-DEV      | CMMI for Development (Geliştirici YOME)  |
| CMMI-SVC      | CMMI for Services (Hizmet YOME)  |
| COCOMO        | Constructive Cost Model (Yapısal Maliyet Modeli)   |
| CPIEfficiency | Cost Performance Index for Efficiency (Maliyet Performans Etkinlik İndeksi)                |
| CRD           | Customer Required Date (Taşınacak birimlerin son noktada bulunmaları gereken en geç zaman) |
| CV            | Cost Variance (Maliyet Değişimi)   |
| DFD           | Data Flow Diagram (Veri Akış Grafiği)  |
| DFM           | Data Flow Modelling (Veri Akış Modellemesi)  |
| DO            | Domain Analysis (Ana Odak Analizi)   |
| EAC           | Estimate At Completion (Tamamlama Tahmini)   |
| ELD           | Integrated Logistics Support (Entegre Lojistik Destek)                                     |
| EM            | Entity/Event Modeling (Birim/Faaliyet Modelleme)   |
| ERD           | Entity-Relationship Diagrams (Varoluş-İlişki Diyagramları)                                 |
| ETARİTM       | Etkin Araç İz Takip Modülü   |
| EV            | Earned Value (Kazanılmış Değer)  |
| FCC           | Federal Communications Commission (Federal İletişim Komisyonu)                             |
| FCL           | The Framework Class Libraries (Bütünleşik Hazır İşlev Kütüphanesi)                         |
| FIFO          | First in Frist out (İlk giren ilk Çıkar)   |

|                |  |
|----------------|--|
| FILO           | First in Last out (İlk giren son çıkar)  |
| FIR            | Flight Information Route (Uçuş Bilgilendirme Rotası)   |
| FSD            | Functional Specifications Document (İşlevsel Özellik Tanımlama Dokümanı)   |
| GPS            | Global Positioning System (Küresel Konumlama Sistemi)  |
| GPRS           | General Packet Radio Service   |
| IDEs           | Integrated Design Elements (Bütünleşik Tasarım Elemanları)   |
| IEC            | International Electrotechnical Commission (Uluslararası Elektroteknik Komisyon)  |
| IEEE           | Institute of Electrical and Electronics Engineers (Elektrik Elektronik Mühendisleri Enstitüsü)                                 |
| IEEE/EIA 12207 | Standard for Information Technology-Software Life Cycle Processes (Bilgi Teknolojileri Yazılım Ömür Devir Süreçleri Standardı) |
| ISA            | Industry Standart Architecture (Standart Endüstri Mimarisi)  |
| ISO 12207      | Standard for Software Life Cycle Proseses-(Yazılım Ömür Devir Süreçleri Standardı)   |
| ISO            | International Standards Organization (Uluslararası Standartlar Organizasyonu)  |
| IT             | Information Technology (Bilgi Teknolojileri)   |
| ITU            | International Telecommunications (Union Uluslar arası İletişim Birliği)  |
| JTS            | Joint Technical Subcommittee (Müşterek Teknik Alt Komite)  |
| KDS            | Karar Destek Sistemi.  |
| KPAs           | Key Process Areas (Anahtar Süreç Alanları)   |
| KURPM          | Kurye/Ring Planlama Modülü   |
| LDS            | Logical Data Structure (Mantıksal Veri Yapısı)   |
| LDM            | Logical Data Modeling (Mantıksal Veri Modellemesi)   |
| LIFO           | Last in First Out (Son giren ilk çıkar)  |
| LILO           | Last in Last out (Son giren son çıkar)   |

|         |   |
|---------|---|
| LOC     | Line of Communication (Ulaştırma/İletişim Hattı)  |
| MSF/SDD | Microsoft Solutions Framework / Software Development Discipline   |
| MTA     | Milestone Trend Analysis (Kilometre Taşı Trendi Analizi)  |
| NMEA    | National Marine Electronic Association  |
| OBDC    | Open Data Base Connectivity   |
| OBİ     | Otomatik Bilgi İşlem  |
| OLE     | Object Linking and Embedding  |
| OPP     | Object-Oriented Programming (Nesne Tabanlı Programlama)   |
| OSI     | Open System Interconnection   |
| PB      | Project Board (Proje Kurulu)  |
| PID     | Project Initiation Document (Proje Başlatma Dokümanı)   |
| PM      | Project Manager (Proje Yöneticisi)  |
| PMI     | ABD Proje Yönetim Enstitüsü (Project Management Institute)  |
| PPRP    | Post-Project Review Plan (Proje Sonrası Gözden Geçirme Planı)   |
| PRINCE  | Project in Controlled Environment (Kontrol Edilebilir Ortamda Proje Yönetimi)                             |
| PMBOK   | Project Management Body of Knowledge (Proje Yönetimi Bilgi Ünitesi)                                       |
| RDD     | Responsibility Driven Design (Sorumluluğa Dayalı Tasarım)   |
| REK     | Raporlanabilir Envanter Kodu  |
| RUP     | Rational Unified Process (Rational Firması Tekleştirilmiş Süreci)   |
| SE-CMM  | Software Engineering CMM (Yazılım Mühendisliği YOM)   |
| SEI     | US Dept. of Defense Software Engineering Institute (ABD Savunma Bakanlığı Yazılım Mühendisliği Enstitüsü) |
| SLAs    | Service Level Agreements (Sözleşme Hizmet Seviyeleri)   |
| SNMP    | Simple Network Management Protocol (Basit Ağ Yönetim Protokolü)   |
| SPICE   | Software Process Improvement Capability dEtermination (ISO  |

|               |   |
|---------------|---|
|               | 15504)  |
| SQA           | Software Quality Assurance (Yazılım Kalite Garantisi)   |
| SPIEfficiency | Schedule Performance Index for Efficiency (Program Performans Etkinlik İndeksi)   |
| SRA           | Software Requirements Analysis (Yazılım İhtiyaçları Analizi)  |
| SRS           | Software Requirements Specification (Yazılım İhtiyaç Özelliklerinin Belirlenmesi)   |
| SSADM         | Structural System Analysis & Design Methodology (Yapısal Sistem Analizi ve Tasarımı Metodolojisi)                         |
| SV            | Schedule Variance (Program Değişimi)  |
| ŞGK           | Şahıs Güvenlik Kleransı   |
| TBN           | Taşıma Bitiş Noktası  |
| TBZ           | Taşınma Başlama Zamanı  |
| TCDD          | T.C. Devlet Demiryolları  |
| TCP/IP        | Transmission Control Protocol / Internet Protocol (İletim Kontrol Protokolü / İnternet Protokolü)                         |
| TÇN           | Taşıma Çıkış Noktası  |
| TDD           | Test Driven Development (Test Kaynaklı Geliştirme Modeli)   |
| TEU           | Twenty-foot Equipment Unit (20 Ayaklık Birim Malzeme/Konteyner)   |
| TPM           | Taşıma Planlama Modülü  |
| TTZ           | Taşınma Tamamlama Zamanı  |
| UC            | Use Case (Durum Çalışması)  |
| UCN           | Unit Control Number (Birim Kontrol Numarası)  |
| UML           | Unified Modeling Language (Birleştirilmiş Modelleme Dili)   |
| UP            | Unified Process (Tekleştirilmiş Süreci)   |
| URDAD         | Use Case, Responsibility-Driven Analysis and Design Method (Durum Çalışması, Sorumluluğa Dayalı Analiz ve Tasarım Metodu) |
| VAC           | Variance at Completion (Tamamlama Değişimi).  |

|         |  |
|---------|--|
| VTYM    | Veri Tabanı Yönetim Modülü                             |
| WBC     | Weight Bearing Capacity i (Ağırlık Taşıma Kapasitesi)  |
| WBS     | Work Breakdown Structure (Sınıflandırılmış İş Listesi) |
| YAMYOM  | Yazılım Mühendisliği-YOM                               |
| YASİYEB | Yazılım Süreç İyileştirme Yetenek Belirleme            |
| YOM     | Yetenek Olgunluk Modeli                                |
| YOME    | Yetenek Olgunluk Modeli Entegrasyonu                   |



## **EK-2: Yazılım Proje Geliştirme ve Yönetim Süreç, Aşama ve Adımları**

### **1. GENEL HUSUSLAR**

#### **1.1. Genel**

Yazılım geliştirme süreci, yazılım geliştirmede kullanılan standart bir taslak yapıdır. Bu süreçte meydana gelen olaylar ile icra edilen görev ve faaliyetlere yönelik yaklaşımları tanımlayan birçok model geliştirilmiştir.

#### **1.2. Süreçler ve Ara Süreçler**

Büyük çoğunluğu savunma sanayii için olmak üzere yazılım geliştirme şirketleri farklı süreç metodolojileri uygulamaktadırlar. Yazılım ömür devrinin seçimi, uygulanması ve izlenmesini esas alan metotların belirlenmesinde uluslar arası ISO 12207 (Yazılım Ömür Devir Süreçleri Standardı-Standard for Software Life Cycle Processes) ve IEEE/EIA 12207 (Bilgi Teknolojileri Yazılım Ömür Devir Süreçleri Standardı-Standard for Information Technology-Software Life Cycle Processes) standartları uygulanır.

ISO 12207, yazılımın geliştirilmesi ve idamesi için gerekli tüm işlemleri tanımlayan ve sistemin satın alma ve konfigürasyon hizmetleri esnasında uygulanacak süreç ve faaliyetlerle yazılım ömür devri sürecini ortaya koyan bir standarttır. Her bir süreç kendisine ilişkin bir çıktı seti vermektedir. ISO 12207 içinde 23 süreç, 95 faaliyet, 325 görev ve 224 çıktı yer almaktadır. Bu standart sayesinde; yazılım geliştirme sürecinde yer alan kullanıcı (alıcı), tedarikçi, ürün geliştirici, bakımçı, yönetici ve teknik personelin ortak bir dil kullanmasını sağlayan bir yapı oluşturulmaktadır. Bu dilde bütün süreçler çok iyi tanımlanmıştır. Kullanan kişinin ihtiyaçlarına cevap verebilecek şekilde esnek ve modüler bir yapıda kurgulanan standart, modülerlik (minimum tekrar ve maksimum entegrasyon) ve sorumluluk (her bir sürece bir sorumluluk yükleyerek bir çok kişinin hukuki olarak yer aldığı projelerde standartın kullanımını kolaylaştırma) olmak üzere iki temel prensibe dayanmaktadır. Süreçler, faaliyetler ve görevler seti yazılım projesine göre düzenlenebilmektedir. Bu süreçler, temel, destekleyici ve kurumsal olmak üzere üçe ayrılmaktadır. Destekleyici ve kurumsal süreçler, kurumdan ve gerçekleştirilen projeden bağımsız olmalıdır. Temel süreçler ise, duruma bağlı olarak değişebilmektedir.

Aynı niteliklere sahip IEEE 12207 standardında yer alan üç süreç kategorisi ve alt elemanları aşağıda sunulmuştur.

- İlk Ömür Devri Süreçleri
  - Satın alma Süreçleri
  - İkmal Süreçleri
  - Geliştirme Süreçleri
  - İşletim Süreçleri
  - Bakım Süreçleri
- Destekleyici Ömür Devri Süreçleri
  - Mali Kontrol Süreçleri
  - Konfigürasyon Yönetimi
  - Ortak Gözden Geçirme Süreci
  - Dokümantasyon Süreci
  - Kalite Sağlama Süreçleri
  - Problem Çözme Süreçleri
  - Doğrulama Süreçleri
  - Geçerleme Süreçleri
- Kurumsal Süreçler
  - Yönetim Süreci
  - Altyapı Süreci
  - Gelişim Süreci
  - Eğitim Süreci

### 1.3. Olgunluk Modeli (Maturity Model)

Olgunluk modeli, bir kuruluştaki olgunluk kriterlerini kesin hatlarıyla tanımlayan esaslara göre düzenlenmiştir. Bir olgunluk modelinde örneğin;

- Nereden başlanacağı,
- Kuruluşun önceki tecrübelerinden faydalanma esasları,

- Paylaşılan bir vizyon ve anlaşılabilir ortak bir dil oluşumu,
- Faaliyetleri önceliklendirmek için bir taslak oluşturulması,
- Kuruluş için gelişimin ne anlama geldiğini tanımlanması gibi bilgiler sunulabilir.

Olgunluk modeli, kıyaslama için bir nirengi noktası ve anlamaya yardımcı bir unsur olarak kullanılabilir (Örn.: Ortak özelliklere sahip farklı kuruluşların bu özelliklerinin kıyaslanmasında kullanılabilir). Yetenek Olgunluk Modelinde olduğu gibi kıyaslamada esas alınan nokta kuruluşun yazılım geliştirme süreçleri olabilir.

### **1.3.1. Yetenek Olgunluk Modeli (YOM) (Capability Maturity Model-CMM),**

Kurumsal süreçlerin anlaşılmasına ve tanımlanmasına yardımcı olan bir süreç modelidir.

YOM'da yapılan bağımsız değerlendirmelerle; kurumların ürettikleri yazılım veya süreçlerinin kalitesi değil, bu kurumların kendi tanımladıkları süreçleri ne kadar iyi takip ettiklerine ilişkin dercelendirme yapılmaktadır.

YOM, ilk olarak Watts Humphrey'nin "Yazılım Süreci Yönetimi (Managing the Software Process)" adlı kitabında (Addison Wesley Professional Basımevi, Massachusetts, 1989) tanımlanmış ve kitaptaki değerlendirmeler Phil Crosby'nin daha önceki çalışmalarına dayandırılmıştır. Önceleri sözleşmesi yapılmış bir yazılım projesi için yüklenici tarafından tanımlanmış süreçleri gerçekleştirme yeteneğinin objektif olarak değerlendirilmesi düşüncesiyle geliştirilmiş olan bu model, asıl gelişimini 1986 yılında Pittsburgh'daki Carnegie Mellon Üniversitesinde konuşlu SEI (ABD Savunma Bakanlığı Yazılım Mühendisliği Enstitüsü, US Dept. of Defense Software Engineering Institute) tarafından yapılan çalışmalarla kaydetmiştir. Model, yazılım geliştirme maksatlı hazırlanmış olmasına rağmen farklı alanlarda faaliyet gösteren kuruluşların yetenek olgunluk süreçlerinin anlaşılmasında da kullanılabilir. (Örn.: yazılım mühendisliği, sistem mühendisliği, proje yönetimi, yazılım bakımı, risk yönetimi, sistem tedarigi, bilgi teknolojileri (IT), personel yönetimi ve çoğunlukla da havacılık sistemleri (avionics) ve hükümet projelerinde kullanılmaktadır). YOM'un yerine daha sonra değişik bir versiyonu olan Yetenek Olgunluk Modeli Entegrasyonu (YOME-CMMI, Capability Maturity Model Integration) kullanılmaya başlanmış ve YOM'un adı Yazılım Mühendisliği-YOM (YAMYOM, SE-CMM, Software Engineering CMM)olarak değişmiştir.

Modelde, kuruluşlar için aşağıdaki beş olgunluk süreci seviyesi tanımlanmıştır:

- Birinci Seviye: Başlangıç Seviyesi (düzensiz, alt seviyede); yeni yazılımın kullanılmaya başlandığı noktadır.
- İkinci Seviye: Tekrarlanabilme (proje yönetimi, süreç disiplini); süreç tekrar tekrar kullanılabilir.
- Üçüncü Seviye: Tanımlı (Kurumsallaşmış) Seviye; sürecin iş süreç standardı şeklinde tanımlandığı ve doğrulandığı seviyedir.
- Dördüncü Seviye: Yönetilme (Ölçüm) Seviyesi; sürecin yönetim ve ölçme değerlendirme faaliyetlerinin icra edildiği seviyedir.
- Beşinci Seviye: Optimizasyon (Süreç İyileştirme) Seviyesi; süreç yönetiminde önceden düşünülmüş süreç optimizasyon ve iyileştirme faaliyetlerinin yürütüldüğü seviyedir.

Bu olgunluk seviyelerinin her birinde, bulunulan seviyeyi ifade eden “Anahtar Süreç Alanları (ASA-KPAs, Key Process Areas) yer alır. Her bir ASA için aşağıdaki beş tanım yapılması gereklidir.

- Hedef Tanımları
- Beklenti (Vaatler) Tanımları
- Yetenek Tanımları
- Ölçüm Tanımları
- Doğrulama Tanımları

YOM'un ikinci ve üçüncü seviyelerinde oluşturulan dokümanlara ilişkin bilgiler aşağıda sunulmuştur.

- **Yazılım Özellikleri Belirleme Dokümanı:** Onay mekanizması ve baş yöneticiden resmi onayın alınmasını müteakip neyin geliştirileceğinin ortaya konulduğu dokümandır. Bu doküman değiştirilebilir veya geliştirilebilir nitelikte değildir. Ancak, yazılım geliştirme sürecinin müteakip aşamasında yüklenici ile işbirliği içinde ilavelere müsaade edebilir.

- **Teknik Özellikler Dokümanı:** Geliştirmeye açık bir doküman olup, Yazılım Özellikleri Belirleme Dokümanında yer alan faaliyetlerin en özlü şekilde nasıl uygulanacağı bu dokümanda ortaya konmuştur.

- **Kod Giriş Gözden Geçirme (Peer Review of Code/Code Review):**

Kod girişi gözden geçirmesi yapılarak, yazılımı geliştiren personelin bir uygulamayı prova etmesi ve iyileştirme veya değişiklikleri teklif etmesi sağlanır. Kötü bir tasarım potansiyel olarak sadece kontrol etmekle düzeltilemeyeceğinden ve geliştirilmiş bir kodun gözden geçirilmesi aslında bir nevi resmi onay niteliği kazandırdığından problem yaratabilmektedir.

- **Versiyon Kontrolü (Version Control):**

Geliştirme sürecinde yazılımın birbirini takip eden versiyonlar şeklinde geliştirilmesi; eksikliklerin görülmesi, yanlış anlamaların giderilmesi, hataların düzeltilmesi ve ortak bir dilin kullanılabilmesine imkan tanınması bakımından oldukça önemlidir.

### **1.3.2. Yetenek Olgunluk Modeli Entegrasyonu (YOME-CMMI, Capability Maturity Model Integration)**

Etkin süreçlerde kullanılması zorunlu unsurları sağlayan bir süreç iyileştirme yaklaşımıdır. YOME'nin en iyi uygulamaları, model adı verilen ve her biri farklı bir ilgi alanına yönelik hususlar sunan dokümanlarda yayınlanmıştır. YOME modellerinde "Geliştirme (Development)" ve "Satın Alma (Acquisition)" olmak üzere iki ilgi alanı tanımlanmaktadır. Mevcut YOME 1.2 Versiyonunun üç farklı alternatifi bulunmaktadır. Bunlar:

- **Geliştirici YOME (CMMI-DEV, CMMI for Development);** Ağustos 2006

ayında yayınlanmış olup, ürün ve hizmet geliştirme süreçlerini ele almaktadır.

- **Tedarik YOME (CMMI-ACQ, CMMI for Acquisition);** Kasım 2007

ayında yayınlanmış olup tedarik zinciri yönetimi, satın alma (acquisition) ve dış kaynak kullanımı (outsourcing) ve Hükûmet ve endüstrideki süreçleri ele almaktadır.

- **Hizmet YOME (CMMI-SVC, CMMI for Services)** Ocak 2009 ayında

yayınlanmış olup, kuruluş içinde ve dış müşterilere hizmetlerin sağlanması konularında klavuzluk yapmaktadır.

YOME'nin en iyi uygulamaları, kuruluşların hangi modeli seçtiklerine bakmaksızın her bir kuruluşun kendi iş hedeflerine uyarlanmıştır. Kuruluşlar YOME sertifikası alamazlar. Bunun yerine kuruluşlar 1'den 5'e kadar değişen seviyelerde derece ile ödüllendirilirler. Geliştirici YOME aşağıdaki 22 süreç alanını dikkate almaktadır:

- **Rasgele Analiz ve Çözümleme**

- Konfigürasyon Yönetimi
- Karar Analizi ve Çözümleme
- Entegre Proje Yönetimi
- Ölçme ve Analiz
- Kurumsal Yenilik ve Gelişme
- Kurumsal Süreç Tanımlama
- Kurumsal Süreç Hedefi
- Kurumsal Süreç Performansı
- Kurumsal Eğitim
- Proje Takip ve Kontrolü
- Proje Planlamaları
- Süreç ve Ürün Kalite Kontrolü
- Ürün Entegrasyonu
- Sayısal (Kantitatif) Proje Yönetimi
- İhtiyaç Yönetimi
- İhtiyaç Geliştirme
- Risk Yönetimi
- Tedarikçi Anlaşama Yönetimi
- Teknik Çözümleme
- Doğrulama
- Geçerleme

Olgunluk modelleri ISO/IEC 15504 standardının bir bölümünü oluşturacak şekilde uluslararası bir standart hale getirilmiştir. ISO/IEC 15504 standardı oluşturulurken ilk olarak ömür devri standardı ISO 12207'den esinlenilmiştir. ISO/IEC 15504 bir uluslararası standartlar standardıdır. Bu alanda uluslararası standart olgunluk modeli önerileri dikkate alınarak oluşturulmuştur. Yazılım Süreç İyileştirme Yetenek Belirleme (YASİYEB, SPICE-Software Process Improvement Capability dEtermination) olarak da bilinen ISO 15504, Uluslararası Standardizasyon Kuruluşu (ISO-International Organization for Standardization) ile

Uluslararası Elektroteknik Komisyon (IEC-International Electrotechnical Commission) arasında tesis edilmiş Müşterek Teknik Alt Komite (Joint Technical Subcommittee) tarafından geliştirilmiş yazılım süreç değerlendirme şablonudur. Olgunluk modellerinde referans alınan bir standart olan ISO/IEC 15504 ile kıyaslama süreci için uygun bir model belirlenmesi amaçlanmıştır. ISO/IEC 15504, yazılım geliştirme faaliyetlerinin yönetilebilmesi, kontrolü, yol gösterimi ve izlenebilmesi için süreçleri modeller. Oluşturulan modeller, daha sonra yazılım geliştirme faaliyeti esnasında geliştirme faaliyetini yürüten kuruluş veya proje timinin ne yaptığının ölçülmesinde kullanılır. Bu bilgi de zayıf yönlerin belirlenmesi ve iyileştirmelerin yapılabilmesi bakımından analiz edilir. Bu aşamada, aynı zamanda bahse konu kuruluş veya tim açısından genel uygulamalarla bütünleştirilmesi veya uygulanmaya devam edilmesi bakımından kuvvetli taraflar da ortaya konur.

ISO/IEC 15504 bir referans modeli de sunmaktadır. Bu modelin süreç ve yetenek olmak üzere iki boyutu mevcuttur.

- Hâlihazırda müşteri-tedarikçi, mühendislik, destekleyici, yönetim ve kuruluş kategorisi olmak üzere beş süreç kategorisinden oluşan süreç boyutuna, Bilgi Teknolojileri (IT-Information Technologies) ve şirket süreç kategorilerinin de eklenmesi beklenmektedir. ISO/IEC 15504 referans modelinin süreç boyutu sayesinde bir teknoloji kuruluşunda süreç iyileştirmesi yapılabilmektedir. Süreç iyileştirme faaliyeti zor ve çoğunlukla başarısızlıkla sonuçlanmaktadır. Bu nedenle, süreç yetenek seviyesinin çok iyi anlaşılması ve iyileştirme projesinden sonra durum değerlendirmesi yapılması gerekmektedir. ISO 15504'ün getirdiği standart sayesinde kuruluşların her bir safhada kapasitelerini kullanabilme durumları değerlendirilmektedir. Bu standartla iyileştirme program ihtiyaçları belirlenmekte ve iyileştirmelerin planlanması ve icrasına ilişkin içinde 8 aşamalı bir iyileştirme programının da yer aldığı rehberlik hizmeti gerçekleştirilmektedir.

- ISO/IEC 15504'te her bir süreç için, aşağıdaki yetenek seviyeleri tanımlanmıştır.

- Seviye 0: Tamamlanmamış Süreçler
- Seviye 1: Gerçekleştirilmiş Süreçler
- Seviye 2: Yönetilen Süreçler
- Seviye 3: Başarılı Olunan Süreçler
- Seviye 4: Önceden Tahmin Süreci

▪ Seviye 5: Optimizasyon Süreci

Süreç özellikleri kullanılarak süreçlerin yetenekleri ölçülebilir. Standart içinde, aşağıdaki dokuz süreç özelliği tanımlanmaktadır.

- Süreç Performansı
- Performans Yönetimi
- İş Ürün Yönetimi
- Süreç Tanımlama
- Süreç Uygulama
- Süreç Ölçümü
- Süreç Kontrolü
- Süreç Yenileştirme
- Süreç Optimizasyonu.
- Dış kaynak kullanımı (outsourcing) ile yazılım geliştirmeyi planlayan bir kuruluşun, işi vereceği potansiyel yüklenicilerin yeteneklerini çok iyi anlayıp tahlil etmesi gereklidir. ISO/IEC 15504 yüklenici seçim kararlarına ilişkin bilgiler de sunmaktadır. İster kendileri ister başka bir kuruluş tarafından değerlendirilsin önerilen yüklenicilerin değerlendirilmesi için bir şablon sunulmaktadır.
- Kuruluşlar, kendi ihtiyaçlarını dikkate alarak yükleniciler için hedef bir yetenek belirleyebilir ve bu hedef yeteneği belirleyen hedef süreç özelliklerini kullanarak hedef yeteneğin değerlendirmesini yapabilirler. Hedef süreç özellikleri, kuruluşun (ki bu bir hükûmet kuruluşu da olabilir) ve özellikle en ucuz teklifi veren yüklenicinin seçilmesi zorunluluğu bulunduğu anda önem kazanır. Bu aynı zamanda yüklenici tarafından kullanıcının istediği seviye ile kendi mevcut yetenekleri arasındaki eksikliklerin belirlenmesine ve sözleşme ihtiyaçlarının karşılanması için kendi yeteneklerinde iyileştirme yapabilmesine imkân tanır. Yetenek belirleme kalitesinin yükseltilmesinde, riski hedef süreç profillerinin belirlenmesinde esas faktör olarak kullanan Pratik Süreç Profilleri (Practical Process Profiles) adı verilen metod kullanılabilir. Risk ve sürecin birleştirilmesi aktif risk azaltımı faaliyetinin iyileştirilmesine katkıda bulunur ve ortaya çıkması muhtemel problem sahalarını da azaltır.



## 2. YAZILIM PROJE YÖNETİMİ AŞAMA VE ADIMLARI

Yazılım proje yönetiminde ilk faaliyet planlamadır. Bu aşamada yüklenici yapacağı faaliyetleri planlamalı ve isimleri kuruluştan kuruluşa değişen proje yönetim planı, kalite planı, risk yönetim planı gibi planlar bu planlama faaliyetlerin çıktıları olmalıdır. Burada üzerinde önemle duracağımız plan konumuz gereği kalite planı olacaktır. Kalite Planında yüklenici bu projenin gerçekleştirilmesinde yerine getireceği tüm kalite faaliyetlerini belirtmek (planlamak) durumundadır.

Bir projenin başarılı olarak yönetildiğinden bahsedilebilmesi için o projenin tüm adımlarının izlenmesi, gerekli noktalarda kilometre taşlarının oluşturulması ve bu noktalardan geriye dönüşlerin düzenlenmesi gerekmektedir. Diğer bir deyişle öncelikle bir iş tanımlanmalı, planlanmalı, zamanlaması ve görev paylaşımı yapılmalı, daha sonra bu iş gerçekleştirilmeli, performansı ölçülmeli ve buna ilişkin kilometre taşı oluşturulmalıdır (Gerekli birçok noktada gözden geçirme faaliyeti gerçekleştirilmeli ve bu faaliyetin sonunda bazı dokümanların veya ürünün konfigürasyonu dondurulmalıdır). Bir aşamada gözden geçirme faaliyetinin amacı aşamanın gözden geçirme faaliyetine kadar gerçekleştirilen bölümlerinin, planlandığı, beklendiği gibi olup olmadığının araştırılmasıdır. Bu noktada eğer faaliyetler ve bunların sonuçları planlandığı (beklendiği) gibi ise süreçte ilerlemeye devam edilecek, aksi takdirde geriye dönüşler yapılarak gerekli düzeltmeler yapılacaktır. O zaman gözden geçirme faaliyeti aşamanın performansını değerlendirmek açısından oldukça önemlidir. Ayrıca bu gözden geçirme faaliyetinin sıklığı geriye dönüşlerin miktarını ve süresini belirleyeceği açıktır. Fakat bu faaliyetin amacına uygun olabilmesi için optimal sıklık belirlenmelidir, genel yaklaşım her önemli faaliyetin veya sürecin ana aşamalarının sonunda bir gözden geçirme faaliyeti gerçekleştirmek yönündedir. Yazılım geliştirme sürecinin ilk aşaması sistem ihtiyaçlarının belirlenmesi ve analiz edilme aşamasıdır.

### 2.1. Sistem İhtiyaçlarının Belirlenmesi Aşaması (ABD Askeri Standardı 498 1994: DI-IPC-81431)

Bu aşamada yazılım ve donanım bir bütün olarak ele alınmakta, yazılım ve donanım için ihtiyaçlar belirlenmektedir. Bu aşamanın sonunda sistem ihtiyaçları dokümanı oluşturulacak ve yapılan gözden geçirme sonunda bu dokümanın konfigürasyonu dondurulacaktır. Sistem ihtiyaçları dokümanının hazırlanması esnasında tedarikçi ile yoğun bir iletişim içerisinde olunmalı, özellikle sözleşmede belirtilmeyen veya belirtilemeyen ihtiyaçların oluşturulmasında, sözleşmede yer alan

ihtiyaçların detaylandırılmasında, oluşturulan tüm ihtiyaçların mantık, doğruluk ve yeterlilik bakımından araştırılmasında bu iletişim artırılmalıdır. Bu aşamanın sonunda oluşturulması gereken dokümanda (sistem ihtiyaçları dokümanında) bulunması gerekenler ve bu süreçte dikkate alınması gereken konular aşağıda verilmiştir.

### **2.1.1. Bilgisayar (Hedef Donanım) Kaynak İhtiyaçları**

Bu bölümde yazılımın kullanacağı yani üzerinde çalışacağı donanımın özellikleri verilmelidir. Bunlar ekipmanların tipi, sayısı, hafıza kapasitesi, işlemcilerin hızı, eepromların çeşidi, girdi çıktı cihazları, ağ ekipmanları ve diğer donanım ekipmanları olabilir.

Yazılımın hedef donanım üzerinde kullanacağı hafıza kapasitesi, girdi çıktı cihaz kapasitesi, ağ ekipmanı kapasitesi, dış hafıza kapasitesi verilmelidir. Bu kapasite değerleri verilirken bu kapasite değerlerine ulaşma şartları da belirtilmelidir.

Bu ihtiyaçlar oluşturulurken bir yazılım sistemine ne kadar dışardan girdi ve sistemden dışarıya ne kadar çıktı veriliyorsa o yazılımın tepki süresinin o kadar düşeceği bilinmelidir.

### **2.1.2. Bilgisayar Yazılım İhtiyaçları**

Bu bölümde yazılımın üzerinde çalışacağı veya çalışırken ihtiyaç duyacağı yazılımlar verilmelidir. Bu tip yazılımlar işletim sistemleri, veri tabanı yönetim sistemleri, iletişim yazılımları olabilir.

Eğer geliştirilecek yazılımın hedef donanımı sözleşmede veya tedarikçi ihtiyaçlarında tanımlandı ise, sistem ihtiyaçlarının oluşturulması sırasında hedef donanımın üzerinde gerekli analiz çalışmaları yapılmalıdır. Eğer hedef donanım üzerinde başka yazılımlar da çalışacak veya halen çalışıyor ise, geliştirilecek yazılımın kullanacağı maksimum donanım kapasitesi üzerinde önemle durulmalıdır. Bu kapasite kullanımının zaman göre değişebileceği yapılan analizler esnasında göz ardı edilmemelidir.

Eğer sistemin birden fazla modda çalışabilme özelliğinin olması ihtiyaçlar arasında ise her bir modun özellikleri verilmelidir. Bu modlara örnek olarak güvenli mod, savaş modu, barış modu gibi modlar verilebilir.

### **2.1.3. Sistem Ara Yüz İhtiyaçları**

Sistemin dışarı ile ara yüz ihtiyaçları belirlenmeli ve tanımlanmalıdır. Örnek olarak bir bilgisayarın hangi girişinden (port) hangi gerçek zaman bilgisinin diğer sistemlere transfer edileceği, bu verinin maksimum hızı v.s. burada verilmelidir.

Ayrıca sistemin kendi içinde ihtiyaç duyacağı ara yüzler de burada verilmelidir (Sistemi oluşturan alt sistemler arasındaki ara yüz ihtiyaçları gibi). Örnek olarak bilgisayar sistemde disk okuyucunun maksimum veri okuma hızı ve veri hatlarına (data-bus) verebileceği veri hızı burada verilebilir.

### **2.1.4. Sistem Kalite Faktörleri**

Kalite faktörleri ile ilgili ihtiyaçlar bu dokümanda verilmelidir. Bu ihtiyaçların belirlenmesinde karşılaşılabilecek en büyük zorluk bu ihtiyaçların birçoğunun tedarikçi tarafından büyük olasılıkla belirlenmemiş olmasıdır.

### **2.1.5. Sistem Tasarımı ile İlgili Sınırlamalar**

Sistem tasarımı ile ilgili sınırlamalar burada verilmelidir. Bu sınırlamalara örnek olarak sistemde tedarikçinin istediği donanımın kullanılması zorunluluğu, sistemin belirli bir büyüklükte olması (donanım veya yazılım) gibi sınırlamalar verilebilir.

### **2.1.6. İhtiyaçların İzlenebilmesi**

Oluşturulan tüm ihtiyaçların izlenebilmesi ihtiyacından yola çıkıldığında, yazılı hael getirilen ihtiyaçların tüm geliştirme süreci boyunca izlenebilir olması da önem kazanmıştır. Bunun için sistem ihtiyaçları gruplandırılabilir (her bir alt sistem için bir grup) ve numaralandırılabilir.

### **2.1.7. Test ve Değerleme İhtiyaçları**

Yapılacak kalite değerlendirme testleri ile ilgili ihtiyaçlar bu sisteme ilişkin hangi kalite değerlendirme testlerinin gerçekleştirileceği burada verilmelidir. Bu durum bir uygunluk matrisi yapılarak verilebilir. Bu uygunluk matrisinde sistemin hangi bölümlerine hangi kalite değerlendirme testlerinin uygulanacağı veya kalite değerlendirme faaliyetleri olarak test mi, gösteri mi veya analiz mi yapılacağı belirlenmelidir.

### **2.1.8. Personel ve Eğitim İhtiyaçları**

Bu sistemi kullanmak için gerekli olan personel, bu personelin yetenekleri, eğitim ihtiyaçları ortaya konmalı, sistemi kullanan personelin olası hataları ve bu hatalara karşı alınabilecek önlemler belirlenmelidir.

### **2.1.9. Lojistik İhtiyaçlar**

Lojistik ihtiyaçlar, sistemin idamesi ve güvenilirliği ile ilgili ihtiyaçlardır. ihtiyaçların bu aşamada oluşturulması, sistem ihtiyaçları gözden geçirme toplantısında analiz edilerek nihai hale getirilmesi ve oluşturulan dokümanların dondurulması ile bu aşamada gerçekleştirilmesi gereken faaliyetler sonuçlandırılmalıdır. Süreçte bir sonraki aşamamız ise sistem tasarımı aşamasıdır.

## **2.2. Sistem Tasarımı Aşaması**

Bu aşamada öncelikli hedef, sistem ihtiyaçları dokümanında belirtilen tüm işlevsel, işlevsel olmayan ve türetilen ihtiyaçları karşılayan bir sistem tasarım dokümanı oluşturmak olacaktır.

Bu tasarım dokümanının hazırlanmasında üzerinde önemle durulacak konular arasında;

- Sistemin kabul edeceği girdiler üreteceği çıktılar, diğer sistemlerle etkileşim esasları, yapısal oluşum (configuration) öğeleri,
- Sistemin girdilere karşı vereceği tepkiler, girdilere yapacağı işlemler, tepki zamanları, istenmeyen bir girdi geldiğinde verilecek tepki,
- Veri tabanı yapısı, veri tabanının kullanımı, veri tabanından bilgi alınıp verilmesi,
- Güvenilirlik, güvenlik, idame ettirilebilirlik v.s gibi kalite faktörlerinin karşılanabilmesi için yaklaşım,
- Kullanılacak donanım (bilgisayar veya ağ cihazları),
- Kullanılacak donanımın fiziksel büyüklüğü, rengi, ağırlığı gibi özellikleri,
- Donanımın rasgele erişimli bellek, sabit disk kapasiteleri, işlemci hızı,
- Girdi çıktı kapasiteleri, donanımın yenilenebilme ihtimali, donanımın kaynak yazılımla çalışabilmesi,

- Donanım tedarik planı ile proje planının uyuşması,
- Tüm yazılım ve donanım konfigürasyon öğelerinin belirlenmesi sayılabilir.

Ayrıca sistem öğelerinin;

- Her birinin girdilere karşı vereceği tepki, girdilere yapacağı işlemler, tepki zamanları, istenmeyen bir girdi geldiğinde davranışı,
- Her birinin ayrı bir öge numarası olması,
- Her birinin hangi sistem gereğini karşılayacağı,
- Hangilerinin geliştirileceği ve hangilerinin mevcut olduğu ve hazır olarak sisteme dahil edileceği belirlenmelidir.

Donanım öğeleri ile ilgili olarak; üreticileri, isim ve modelleri, fiziksel ve fonksiyonel özellikleri de verilmelidir.

### **2.2.1. Ara Yüz Tasarımı**

Yapısal oluşum öğelerinin kendi aralarındaki ve sistemin diğer sistemlerle ilgili ara yüzü verilmelidir. Tüm öğeler arasındaki bağıntı ve ilişkiler bilgi akış diyagramları gibi diyagramlarla verilebilir. Bu aşamada tüm ara yüzlerin net olarak tanımlanmamasının zorluğu göz önüne alındığında, ara yüzlerin daha sonraki aşamada hazırlanabilecek ara yüz tasarım dokümanında net olarak verilebileceği unutulmamalıdır.

### **2.2.2. Sistem İhtiyaçlarının İzlenmesi**

Sistem ihtiyaçlarının belirlenmesi aşamasında tüm sistem ihtiyaçları oluşturulmuş yazılı hale getirilmiş ve numaralandırılmıştır. Bu ihtiyaçların tümünün sistem tasarımı aşamasında göz önünde bulundurulduğu ve sistem tasarımının bu ihtiyaçları karşılayacak şekilde gerçekleştirildiğinden emin olunması için gerekli analiz ve araştırmaların, çalışmaların, bu aşamada yapılması gerekmektedir. Hangi ihtiyaçların hangi konfigürasyon öğesi veya öğeleri tarafından karşılandığının hazırlanacak bir uygunluk matrisi ile takip edilmesi, başvurulacak yöntemlerden biri olmakla beraber, oluşturulan sistem ihtiyaçları dokümanı da ihtiyaçların izlenebilmesi için uygun olmalıdır.

### 2.2.3. Sistem Tasarımının Gözden Geçirilmesi

Sistem tasarım dokümanının konfigürasyonunun dondurulması, bu aşamada gerçekleştirilen faaliyetlerin gözden geçirilmesi ve projenin ilerlemesinde bir kilometre taşı oluşturulması için sistem tasarım gözden geçirmesi faaliyeti gerçekleştirilmelidir.

Bu faaliyette;

- Her bir gereğin yazılım ve/veya donanım öğeleri tarafından karşılandığının araştırılması ve tersinir olarak her bir öğenin bir grup öğeyle bağıntısının kurulmuş olması,
- Karşılınmayan hiçbir gereğin kalmadığından emin olunması,
- Eğer ihtiyaçlara değişiklik önerileri geldiyse bu önerilerin gerekli onay mekanizmalarından geçerek kabul edildiği ve sistem tasarım dokümanının buna göre oluşturulup oluşturulmadığı,
- Sistem tasarım dokümanında anlaşılmayan noktaların kalmamış olması,
- Tüm konfigürasyon öğeleri arasında ara yüzlerin tanımlanması,
- Sistem seviyesinde hangi kalite değerlendirme testlerinin gerçekleştirileceği ve bu testler için gerekli yazılım ve donanımın tanımlanmış olması,
- Sistemde yer alacak olan donanımların belirlendiğinden ve bu donanımın geliştirilecek yazılımla çalışabilir olması,
- Bu gözden geçirme faaliyetinde bu aşamada gerçekleştirilen faaliyetlerin ve oluşturulacak dokümanın proje standartlarına, planlara uygun olarak gerçekleştirilip gerçekleştirilmediğinin araştırılması üzerinde önemle durulması gereken konulardır.

### 2.2.4. Ana Odak (Domain) Analizi

Bir yazılım tasarımında ilk adım, ister mevcut bir yazılıma ilave/yeni bir uygulama/yeni bir alt sistem, ister bir bütün olarak yeni bir sistem olsun Ana Odak Analizi yapılarak nereden başlanacağına tespit edilmesidir. Yazılım geliştiren kişilerin yeni yazılım konusunda yeterli bilgiye sahip olmadıkları dikkate alındığında, ilk iş olarak gerçekleştirilecek yazılımın ana odağı araştırılmalıdır. Bu analizin diğer

bir hedefi; daha sonra saha uzmanlarından ihtiyaları temin edecek ve toparlamaya alıřacak analizcinin bu uzmanlar tarafından ne sylendiđini daha iyi anlamasını kolaylařtıran ana odak noktası (domain) terminolojisinde konuřmasını sađlamaktır. Analizci uygun alıřma yapmadıđı takdirde “Sylediđimi sandıđınızı anladıđınıza inandıđınızı biliyorum, ancak sylemek istediđimin duyduđunuzdan anladıđınızdan farklı olup olmadıđından emin deđilim” karmařası ortaya ıkar (Frakes ve Kang 2005:529-536).

1980’lerin bařında James Neighbors tarafından ortaya atılmıř olan Ana Odak Analizi (rn Hattı Analizi), yazılım mhendisliđinde, aslında bir ana odaktaki (domain) farklı ve ortak yanların ortaya konmasında yazılım sistemlerine iliřkin analizlerin yapılma srecidir. Ana Odak Analizi, Ana Odak Mhendisliđinin ilk safhası olup sistematik řekilde yazılımın yeniden kullanımına iliřkin esasların anlařılmasında anahtar bir metottur.

Ana Odak Analizi iin nerilen metotlardan her biri zellik tablosu, ara yz tablosu, ara yz řablonları ve jenerik mimariler gibi saha modelleri sunar. Jenerik bir mimari, bir ana odak noktasındaki btn sistemleri tanımlar (Frakes ve Kang 2005:529-536) ve ana odak analizi iin nerilen metotlara iliřkin bir deđerlendirme sunar.

Ana Odak Analizi rnleri, bazen Birleřtirilmiř Modelleme Dili (UML-Unified Modeling Language) kullanılan nesne tabanlı modeller (object-oriented models) ve/veya Varoluř-İliřki Diyagramları (ERD-Entity-Relationship Diagrams) kullanılan veri modelleri řeklinde karřımıza ıkar. Yazılım geliřtiriciler bu modelleri yazılım mimarisi ve uygulamalarının kullanılmasında temel alırlar.

### **2.3. Yazılım İhtiyalarının Belirlenmesi ve Analizi**

Bir yazılım oluřturmadaki en kritik adım kullanıcı ihtiyalarının ortaya konmasıdır. Kullanıcılar son rn olarak ne istediklerini bildikleri halde yazılımın ne yapması gerektiđini bilmezler. Bu noktada, tamamlanmamıř, tecrbeli ve bilgili yazılım mhendisleri tarafından birden ok anlama gelebilen/yoruma aık veya tamamen zıt ihtiyalar dahi anlařılmıř olabilir. Yazılan kodların sık sık kullanıcıya grř almak iin sunulması yanlıř ihtiya tanımlamaları riskini azaltacaktır.

Yazılım ihtiyaları analizi, muhtemelen birbirleriyle atıřan farklı seviyedeki kullanıcı ihtiyaları da dikkate alınarak, yeni yazılacak veya zerinde deđerliklik yapılacak rn ihtiyalarının veya zel durumların belirlenmesi faaliyetlerini ierir.

Sistematik karakterdeki bu ihtiyaç analizi aynı zamanda ihtiyaç toplama, ihtiyaç belirleme, ihtiyaç mühendisliği ve ihtiyaç özellikleri belirleme olarak da tanımlanmaktadır (Young 2001). İhtiyaç analizi, geliştirilecek projenin başarısında kritik önem arz etmektedir (Wiegiers 2003).

İhtiyaçlar; karşılanabilir, ölçülebilir ve test edilebilir olması yanında kullanılacağı iş yeterliğini sağlayan ve sistem tasarımı için her seviyede tanımlı yapıda olmalıdır.

Mühendislikte, ihtiyaç denildiğinde belirli bir ürün veya hizmetin ne olması veya nasıl yapılması gerektiğini belirten yazılı hale getirilmiş gereklilikler anlaşılır. Bu analizin, genellikle sistem ve yazılım mühendisliklerinde resmen kullanılan, kullanıcıya kolaylık sağlayan, kullanımda ihtiyaç duyulan niteliklerin/yeteneklerin/özelliklerin/sistemin kalitesini ortaya koyan tanımlamalar bütünü olduğu söylenebilir (Young 2001).

Klasik mühendislik yaklaşımında, ihtiyaçlar seti ürün geliştirme tasarım aşamasında girdi olarak kullanılmaktadır. İhtiyaçlar, belirli bir proje için hangi öge ve işlevlerin gerektiğini gösterir.

İhtiyaç geliştirme safhası öncesi uygulanabilirlik (feasibility) çalışması veya projenin konsept analiz safhası icra edilebilir. İhtiyaçlar safhası;

- İhtiyaçların toparlanması (her seviyedeki kullanıcılardan ihtiyaç duydukları hususların öğrenilmesi ve raporlanması),
- Analizi (tutarlılıklarının ve bütünlüklerinin kontrol edilmesi),
- Özelliklerinin Tanımlanması (İhtiyaçların yazılı hale getirilmesi, specification)
- Doğrulanması (belirlenen ihtiyaçların doğruluğunun denetlenmesi, verification) faaliyetlerinden oluşur (Wiegiers 2003).

### **2.3.1. Ürün veya Süreç İhtiyaçları**

Projelerde üç çeşit ihtiyaç bulunur.

- İş İhtiyaçları (Business Requirements), istenilen değer elde edilmesi veya başarılması için yapılması gerekenleri iş terminolojisi dâhilinde tanımlar.
- Ürün İhtiyaçları (Product Requirements), iş ihtiyaçlarının karşılanabilmesi için mümkün yollardan biri olan sistemi veya ürünü tanımlar.



- Süreç İhtiyaçları (Process Requirements), geliştirilen yazılımın maruz kalacağı süreçleri ve tabi olacağı kısıtları tanımlar.

Ürün ve Süreç İhtiyaçları birbirleriyle yakın ilişkilidir, ancak süreç ihtiyaçları genellikle daha yüksek seviye ürün ihtiyaçlarını tanımlamakta kullanılır. Örneğin; bir süreç ihtiyacı olan maksimum geliştirme maliyet ihtiyaçları bir ürün ihtiyacı olan maksimum satış ücreti belirleme ihtiyacının karşılanmasına yardım edebilir. Ürün için belirlenen bir ihtiyaç (Ürün İhtiyacı); nesne tabanlı programlama (object-oriented programming) benzeri özel geliştirme yöntemlerinin, yöntem kullanım esaslarının veya gözden geçirme/denetleme sürecinin uygulanmasını öngören ihtiyaçlar(Süreç İhtiyaçları) tarafından ayrıntıları ortaya konularak değerlendirilir.

### 2.3.2. Sistem ve Yazılım Mühendisliklerinde İhtiyaçlar

Sistem mühendisliğinde, bir ihtiyaç sistemin ne yapması gerektiğinin tanımlanmasını esas alır ve İşlevsel İhtiyaçlar (Functional Requirement) olarak ifade edilir. Bu tip ihtiyaçlar, hazırlanan sistemin imkân ve kabiliyetlerinin belirlenmesini öngörür. Başka bir ihtiyaç tipi ise; sistemin kendisi hakkındaki esasları belirler ve işlevlerini ne kadar iyi gerçekleştirebileceğini ortaya koyar. Bu ihtiyaçlar ise genellikle “İşlevsel Olmayan İhtiyaçlar (Non-functional requirements)“, “Performans İhtiyaçları“ veya “Hizmet İhtiyaçları Kalitesi“ isimleriyle anılır. Bu ihtiyaçlara; mevcudiyet (availability), test edilebilirlik (testability), idame edilebilirlik (maintainability), ve kullanım kolaylığı (ease-of-use) örnek olarak verilebilir.

“İyi“ olarak tanımlanan bir ihtiyaç listesi, uygulama esası (implementation bias) veya çözümlenme (solutioneering) olarak da bilinen ihtiyaçların nasıl karşılanması gerektiği konusunda kararı kendisi vermemeli ve bunu sistem tasarımcısına bırakmalıdır.

Yazılım mühendisliğinde, sistem mühendisliğinde tanımlanan ihtiyaçlara ilişkin esaslar aynen uygulanır. Ancak, burada ilgi odağı yazılımın kendisidir.

### 2.3.3. İhtiyaç Geliştirme Sürecine Etki Eden Faktörler

- **Sınıflandırma:** İhtiyaçlar, yukarıda açıklandığı şekilde işlevsel ve işlevsel olmayan ihtiyaçlar olmak üzere iki kategoride sınıflandırılmaktadır. Yazılımın ilerleyen safhalarında ya “performans ihtiyaçları“, “bakım ihtiyaçları“, “emniyet ihtiyaçları“, “güvenilirlik ihtiyaçları“ ya da diğer birçok ihtiyaç tipinden birine göre sınıflandırılabilir.

- **İyi Tanımlanmış İhtiyaçlar:** İyi tanımlanmış ihtiyaçlar, birçok yazar tarafından farklı olarak ifade edilmekle birlikte, her bir yazar genellikle, özellikleri kendi genel bilgilerine uygun şekilde veya bahsedilen hedef teknoloji odağına göre vurgulamaktadır. Ancak, aşağıdaki nitelikler genel olarak hepsi tarafından ortak olarak benimsenmektedir (Boehm 1988:1462-1477, Bridges 1995, Brinkkemper 1996:275-280, Davis 1993).
- **Tekillik (Cohesion):** Her bir ihtiyaç sadece ama sadece tek bir konuya yönelik olmalıdır.
- **Bütünlük (Completeness):** Her bir ihtiyaç her hangi bir eksik bilgi bulundurmadan tamamıyla bir yerde açıklanmış olmalıdır.
- **Uyumluluk (Consistency):** Her hangi bir ihtiyaç diğeriyle çelişmemeli ve bağlayıcı nitelikteki her türlü mevzuata uygun olmalıdır.
- **Doğruluk (Correctness):** Her bir ihtiyaç, yetkili kullanıcılar tarafından açıkça ortaya konmuş bir iş ihtiyacının tamamını veya bir kısmını karşılar nitelikte olmalıdır.
- **Zamansal Geçerlilik (Current):** Her bir ihtiyaç zaman geçtikçe kullanılmaz öğeler içermemelidir.
- **Dıştan Gözlemlenebilirlik (External Observability):** Her bir ihtiyaç, kullanıcı tarafından tecrübe edilen veya dıştan gözlemlenebilen nitelikleri belirtmelidir. İhtiyaçlar; iç mimariyi, tasarımı, uygulamayı veya test etme kararlarını açıklayan kısıtlar olup "ihtiyaçlar dokümanı"nın "Kısıtlar" bölümünde açık şekilde ifade edilmelidirler.
- **Uygulanabilirlik (Feasibility):** Her bir ihtiyaç projenin kısıtları dâhilinde uygulanabilir olmalıdır.
- **Anlam Karmaşıklığı (Unambiguity):** Her bir ihtiyaç teknik jargon, kısaltma (acronym) (İhtiyaçlar dokümanında tanımlanmadıysa) veya belirli bir gruba hitap eden diğer laf kalabalıklarından (esoteric verbiage) uzak olarak özlü ve herkesin anlayabileceği açıklıkta olmalıdır. Kişiyeye özgü görüşleri değil, objektif faaliyetleri ifade etmelidir. Anlaşılması zor başlıklar, sıfatlar, zarflar, fiiller ve konusal kalıplardan kaçınılmalıdır . Olumsuz cümlecikler ve bileşik cümleler ise hiçbir şekilde kullanılmamalıdır.

- **Zorunlu Hususlar (Mandatory Issues):** Her bir ihtiyaç kullanıcısı tarafından tanımlanmış ve bulunmadığı takdirde düzeltilemez hatalara neden olabilecek nitelikleri temsil etmelidir.

- **Doğrulanabilirlik (Verifiability):** İhtiyacın uygulaması “muayene”, “analiz”, “gösterim” veya “test” olmak üzere dört metottan biriyle belirlenmiş olmalıdır.

- **Test Edilebilirlik (Testability):** İhtiyaçlar test edilebilir olmalıdır. Bu mümkün olmadığında, analiz, muayene veya tasarım gözden geçirme gibi başka bir doğrulama metodu uygulanmalıdır. Test edilebilir ihtiyaçlar geçerlemenin önemli birer bileşenidir. Kesinleşmiş ihtiyaçlar, yapılarından dolayı test edilebilir nitelikte değildir. Çünkü bunlar, sistemin belirli bir yeteneği veya işlevi ya yapabileceğini ya da yapamayacağını kesin olarak belirleyen ihtiyaçlardır. Bu ihtiyaçların doğru şekilde testi sonsuz bir test döngüsü gerektirir. Bu nedenle bu gibi ihtiyaçlar, genellikle daha kullanışlı bir zaman periyodunun tanımlanabilmesi amacıyla yeniden yazılmalıdır. Test edilemeyen ve işlevsel olmayan ihtiyaçlar kullanıcının bakış açısının ve niyetinin anlaşılabilmesi için yazılı olarak muhafaza edilebilir. Ancak, bunlar genellikle karşılanabilmelerini sağlayacak kullanışlı bir yöntem belirlenmesi amacıyla süreç ihtiyaçlarına kaydırılır. Örneğin; işlevsel olmayan bir ihtiyaç arkakapı yöntemlerinden uzak tutularak çift yönlü programlama mantığının (pair programming) kullanılması amacıyla bir süreç ihtiyacı ile yer değiştirilerek karşılanabilir. Zorunlu bir özellik olan test edilebilirlik, sisteme ilişkin verilerin geçerlenmesi bakımından açıklık sağlasa da diğer önemli konulardan dikkatleri başka yönlelere de kaydırabilir. Test edilebilirlik genellikle hatalı tanımlanmış ihtiyaçları belirleyemediğinden, hatalı olan bir ihtiyaç tanımı halen test edilebilir nitelikte olabilir. Basit analiz, muayene veya gözden geçirme tek başına bunlardan bazılarını açığa çıkarabilse de istenilen seviyede başarı sağlayamaz. Bu nedenle ihtiyaçların ve/veya tasarımın yeterliliğine ilişkin test ve değerlendirilme sağlayan yöntemler geliştirilmiştir.

#### 2.3.4. İhtiyaçların Analizi

İhtiyaçlar, yoruma açıklık, tamam olmama veya tutarsızlık konularında hassastır. Bu konulardaki sorunları gidermeye yönelik olarak “Sıkı Muayene (Rigorous Inspection)” gibi teknikler geliştirilmiştir. İhtiyaç tanımlama aşamasında yoruma açıklık, tamam olmama veya tutarsızlık bakımından hatalı tanımlanan ihtiyaçlarına, ürün geliştirme faaliyetinin ileriki aşamalarında benzer problemlerle

karşılaştığında getirilecek çözüm maliyetlerinden daha az maliyetle çözüm getirilebilir. İhtiyaç analizi aslında bu problemlerin önceden giderilmesine çalışır.

Üretimi uzun süren, mevcut uygulama seçeneklerini sınırlamaya başlayan ve yüksek maliyetli olan çok kapalı veya çok detaylı iki ihtiyaç arasında dikkate alınması gereken bir mühendislik değerlendirmesi (trade-off) mevcuttur.

İhtiyaçlar genellikle yazılı hale getirilerek farklı kullanıcılar arasında bir iletişim aracı gibi kullanılır. Bu nedenle tanımlanan ihtiyaçların hem normal kullanıcılar ve hem de yazılım geliştiricilerin kolayca anlayabileceği seviyede yazılmalıdır. Bir ihtiyacı belirlerken sistemin ne yapmasının istendiğinin tam olarak tanımlanması çoğunlukla kullanılan bir yoldur (Örneğin: “Yüklenici ürünü GG/AA/YY tarihinden önce teslim etmelidir”). Diğer yollar ise kullanıcı tanımlamaları (user stories) ve durum çalışmalarıdır (use cases).

İhtiyaçlar genellikle zamanla değişmeye başlar. İhtiyaçların tanımlanıp onaylanmasını müteakip, değişim kontrolüne (change control) tabi tutulması gereklidir. Çoğu projede, ihtiyaçlar daha sistem tamamlanmadan değiştirilir. Bu kısmi olarak bilgisayar yazılımlarının karmaşıklığına ve kullanıcıların birşeyler görmeden önce neyi istediğini bilmemesinden kaynaklanır.

İhtiyaç analiz konseptinde aşağıdaki üç çeşit faaliyet tanımlanır.

- **İhtiyaçların Temini (Eliciting):**

İhtiyaç mühendisliğinde ihtiyaç temini kullanıcı ve müşterilerden sistem ihtiyaçlarının elde edilme faaliyetidir (Sommerville ve Sawyer 1997). Kullanıcı ve müşterilerle ihtiyaçlarının neler olduğunu anlamak için iletişim kurma faaliyetlerinin tamamı bu kapsamda gerçekleştirilir.

Yeni sistemler, insanlar arasındaki ilişkileri ve çevreyi değiştirdiğinden, bütün kullanıcıların yeni sistemin getireceği karmaşıklığı anladıklarından emin olunması ve bütün ihtiyaçları hesaba katıp katmadıklarının belirlenmesi çok önemlidir. Analizciler, kullanıcılardan ihtiyaçların öğrenilebilmesi için ikili görüşme, anket uygulama, kullanıcı gözlemleri, birlikte çalışmalar, beyin fırtınaları, durum çalışmaları, rol yapma ve prototip hazırlama gibi bir çok teknik uygularlar. Gerekliğinde bu metotları birleştirerek iş ihtiyaçlarını karşılayacak bir sistem uygulayabilir. Bu tekniklere ilişkin kısa bilgiler aşağıda sunulmuştur.

- **İkili Görüşmeler:**

İhtiyaç analizinde kullanılan ortak metotlardan biridir. Bu görüşmeler, proje kapsamında daha önce düşünülmemiş ihtiyaçları gösterebilir ve belirlenecek ihtiyaçlar tamamen zıt hususlar içerebilir. Ancak, her bir kullanıcı kendi şahsi ihtiyaçlarını veya beklentilerine ilişkin fikirlerini tanımlayabilir.

- **Müşterek İhtiyaç Geliştirme Seansları (Birlikte Çalışmalar)**

İhtiyaçlar genellikle, ikili görüşmeler esnasında eksik tanımlanmış veya hiç tanımlanmamış ve her kullanıcı tarafından bilinmeyen işlevsel karmaşıklıkta olurlar. Bu karmaşıklık müşterek kontrol altında olan bir çevrede eğitilmiş personel tarafından gerçekleştirilen ve kullanıcıların ihtiyaç temini için içinde aktif olarak yer aldıkları tartışmaların ve detaylı analizlerin yapıldığı ihtiyaç geliştirme seansları icra edilerek giderilebilir. Konusuna hakim bir yazman veya İş Analiz Uzmanı yapılacak değerlendirmelerin kaydedilmesi için bu seanslarda hazır bulunmalıdır.

- **Kontrat Şeklinde İhtiyaç Listesi Hazırlama**

İhtiyaçların belgelenmesi veya yazılı hale getirilmesinde geleneksel bir yöntem de kontrat şeklinde ihtiyaç listesi hazırlanmasıdır. Karmaşık bir sistemde bu ihtiyaçlar listesi yüzlerce sayfayı bulabilir.

- **Ölçülebilir Hedefler**

En iyi denemelerde birleştirilmiş ihtiyaçlar listesini sadece bir ipucu gibi alınır ve gerçek iş amaçları belirleninceye kadar defalarca "Niçin?" sorusu sorulur. Kullanıcılar ve yazılımı geliştirenler bu sayede o ana kadar her bir hedefin gerçekleştirilme seviyesini ölçen testler geliştirebilirler. Bu hedefler belirlenmiş ancak ölçülmemiş uzun ihtiyaç listesinden daha yavaş değişim gösterir. Küçük bir kritik ölçülmüş hedefler setinin oluşturulmasını müteakip, projenin daha yarısına bile gelmeden oldukça önce gerçek kullanıcı isteklerini karşılayan prototip hazırlama ve kısa yenilemeli geliştirme aşamaları uygulanır.

- **Prototip Hazırlama**

1980'lerin ortalarında prototip hazırlama ihtiyaç analiz probleminin çözümü gibi algılandı. Prototipler yazılım için birer taklittir. Taklitler, henüz tamamlanmamış uygulamanın nasıl çalıştığının kullanıcılar tarafından görülmesini sağlar. Sistemin nasıl görüneceğine ilişkin kullanıcıya fikir verir ve

sistemin tamamlanmasını beklemeden kullanıcıların tasarıma ilişkin kararlarını vermelerini kolaylaştırır. Prototiplerin kullanımı sayesinde kullanıcı ile geliştirici arasında iletişimde büyük iyileşme sağlanır. Uygulamaya ilişkin erken değerlendirmeler ilerleyen aşamalarda daha az değişiklik yapılmasını ve toplam maliyetlerin önemli oranda azaltılmasını sağlar.

Ancak, faydalı bir teknik olduğu ispatlanmasına rağmen prototip hazırlanması ihtiyaç analizinde karşılaşılan problemlerin tamamını aşmada aşağıdaki nedenlerle yetersiz kalmaktadır.

- Yöneticiler, prototipi gördüğünde aslında uzun sürecek proje tamamlama sürecinin neden uzun olduğunu anlamada zorlanırlar.

- Tasarımcılar, gerçek sistemi hazırlarken yeniden başlayarak zaman kaybetmekten çekindikleri için parça hazırlanmış prototip kodlarını birleştirme zorunluluğu duyarlar.

- Prototipler, prensip olarak tasarım kararlarının alınmasına ve kullanıcı arayüzü tasarımının hazırlanmasına yardımcı olmasına rağmen, ihtiyaçların gerçekte ne olduğunu gösteremezler.

- Tasarımcılar ve uç kullanıcılar kullanıcı arayüzü üzerinde iş sürecine hizmet edecek bir sistem üretimine oranla daha çok odaklanabilir

#### ▪ **Durum Çalışmaları (Use Cases)**

Durum çalışması, yeni sistem veya yazılım değişikliği potansiyel ihtiyaçlarının yazılı hale getirilmesinde kullanılan bir tekniktir. Her bir durum çalışması içinde, belirli bir iş hedefinin gerçekleştirilmesinde uç kullanıcı veya diğer sistemlerle nasıl etkileşimde bulunulması gerektiğini gösterecek bir veya daha fazla senaryo yer alır. Durum çalışmalarında ana odakta (domain) yer alan uzmanın veya uç kullanıcının kullandığı dil tercih edilerek teknik dil (jargon) kullanımından kaçınılır. Durum çalışmaları genellikle kullanıcılarla ve ihtiyaç mühendisleriyle birlikte yazılır.

Durum çalışmaları, yazılım ve sistemlerin hareket tarzlarını tanımlayan yanılmaya müsait basit araçlardır. Bir durum çalışması, hedef kullanıcının yazılım veya sistemi kullandığı tüm yolların yazılı tanımlarını içerir. Durum çalışmaları, sistemin yapısal çalışma sistemlerine yönelik tanımlama veya sistemin nasıl uygulanacağına ilişkin her hangi bir açıklama içermez. Basit şekilde

bir görevi başarmak için kullanıcının izleyeceği adımları gösterir. Kullanıcının sistemle etkileşimde bulunduğu tüm yollar bu şekilde tanımlanabilir.

1990'larda, durum çalışmaları işlevsel ihtiyaçların belirlenmesinde en yaygın yöntem olmuştur. Bu durum, özellikle onları oluşturan nesne tabanlı (object-oriented) ortamda söz konusu olmuştur. Ancak, durum çalışmaları nesne tabanlı bir tabiata sahip olmadığından uygulanabilirlikleri sistemlerle sınırlı değildir.

Her bir durum çalışması basit tek bir hedef veya görevin nasıl başarılacağını tanımlamaya odaklanır. Yazılım mühendisliği geleneksel bakış açısından bir durum çalışması, sistemin tek bir özelliğini tanımlar. BU çoğu yazılım projesi için, yeni sistem özelliklerinin tamamıyla tanımlanabilmesinde onlarca belki de yüzlerce durum çalışmasına ihtiyaç duyulacağı anlamına gelmektedir. Belirli bir yazılım projesine ait resmiyet derecesi ve projenin uygulamaya konması her bir durum çalışmasında ihtiyaç duyulan detay seviyesini etkileyecektir.

Bir durum çalışması, iş hedefine ulaşılabilmesi için dış aktörler (sistem dışında olup sistemle etkileşim kuran sistemler, bir kullanıcı sınıfı, kullanıcıların icra edeceği bir rol, başka bir sistem) ile yapılması düşünülen sistemin etkileşimlerini tanımlar.

Durum çalışmaları, sistemi ve sistem dışından gelen etkilere sistemin verdiği cevapları da içeren sistemle her türlü etkileşimi bir kara kutu olarak ele alır. Bununla tanımlama ihtiyaçları basitleştirildiğinden ve bir işlevselliğin nasıl sağlanacağı konusunda yorumda bulunma tuzağına düşmekten kaçınıldığından bu mantığın bir tedbir politikası olduğu kabul edilebilir.

Bir durum çalışması:

- İş hedefine hizmet eden bir iş görevini tanımlamalı,
- Uygun detay seviyesinde olmalı,
- Tek bir yazılım geliştiricinin basit bir adımda uygulayabileceği kısalıkta olmalıdır.

Durum çalışmaları, bazı projelerde işlevsel ihtiyaçların ortaya konmasında etkin bir yöntem olsa da tüm proje çeşitleri için geçerli değildir. Durum çalışmaları, kullanıcının sistemle etkileşimine odaklanır ve uç kullanıcı uygulamaları gibi bazı etkileşimlerde oldukça iyi çalışır. Ancak durum çalışmalarının, yığınsal

süreçler (batch processing), veri depolanması (data warehousing) veya karmaşık ve detaylı hesaplamalara dayanan sistemlerde olduğu gibi yoğun karmaşıklığın kullanıcı ihtiyaçlarına dayanmadığı projelerde kullanılması önemli değildir.

Durum çalışmalarının işlevsel olmayan ihtiyaçların belirlenmesinde kullanılması uygun değildir. Ancak, performans mühendisliği her bir kritik durum çalışmasının en az bir adet performans tabanlı işlevsel olmayan ihtiyaca sahip olduğunu belirlemiştir.

#### ▪ **Yazılım İhtiyaç Özelliklerinin Belirlenmesi (SRS-Software requirements specification)**

SRS, yazılımın tamamı tamamına bütün özelliklerinin apaçık şekilde ve mümkün olan en üst derecede yazılı hale getirilmesi faaliyetidir. SRS faaliyeti, zaten iyi derecede geliştirilmiş uygulamaların daha iyi anlaşılması ve bu uygulamalara ince ayar yapılması (fine tuning) için gerçekleştirilmekte ve bulunması gerekli özellikler (yetenekler) mümkün olduğunca yazılı hale getirilmektedir. Ancak, bu yapılırken emniyet ve gizlilik gerektirdiği değerlendirilen yazılım sistemlerinin uygulama geliştirme faaliyetleri öncesi ayrıştırılmasına ve dış ara yüzler için önem arz eden özelliklerin sabitlenmesine özen gösterilir.

Bir yazılım ihtiyaç özelliği; geliştirilecek sistemin bir işlevine ilişkin eksiksiz bir tanımdır ve kullanıcının yazılımla sahip olacağı bütün yetenekleri (etkileşimleri) tanımlayan aynı zamanda işlevsel ihtiyaçlar olarak da bilinen bir durum çalışması seti içerir. Durum çalışmalarına ilaveten SRS aynı zamanda işlevsel olmayan ve bütünleyici nitelikteki ihtiyaçları da içerir. İşlevsel olmayan ihtiyaçlar performans ihtiyaçları, kalite standartları ve tasarım kısıtları gibi uygulama ve tasarımda bir takım kısıtlar ortaya koyar.

Yazılım geliştirmede işlevsel bir özellik [functional specification, functional spec, specs, İşlevsel Özellik Tanımlama Dokümanı (FSD-functional specifications document) veya Program Özellikleri olarak da bilinir] bir sistemin arzu edilen hareket tarzlarını tanımlayan bir belgeleme (dokümantasyon) setidir. Belgeleme sistem kullanıcısının ihtiyaçlarının ne olduğunu (tasarım hedefi), hangi iç işlevlerin gerektiğini ve bunun yanında yazılım sisteminin arzu edilen girdi ve çıktı özelliklerinin tanımlanması anlamına gelir. Bir tasarım dokümanı olarak, bir işlevsel özellik önerilen sistemin iç işlemlerini tanımlamaz ve sistem işlevinin nasıl uygulanacağına ilişkin özelliklerin halen içerikte bulunmadığı anlamına gelir. Aksine,



sistemle etkileşim esnasında çoğu dış unsurun (programı kullananlar, bilgisayar ortamı veya diğer bilgisayarlar gibi) ne gözlemleyeceği üzerine odaklanır.

Örnek bir işlevsel tanımlama aşağıdaki gibi olabilir.

*“Kullanıcı “Tamam” düğmesine bastığında, diyalog kutusu kapanacak ve kapanan diyalog görüntülenmeden önceki ana ekran ortamına dönüş sağlanacaktır.”*

Bu gibi bir ihtiyaç, kullanıcı ve yazılım sistemi arasında bir etkileşimi tanımlar. Kullanıcı “Tamam” tuşuna basarak sisteme veri girdiğinde, program “Tamam” düğmesinin de üzerinde bulunduğu diyalog penceresini kapatarak cevap vermelidir.

İhtiyaçların; geliştiricinin bakış açısına göre hazırlanmış kullanıcı el kitabı olarak kabul edilebildiği durumlarda gayri resmi ve programlama veya matematiksel terimlerle tanımlanan belirli anlamlar yüklendiği durumlarda da resmi olarak kabul edilebilir. Genellikle önceden belirlenmiş olan ve kritik emniyet gerektiren yazılım sistemleri kullanılmasına rağmen en başarılı özellikler, oldukça yüksek seviyede geliştirilmiş yazılımlara ince ayar yapılması ve yazılımın daha iyi anlaşılması için yazılı hale getirilmektedir. Özellikler değişmemesi gereken dış arayüzler için en önemli husustur.

İşlevsel ihtiyaç özelliklerinin belirlenmesinin birçok amacı vardır. Takım projelerindeki başlıca amaçlardan biri çok zaman alan test durumlarının (test cases) ve kaynak kodlarının (source code) yazılması ve bunları müteakip geliştirilmekte olan sistem hatalarının giderilmesi vb. faaliyetlerin icrası öncesi programın ne yapması gerektiği konusunda takım içinde görüş birliğinin sağlanmasıdır. Bu görüş birliği, yazılım ihtiyaçlarının karşılayabilecek maliyet etkin bir yol üzerinde anlaşılmasını müteakip, mevcut projede bir veya daha fazla kez kullanıcı görüşlerinin alınması ile sağlanabilmektedir.

Sipariş edilen endüstriyel yazılım mühendisliği ömür devrinde (şelale modeli, waterfall model), işlevsel özellikler hangi hususların uygulanması gerektiğini belirler. Müteakiben sistem özellikleri dokümanı işlevlerin belirlenmiş bir yazılım ortamı kullanılarak nasıl anlaşılacağını ortaya koyar. Prototipli sistem geliştirme gibi endüstriyel olmayan durumlarda işlevsel özellikler ihtiyaç analizinin bir parçası olarak veya ihtiyaç analizi sonrasında belirlenir.

Yazılım ihtiyaç özellikleri konusundaki tavsiye edilen yaklaşımlar IEEE 830-1998 standardı içinde belirtilmiştir. Bu standart, yazılım ihtiyaç özelliklerinin muhtemel yapısı, arzu edilen içeriği ve kalitesine ilişkin standartları tanımlamaktadır.

### 2.3.5. İhtiyaç Analiz Sorunları

- **Kullanıcılara İlişkin Sorunlar**

Steve McConnell, "Rapid Development" (McConnel 1996) adlı kitabında, ihtiyaç topalayan kullanıcının karşılaştığı sorunlara ilişkin aşağıdaki detayları sunmaktadır.

- Kullanıcılar genellikle ne istediklerini bilmezler veya kendi ihtiyaçları konusunda açık fikirleri yoktur.
- Kullanıcılar yazılı bir ihtiyaç setine bağlı kalmazlar
- Kullanıcılar, maliyet ve program sabitlendikten sonra yeni ihtiyaçlar olduğu ve karşılanması gerektiği konusunda ısrarlı olurlar.
- Kullanıcılarla iletişim yavaştır.
- Kullanıcılar ikili görüşmelere katılmazlar veya ellerinde olmayan sebeplerle katılamazlar.
- Kullanıcılar teknik olarak tecrübesiz ve yetersizdir.
- Kullanıcılar geliştirme sürecini anlamazlar.
- Kullanıcılar mevcut teknoloji konusunda bilgi sahibi olmadıklarından sistem veya ürün geliştirme faaliyetleri başladığında bile kullanıcı ihtiyaçlarının değiştiği bir duruma sebebiyet verirler.

- **Mühendis/Geliştiricilere İlişkin Sorunlar**

Mühendis veya geliştiricilerin ihtiyaç analizinde sebep oldukları sorunlar (Wieggers 2003) aşağıda açıklanmıştır.

- Teknik personel ve uç kullanıcılar farklı terminoloji kullandıklarından, sonuçta bitmiş ürün temin edilene kadar kusursuz bir görüş birliği sağladıkları hatasına düşerler.

- Mühendis veya geliştiriciler müşteri ihtiyaçlarına özel bir sistem geliştirmekten çok ihtiyaçları mevcut bir sistem veya modele uydurmaya çalışırlar.

- Analizler, müşteri ihtiyaçlarını anlayabilecek bilgi ve beceriye sahip personelden çok genellikle mühendis veya programcılar tarafından gerçekleştirilir.

- **Çözüm Girişimleri**

İletişim problemlerine ilişkin bir çözüm girişimi olarak sistem analizi veya iş konusunda uzman kişilerin işe alınması gösterilebilir.

1990'larda ortaya atılan Prototip hazırlama (Prototyping), Birleştirilmiş Modelleme Dili (UML-Unified Modeling Language), durum çalışmaları (Use cases) ve hızlı yazılım geliştirme (Agile software development) gibi teknikler önceki metotlardaki karşılaşılan problemlere bir çözüm getirilmesi için üretilmiştir.

Aynı zamanda pazara yeni bir sınıf uygulama simülasyonu veya uygulama tanımlama araçları girmiştir. Bu araçlar, iş kullanıcıları ile bilgi teknolojileri (IT) şirketleri arasında iletişim eksikliğinin giderilmesi ve aynı zamanda her hangi bir kod yazılmadan yazılımların test pazarına sunulmasına imkân tanımaktadır. Bu araçların en iyileri aşağıdaki yeteneklere sahiptir (Stellman ve Greene 2005).

- Uygulama iş akışlarının ve test alternatiflerinin krokilenmesini sağlayan elektronik beyaz tahtalar,

- İş mantığı ve veri ihtiyaçlarını belirleme yeteneği

- Uygulamanın son halini kolayca taklit edebilecek yüksek kalitede prototip hazırlama yeteneği,

- Ortak hareket etme ve etkileşim,

- Diğer yorum ve anlamdan çıkarılan ihtiyaçların eklenebilmesi yeteneği,

- Uzaktaki ve dağınık durumdaki kullanıcı gruplarının simülasyon ile etkileşim başlatabilme yeteneği.

### 2.3.6. İhtiyaç Kapsamı

Kullanıcıdan alınan genel ihtiyaçlar sabırla seçilip ayrıldıktan sonra geliştirmenin kapsamına ilişkin analiz yapılmalı ve sonuçları “Kapsam Dokümanı” şeklinde açıkça ortaya konulmalıdır. Geliştirme başlangıcında iyi açıklanmamış ihtiyaçların veya maliyet fonksiyonunun bir sonucu olarak belli işlevsel özellikler kapsam dışında kalmış olabilir. Geliştirme kuruluş dışında bir firmaya yaptırılıyorsa, ihtiyaç kapsam dokümanı; şüpheler veya farklı yorumlanabileceği değerlendirilen hususlar olduğunda, sözleşme kapsamında kullanıcıya garanti edilen hususların açıklanmasında resmi bir doküman olarak kabul edilebilir.

### 2.4. Yazılım Mimarisi ve Tasarımı

Yazılım mimarisi, sistemi soyut olarak gösteren ve yazılım bileşenlerini bu bileşenlerin dıştan görülebilen özelliklerini ve birbirleri arasındaki ilişkileri ortaya koyan sistem ve yapılarıdır. Yazılım mimarisi ilgi alanında, yazılım sisteminin ürün ihtiyaçlarının karşılanması yanında gelecekteki muhtemel ihtiyaçların da anlaşılması yer alır. Başka bir deyişle; yazılım sistemi ile diğer yazılım sistemleri arasında karşılıklı iletişimi (veri aktarımı) sağlayacak ara yüzlerin tanımlanması yanında, mevcut donanım durumuna ve kullanıcının işletim sistemine uygun şekilde planlama yapılması esastır. Yazılım mimarisi terimi aynı zamanda sistemin yazılım mimarisinin belgelenmesi (dokümantasyonu) anlamında da kullanılmaktadır. Yazılım mimarisinin belgelenmesi kullanıcılar arasındaki iletişimi kolaylaştırır, yüksek seviye tasarım konusunda alınan erken kararları yazılı hale getirir ve iki proje arasında tasarım bileşen ve modellerinin yeniden kullanılabilmesine imkân tanır (Bass ve Clements 2003:21-24).

Bilgisayar bilimi, ortaya çıktığından beri karmaşıklığa ilişkin problemlerle karşılaşmıştır (University of Waterloo 2006). Ortaya çıkan ilk karmaşıklık problemleri geliştiriciler tarafından doğru veri yapılarının seçilmesi, algoritmaların geliştirilmesi ve ilgi alanlarının ayırt edilmesi (separation of concerns) ile çözülmüştür. Yazılım mimarisi terimi endüstriyel anlamda oldukça yeni bir terim olmasına rağmen, 1980’lerin ortalarından beri yazılım mühendisliğinin öncüleri tarafından zaman zaman temel prensipleri uygulanmıştır. İlk sistem yazılım mimarisini ortaya koyan ve açıklayan yaklaşımlar, şekilsel olarak yetersiz ve dağınık olmaları yanında genellikle kutu ve çizgi diyagramları seti şeklinde gösterilmiştir (IEEE 2006). Yazılım disiplininin temel özelliklerinin tanımlanması ve kodlanması konusundaki asıl yoğun çaba 1990’larda sarf edilmiştir. İlk tasarım modelleri, stilleri, en iyi uygulamaları,

tanımlama dilleri ve resmi mantık setleri bu dönemde geliştirilmiştir. Yazılım mimarisi bilimi, soyutlama (abstraction) ve ilgi alanlarının belirlenmesi sayesinde karmaşıklığın azaltılması konusuna odaklanmıştır. Yazılım mimarisi teriminin hâlihazırda kabul görmüş basit bir tanımı bulunmamaktadır (SEI 2006).

Sistem kurmada doğru yolu göstereceği değerlendirilen ve bilinen/anlaşılabilen hiç bir kuralı olmayan yazılım mimarisi tasarımı gittikçe gelişmesine rağmen halen daha sanat ile bilim karışımı bir yapıdadır. Ticari bir yazılım sistemi bir misyonu veya bir iş alanını desteklediğinden bir sanat olarak değerlendirilmektedir. Bir sistemin önemli iş faaliyetlerini nasıl desteklediği, sistemin işlevsel olmayan ihtiyaçlarının (aynı zamanda kalite seviyesi olarak da bilinir) karşılanabilmesi için sistemin nasıl hareket edeceği hazırlanan senaryolarla tanımlanır. Her sistem, desteklediği iş faaliyetlerinin doğasına bağlı olarak farklı yapıdadır [hata toleransı, geriye dönük uyumluluk (backward compatibility), genişletilebilirlik (extensibility), güvenilirlik (reliability), bakım ve idame edilebilirlik (maintainability), güvenlik (security), kullanılabilirlik (usability), mevcudiyet (availability) ve her bir uygulamaya göre değişen diğer özelliklerde olduğu gibi, bir sistem tarafından sergilenen kalite seviyelerinin derecesi örneğinde olduğu gibi] (SoftwareArchitectures.com).

Kullanıcı bakış açısının yazılım mimarisine kazandırılması kapsamında, her bir kullanıcının (yazılım sistemi kullanıcıları, yazılım geliştiriciler, yazılım sistemi faaliyet destek grubu, yazılım bakım ve idame uzmanları, uygulayıcılar, test ediciler ve uç kullanıcılar) kendi özel ortamında veya ilgi alanında görevlerin icra edilmesi ve adımların atılması için yönlendirici direktifler verdiği söylenebilir. Bu açıdan bakıldığında yazılım mimarisinin gerçekte sistemin temsil ettiği çok sayıda bakış açısının birleştirilmesi sayesinde oluşturulduğu görülebilir. Bir projede ihtiyacın ispatı ve yazılım mimarisinin doğrulanması gibi mimari içinde bir araya konan çok sayıda farklı bakış açıları olgunlaşmayı sağlar.

#### **2.4.1. Mimari Tanımlama Dilleri (ADLs-Architecture Description Languages)**

Yazılım mimarisinin tanımlanması kapsamında farklı kuruluşlar tarafından [AADL (SAE standardı), Wright (Carnegie Mellon tarafından geliştirildi), Acme (Carnegie Mellon tarafından geliştirildi), xADL (UCI tarafından geliştirildi), Darwin (Imperial College London tarafından geliştirildi), DAOP-ADL (Málaga Üniversitesi tarafından geliştirildi)] çok sayıda farklı tanımlama dili geliştirilmiştir. Bir tanımlama dilinin ana elemanları; bileşen, birleştirici ve konfigürasyondur.

## 2.4.2. Görünümler

Yazılım mimarileri genellikle yapı mimarisinde de kullanılan mavi kopyalara (blueprints) benzer yapıda görünümler şeklinde düzenlenmiştir. (Clements, Bachmann, Bass, Garlan, Ivers, Little, Nord ve Stafford 2003:13-15). ANSI/IEEE 1471-2000 standardı ile ortaya konan yapı içinde, görünümler mimarinin kullanıcılar ve onların ilgi alanlarına yönelik bakış açıları tarafından sorgulanarak tanımlandığı bakış açılarının anlık görüntüleri olarak ifade edilebilir.

ANSI/IEEE 1471-2000 standardı ile ortaya konan bazı muhtemel görünümmler aşağıda sıralanmıştır.

- İşlevsel/Mantık Görünüm
- Kod/Modül Görünümü
- Geliştirme/Yapısal Görünüm
- Mutabakat/Süreç/Çizgisel Görünüm
- Fiziksel/Uygulama Görünümü
- Kullanıcı Faaliyeti/Geri Besleme Görünümü
- Veri Görünümü

Yazılım mimarisinin tanımlanması için birçok dil geliştirilmesine rağmen hangi sembol seti veya hangi görünüm sisteminin kullanılması gerektiği konusunda görüş birliği sağlanamamıştır. UML sadece yazılım değil sistemlerin modellenmesi için geliştirilmiş bir standarttır ve yazılım mimarisindeki görünümlere uygulanır. Bazıları yazılımların etkili şekilde geliştirilebilmesi için her bir problemin her bir kısmının anlaşılması gerektiğini düşündüklerinden, evrensel işaretlemeler (notasyon) geliştirilmiştir. Çünkü, her biri bazı görev setleri için notasyonları kullanışsız veya tehlikeli hale getiren bir işaretsel temel (notational bias) sunar.

Stratejik tasarım olarak da adlandırılan yazılım mimarisi, aynı zamanda Programlama paradigmaları (programming paradigms), mimari stiller, bileşen esaslı yazılım mühendisliği (component-based software engineering) standartları ve tasarım prensipleri gibi genel tanımlama kısıtları ile ilgilenen bir faaliyettir. Taktik tasarım olarak da bilinen ayrıntılı tasarım ise; tasarım dokuları (design patterns), mimari dokular (architectural patterns), programlama deyimleri (programming idioms) ve yeniden esas belirleme (refactorings) gibi yerel tasarım kısıtları ile ilgilenen bir faaliyettir.

Mimari bir tasarımıdır, ancak her tasarım bir mimari değildir. Gerçekte, yazılım mimarisi ile ayrıntılı tasarım arasındaki çizgiyi belirleyen mimarın kendisidir. Her duruma uyacak bir şablon ise yoktur.

### 2.4.3. Mimari Stil ve Doku Örnekleri

Bilgisayar yazılım modüllerinin ve birbirleri arasındaki iletişimin tasarımı konusunda uygulanabilecek yöntemler aşağıda sıralanmıştır.

- Karatahta (Blackboard)
- İstemci-Sunumcu (Client-server)
- Veri Tabanı Merkezli Mimari (Database-centric architecture)
- Dağıtılmış Hesaplama (Distributed computing)
- Faaliyet Esaslı Mimari (Event Driven Architecture)
- Tam Mana (Implicit invocation)
- Yekpare Uygulama (Monolithic application)
- Uçtan Uca Mimari (Peer-to-peer)
- Boru ve Fitreler (Pipes and filters)
- Eklenti (Plugin)
- Parçacıklı Mimari [Structured (modül esaslı yapıda olup genellikle bütünseldir/yekpare yapıdadır)]
  - Bileşenli Yazılım [Software componentry (tamamına yakını modül esaslı olup, modül içinde genellikle nesne tabanlı programlama (object-oriented programming) kullanılır ve çok az oranda bütünseldir)]
    - Hizmet Yönelimli Mimari (Service-oriented architecture)
    - Araştırma Yönelimli Mimari (Search-oriented architecture)
    - Alan Esaslı Mimari (Space based architecture)
    - Paylaşım Kapalı Mimari (Shared nothing architecture)
    - Üç Katmanlı Model (Three-tier model) **(Bu model geliştirilen veri tabanı modeli tarafından da kullanılmaktadır).**

#### 2.4.4. Yazılım Tasarımı

Yazılım Tasarımı, bir yazılım çözümü planlaması ve problem çözme sürecidir. Yazılımın amaç ve özellikleri belirlendikten sonra, yazılım geliştiriciler bir çözüm planı geliştirmeleri için tasarımcıları görevlendirir veya kendileri tasarlar. Mimari görünüm yanında alt seviye bileşen ve algoritma uygulama konularını da içerir.

Yazılım geliştirme sürecinin yazılım ihtiyaçları analizi (SRA-software requirements analysis) adımı yazılım mühendisliğinde kullanılan özellikleri işaret eder. Yazılım yarı otomatik veya kullanıcı merkezli olduğunda, yazılım tasarımı bu özelliklerin belirlenmesi kapsamında kullanıcı tecrübesine dayalı tasarımı (user experience design) olarak nitelenen ve kullanıcı deneyimlerini esas alan metotlardan faydalanabilir. Yazılım tamamen otomatik olduğunda (kullanıcı veya kullanıcı arayüzüne ihtiyaç olmadığı) yazılım tasarımı olayların akış sırasını tanımlayan metin veya bir iş akış çizelgesi gibi basit yapıda olabilir. Tekleştirilmiş Modelleme Dili (Unified Modeling Language) ve Temel Modelleme Konseptleri (Fundamental modeling concepts) gibi yarı standart metotlar da mevcuttur. Her iki durumda da plana ait bazı dokümanlar genellikle tasarım ürünü olarak ortaya çıkarlar.

Yazılım tasarımı, tasarımın ihtiyaç duyduğu teknolojinin mevcudiyetine bağlı olarak her hangi bir platformdan (yazılım geliştirme metodu) bağımsız (platform-independent) veya bir platforma özel (platform-specific) olarak düzenlenmiş olabilir.

##### 2.4.4.1 Tasarım Esasları

Bir yazılım parçasının tasarımında dikkate alınması gerekli bir çok husus mevcuttur. Her biri yazılımın ulaşmaya çalıştığı bir veya daha fazla hedefi yansıtması bakımından önem arz eden bu hususlardan bazıları aşağıda sıralanmıştır.

- **Pazarlanabilirlik (Marketability):** Yazılımın yüksek oranda pazarlanabilmesi için bir "Pazar" mevcut olmalıdır. Bu kapsamda, hedef pazarın ve onun ihtiyaçlarının belirlenmesi maksadıyla araştırma yapılması gereklidir.
- **Kullanılabilirlik (Usability):** Yazılım kullanıcı arayüzü, hedef kullanıcının ihtiyacını karşılayabilecek şekilde kullanımı kolay ve genellikle de estetik açıdan tatmin edici bir yapıda olmalıdır.



- **Paketleme (Packaging):** Kutu veya kullanım klavuzu gibi basılı malzemeler kullanılabilirliği artırmalı ve hedef pazarın ihtiyaçlarına ve stiline uyacak şekilde düzenlenmelidir.
- **Niteliksel Genişletilebilirlik (Extensibility):** Yeni yetenekler ana mimarinin çizdiği hattı bozmayacak şekilde yazılıma kolayca eklenebilmelidir.
- **Sezinleme ve Zindelik (Robustness):** Yazılım önceden tahmin edilemez veya geçersiz olan girdilere müsamaha edebilmeli veya yüksek iş yoğunluğunda çalışabilmelidir. Örneğin, yazılımı yavaşlatan düşük hafıza durumlarını kolayca ortadan kaldırarak yazılımı eski hızına döndürebilen bir tasarım sunabilir.
- **Güvenilirlik (Reliability):** Yazılım belirlenmiş bir zaman diliminde belli durumlar için ihtiyaç duyulan bir işlevi doğrulukla gerçekleştirebilmelidir.
- **Hata Toleransı:** Yazılım, bileşenlerinin çalışmamasının önüne geçebilmeli veya eski haline kolayca döndürebilmelidir.
- **Güvenlik (Security):** Yazılım, kendisini dostane olmayan işlevler ve etkilere karşı korumalıdır.
- **İdame Edilebilirlik (Maintainability):** Yazılım belirli bir zaman aralığında kendini yenileyebilmeli veya etkinliğini koruyabilmelidir. Örneğin, bir antivirüs yazılımı yazılımın etkinliğinin korunması için sürekli olarak belirli zaman aralıklarıyla virüs tanımlarını güncelleyebilir.
- **Uyumluluk (Compatibility):** Yazılım başka bir yazılımla birlikte çalışılabilirliğin sağlanabilmesi için tasarlanmış diğer ürünlerle beraber çalışabilmelidir. Örneğin, bir yazılım, kendisinin eski bir versiyonunun üzerine kurulabilecek ve gerektiğinde geri alınabilecek uyumlulukta olabilir.
- **Çok Bölümlülük (Modularity):** Yazılım tamamlandığında iyi tanımlanmış ve birbirinden bağımsız yapıda bileşenlerden oluşmalıdır. Bu, sistemin bakım ve idamesini kolaylaştırır ve bileşenler bu sayede, arzu edilen yazılım sistemine entegre edilmeden önce parça parça test edilebilir ve denenebilirler. Bu da yazılım projesi geliştirilmesinde iş bölümüne imkân tanır.
- **Tekrar Kullanılabilirlik (Reusability):** tek başına çalışabilir yapıda tasarlanmış bileşenler, kendilerinden beklenen işlevselliği ne eksik ne de fazla olacak şekilde yerine getirmelidir. Başka tasarımlarda benzer ihtiyaç olduğunda, tek

bir amaca hizmet etmek için hazırlanmış bileşenlerin bu sayede diğer tasarımlardaki benzer ihtiyaçlar için tekrar kullanılması sağlanabilir.

#### **2.4.4.2 Tasarım Dokuları (Design Pattern)**

Bir yazılım tasarımcısı veya mimarı başkaları tarafından daha önce çözümlenmiş bir tasarım problemi belirleyebilir. Ortak bir problemin çözümünü tanımlayan şablon veya dokuya tasarım dokusu adı verilir. Bu gibi dokuların tekrar kullanımı, önceden test edildikleri ve doğrulamaları yapıldığı için yazılım geliştirme sürecini hızlandırabilir.

#### **2.4.4.3 Tasarım Metodolojileri**

Tasarım metodolojileri, bir sistemin gerçek tasarımı için bir şablon oluşturulmasını, tasarım kalitesini iyileştiren bazı standart tasarım prensiplerinin güçlendirilmesini ve gerçek sistem tasarım sürecinin basitleştirilmesini amaçlar. İlk tasarım metodolojilerinden biri; mimari ve teknoloji uygulama seçimini resmeden ve teknolojiden bağımsız bir tasarım üretmeyi hedefleyen Durum Çalışması, Sorumluluğa Dayalı Analiz ve Tasarım (URDAD- Use Case, Responsibility-Driven Analysis and Design) metoduna esas oluşturan ve Rebecca WIRTH'ün öncülük ettiği Sorumluluğa Dayalı Tasarım (RDD-Responsibility Driven Design)'dir

### **2.5. Kodlama (Code Implementation)**

Uygulamalı matematiğin (applied mathematics) bir kolu olarak da kabul edilen programlama/kodlama, bilgisayar programlarının bir programlama diliyle geliştirilen kaynak kodlarının yazımı, sistematik olarak testi, hatalarının ayıklanması ve idamesi sürecidir. Kod, yeni bir sistem için olabileceği gibi mevcut bir sistemin değiştirilmesi, yenilenmesi, yeni özellik kazandırılması veya kısmi olarak düzenlenmesi için yazılıyor olabilir. Programlamanın amacı, belirli arzu edilen bir davranışı (özelleştirme-costumization) gösteren bir program hazırlanmasıdır. Kaynak kodu yazım süreci uygulama ana odağı (application domain), özelleştirilmiş algoritmalar (specialized algorithms) ve resmi mantık (formal logic) gibi farklı birçok konuda uzmanlık gerektirir. Yazılım mühendisliğinde programlama, yazılım geliştirme sürecinin bir safhası olarak değerlendirilir.

Program yazımının bir sanat mı, bir araç mı yoksa bir mühendislik disiplini mi olduğu açılımında tartışmalar halen devam etmektedir. İyi programlama etkin ve idamesi kolay bir yazılım çözümünün üretimi hedeflenerek sanat, araç ve mühendislik boyutlarının ölçülebilir uygulaması olarak değerlendirilebilir (etkin ve

idame edilebilme kriterleri duruma ve kişilere göre büyük farklılıklar gösterebilir). Bu disiplin, programcıların kendilerini programcı veya yazılım mühendisi olarak gösterebilmeleri için, genellikle her hangi bir standart (veya kanunlarla düzenlenmiş) onama testini geçme veya lisans alma ihtiyacı duymadıkları ve diğer birçok teknik meslekten farklı bir yapıdadır.

Program yazımında kullanılan programlama dilinin yazılımın alacağı son hali etkilemesi kapsamında başka bir tartışma daha vardır. Bu tartışma özel bir dilin doğasının onu kullananların düşünce yapılarından etkilendiğini ifade eden dil biliminde Sapir-Whorf Varsayımını (Sapir-Whorf hypothesis) (Iverson 1980:444-465) içine alan tartışmaya benzer yapıdadır. Farklı dil dokuları farklı düşünce dokularını kabul eder.

Farklı programlama dilleri farklı programlama stillerini (programming paradigms) destekler. Kullanıcak dilin seçimi şirket politikası, göreve uygunluk, üçüncü taraf paketlerinin mevcudiyeti veya bireysel tercih gibi birçok hususun dikkate alınmasını gerektirir. İdealde icra edilecek göreve en iyi uyacak programlama dilinin seçilmesi yanında, bir tim oluşturmak için programlama dilini bilen yeteri kadar programcı bulunması, bahse konu programlama dili için derleyicilerin (compilers) mevcudiyeti ve belirlenen dille programın yazılmasıyla elde edilecek etkinlik şeklindedir.

Allen DOWNEY, Bilgisayar Bilimci Gibi Nasıl Düşünülür? (*How To Think Like A Computer Scientist*) adlı kitabında şu yorumu yapmıştır:

“Detaylar farklı dillerde farklı gözükmesine rağmen bir kaç temel açıklama hemen her dilde ortaya çıkar; Girdi (input): klavyeden, bir dosyadan veya başka bir cihazdan veri alınması. Çıktı (output): verilerin ekranda görüntülenmesi veya verilerin başka bir cihaz veya dosyaya gönderilmesi. Matematik (math): ekleme ve çarpma gibi bazı temel matematik hesaplamaların gerçekleştirilmesi. Durumlara göre icra (conditional execution): belli durumların kontrolü ve oluş sırasına uygun şekilde talimatların yerine getirilmesi. Tekrar (repetition): bazı faaliyetlerin genellikle ufak tefek farklılıklarla tekrarlanması.”

Çoğu programlama dili, kullandığı veri kütüphaneleri tarafından gerçekleştirilen işlevleri isimlendiren bir mekanizmaya sahiptir. Bir veri kütüphanesindeki işlevler, uygun çalışma zamanı sırasını izlediğinde her hangi bir dilde yazılabilir.

Modern programlama dillerinden hangisinin en iyi olduğunu belirlemek oldukça güçtür. Bazı diller farklı uygulama çeşitlerinin yazımında kullanılabilirken, bazıları da özel uygulama çeşitlerinde kullanılmaktadır. (Örn.: COBOL daha çok büyük ana yapılar içeren ortak veri tabanlarında, FORTRAN mühendislik uygulamalarında ve "C" birbiri içine girmiş uygulamalarda en popülerdir).

Dil popülerliğini ölçebilecek yöntemlerden bazıları aşağıda sunulmuştur:

- Dili belirleyen bütün iş ilanları miktarının belirlenmesi,
- Dili öğreten kitapların satış adedi (bu, yeni dillerin önemini yüksek gösterebilir)
- Dille yazılmış mevcut kod satırlarının miktarının tahmin edilmesi (bu, COBOL gibi iş dillerini kullananların sayısını eksik tahmin edebilir)

Uygulama teknikleri içinde zorunlu diller (imperative languages) [nesne tabanlı, (object-oriented) veya işlemsel (procedural)], işlevsel diller (functional languages) ve mantık dilleri (logic languages) yer alır.

### 2.5.1. Kodlama Kalite İhtiyaçları

Yazılım, kullanılacak geliştirme yaklaşım ne olursa olsun, küçük program hatalarının ayıklanması (debugging), müteakip geliştirme ve kullanıcı desteğine bağlı şekilde zaman ve/veya para olarak programlama maliyetlerinin düşürülmesi gibi akılda tutulması gereken bazı temel özellikler sunmalıdır. Kodlama kalitesi çok sayıda yöntemle sağlanabilmektedir. Ancak, hemen her programda bulunan en geçerli beş özellik aşağıda sunulmuştur.

- **Etkinlik (Efficiency):** Sistem kaynak kullanımının en düşük seviyede tutulması (bilgisayar işlemcisi, hafıza, yavaş cihazlar, ağlar ve hatta bazı durumlarda kullanıcı ara yüzü) olarak tanımlanır.
- **Güvenilirlik (Reliability):** Programın sunduğu sonuçlar; doğru kodlamayla ulaşılmış olması yanında, tipik hata (Örn: veri dönüşümünden kaynaklanan hatalar, sıfırla bölüm hataları, veri akışı zayıflığı (underflow) ve aşırı veri akışı (overflow) üretimini (error propagation) azaltacak ve/veya önleyecek şekilde tam doğru olmalıdır
- **Sezinleme ve Zindelik (Robustness):** Bir program, çalışma zamanı hataları ve programın durmasıyla sonuçlanabilecek veri tipi çatışmaları gibi diğer

tüm uyumsuzlukları önceden tahmin etmelidir. Bu bakış açısında, kullanıcı ile etkileşimin gerçekleştirilmesi ve hata mesajlarının elleçlenebilmesi önem arz eder.

- **Taşınabilirlik (Portability):** Yeniden kodlama yapılmadan programın başka yazılım ve donanım ortamında çalışabilmesi yeteneğidir.
- **Okunabilirlik (Readability):** Ana programın ve her bir alt sisteminin amacı, uygun ifadeler kullanılarak ve kendinden açıklamalı sembolik isim (sabit, değişken, işlev adı, sınıflar, metotlar, vb.) seçimi ile çok net şekilde tanımlanmalıdır.

### 2.5.2. Hata Ayıklama (Debugging)

Hatalı bir program kullanıcıları için oldukça önemli sonuçlar doğurabileceğinden, hata ayıklama yazılım geliştirme sürecinde çok önemli bir faaliyettir. Bazı diller, diğerleri gibi çok sayıda kontrol gerçekleştiren derleyicilere (compilers) ihtiyaç duymadıklarından bazı hata çeşitlerine açıktır. Statik analiz (static analysis) aracı kullanımı bazı muhtemel problemlerin tespitinde kullanılabilir.

Hata ayıklama genellikle Visual Studio, NetBeans, ve Eclipse gibi entegre tasarım elemanlarıyla (IDEs) yapılır. GDB gibi bağımsız hata ayıklayıcılar da kullanılmaktadır

### 2.5.3. Algoritmik Karmaşıklık

Verilen bir problem grubu için en uygun algoritmanın bulunması ve uygulanması; bilgisayar programlama konusundaki hem akademik çalışmalar ve hem de mühendislik uygulamalarında üzerinde önemle durulan hususlardan biridir. Bu amaçla, algoritmalar Büyük "O" Gösterimi (Big O notation) olarak da bilinen talimatlar (orders) şeklinde sınıflanmıştır. "O", icra zamanı ve hafıza kullanımı gibi kaynak kullanımını girdinin ebatları şeklinde ifade eder. Uzman programcılar, iyi kurgulanmış algoritma çeşitlerine ve getirdikleri karmaşıklıklara aşinadır ve bu bilgilerini değişen durumlara en iyi uyacak algoritmanın seçiminde kullanırlar.

## 2.6. Test ve Muayene

Test ve muayene faaliyetleri yazılım geliştirme sürecinin bütünsel ve önemli bir bölümünü oluşturur. Bu aşamada sistem içindeki hataların (bugs) mümkün olduğunca erken teşhis edilmesi sağlanır.

Yazılım test ve kontrolü, yazılımın ihtiyaç duyulan işlevleri ihtiyaç analiz safhasında belirlenen ve kullanıcı ile geliştirici arasında üzerinde ortak mutabakat

sağlanan yazılım işlev ihtiyaçlarının karşılanıp karşılanmadığının belirlenmesi ve hataların tespit edilmesi bakımında kontrol edilme sürecidir.

Yazılım test ve kontrolü, test edilen hizmet ya da ürünün (içinde kullanılacağı değerlendirilen içeriğe uygun olarak) kalitesi hakkında kullanıcıya bilgi sağlamak için icra edilen tecrübî bir araştırmadır (Kaner 2006). Yazılım hatalarının bulunması amacıyla yazılım veya programın işletilmesi sürecini içerir. Ancak bununla sınırlı değildir.

Kalite bir zorunluluk olmamasına rağmen bazı kişilere bir değer ifade ettiği düşüncesi akılda tutularak, yazılım test ve kontrolü asla rasgele bir bilgisayar yazılımının tamamen doğru olduğunu ortaya koyamaz. Yazılım test ve kontrolü, bir özellik açısından ürünün hareket tarzını ve durumunu kıyaslayan bir eleştiri veya kıyas sağlar. Burada önemli bir nokta da yazılım test ve kontrolünün, sadece test olmayıp bütün iş süreç alanlarını çevreleyen bağımsız disiplin olan Yazılım Kalite Garantisi (SQA-Software Quality Assurance)'ninden farklı olması gerektiğidir.

Bütün yazılım ürünleri hedef bir dinleyici kitlesine sahiptir. Örneğin; video oyunu yazılımı dinleyicileri bankacılık yazılımı dinleyicilerinden tamamen farklıdır. Bu nedenle, bir kuruluş bir yazılım ürünü geliştirmeye başlamadan veya ona yatırım yapmadan önce, uç kullanıcıları, alıcıları veya hedef dinleyicileri açısından bahse konu yazılımın uygun olup olmadığını değerlendirmesi gereklidir. Yazılım test ve kontrolü bu değerlendirmenin yapılması için girişimde bulunma sürecidir.

NIST tarafından 2002 yılında hazırlanan bir rapora göre yazılım hataları ABD ekonomisine yıllık bazda 59,6 milyon dolara mal olmaktadır. Bu maliyetin üçte birinden fazlasından daha iyi yazılım test ve kontrolü gerçekleştirilerek kaçınılabılır (NIST 2002).

### **2.6.1. Kapsam**

Test etmenin ana amaçlarından biri de yazılımın kontrol edilerek hataların bulunması ve düzeltilmesidir. Test etme, bir ürünün belli koşullar altında düzgün çalışmadığını gösterebilse de her koşulda düzgün çalışıp çalışmadığını ortaya koyamaz (Falk ve Nguyen 1999). Yazılım testinin kapsamı genellikle kodlamanın farklı ortam ve koşullarda yapılması ve kod kalitesinin kontrolü yanında “bekleneni veriyor mu? Yapması gerekeni yapıyor mu?” sorularına cevap verecek şekilde kodun doğruluk yönünden kontrol edilmesi faaliyetlerini içerir. Yazılım geliştirme kültüründe genellikle testi yapan kurum ya da kuruluşlar geliştirme ekibinden farklıdır

ve testi gerçekleştirecek ekip üyelerinin farklı görevleri vardır. Yazılım testinden elde edilen bilgi yazılımın geliştirildiği süreçlerin düzeltilmesinde kullanılabilir.

### 2.6.2. Hatalar ve İflaslar

Yazılım hataları şu şekilde oluşur: Programcı bir yanlış yapar, yanlış yazılım kaynak kodunda hataya (defect) sebebiyet verir. Bu hataya rağmen uygulamaya geçildiğinde, belli durumlarda sistem iflasa sebebiyet veren yanlış sonuçlar üretir (ISTQB, 2009). Bütün hatalar iflasa neden olmazlar (Örn.: Ölü kod (dead code) olarak nitelendirilen yazılım içinde her hangi bir yerde kullanılmamasına rağmen önceden yazılmış ancak silinmesi veya temizlenmesi unutulmuş kodlar iflasa sebebiyet vermezler). Yazılımın yeni bir donanım üzerinde işletilmesi, kaynak kodunda değişiklik yapılması veya farklı yazılımlarla etkileşim sağlanmasını da içeren ortam değişiklikleri bunlara örnek olarak gösterilebilir (ISTQB, 2009).

### 2.6.3. Girdi Kombinasyonları ve Ön Hazırlıklar

Yazılım test ve kontrolünde karşılaşılan bir problem basit bir ürünün bile bütün girdi çeşitleri ve ön hazırlıkları kullanılarak test ve kontrolünün yapılmasının uygulanabilir olmamasıdır (Falk ve Nguyen 1999, ISTQB 2009). Bu; yazılım ürünündeki hataların çok büyük olabilmesi ve hatalar çok sık ortaya çıkmadığından, bulunarak test edilmesinin neredeyse imkânsız olması anlamına gelir.

### 2.6.4. Statik ve Dinamik Test ve Kontrol

Yazılım test ve kontrolü konusunda birçok yaklaşım mevcuttur. Gözden geçirme (Review), Prova Etme (walkthrough) veya inceleme (inspection), statik test (static testing), programlanmış kaynak kodunun hazır verilmiş test meseleleriyle (test cases) denenmesi ise dinamik test (dynamic testing) olarak nitelendirilir.

### 2.6.5. Yazılım Doğrulama ve Geçerleme

Yazılım test ve kontrolü aşağıda açıklanan doğrulama ve geçerleme (verification and validation) (Tran 1999) ilişkili olarak gerçekleştirilir.

- **Doğrulama (Verification):** Yazılımı doğru olarak kurguladık mı? (Beklenen özelliklere uyuyor mu?) Yazılımdan beklenen özellikler mevcut mu?
- **Geçerleme (Validation):** Doğru bir yazılım mı kurguladık? (Kullanıcı gerçekten bunu mu istiyor?) Var olduğu değerlendirilen özellikler gerçekten beklenen işlev(ler)i gerçekleştiriyor mu?

### **2.6.6. Yazılım Test ve Kontrol Timi**

Yazılımlar, yazılım testçileri (*software testers*) tarafından test edilir. 1950'lere kadar yazılım testçisi terimi genellikle kullanılmıştır. Ancak, sonraları ayrı bir meslek olarak ortaya çıkmıştır. Yazılım test ve kontrolünde zaman aralıkları ve farklı hedeflere göre test yöneticisi/lideri, test tasarımcısı, test edici, test otomasyon geliştiricisi ve test icra yönetmeni gibi farklı roller tesis edilmiştir.

### **2.6.7. Yazılım Kalite Garantisi (SQA- Software Quality Assurance)**

Tartışmaya açık olduğu değerlendirilse bile, yazılım test ve kontrolü yazılım kalite garantisi sürecinin önemli bir bölümü gibi gözüktür. SQA'da, yazılım süreci uzmanları ve hesap kontrolörleri yazılım ve onun geliştirilmesine daha geniş bir bakış açısıyla bakarlar. Yazılım mühendislik sürecini hata oranını yükselten yanlışların miktarını azaltmak amacıyla kendi içinde inceler ve değiştirirler. Yazılımın tabiatına uygun şekilde kabul edilebilir bir hata oranı belirlerler. Bir uçağın uçuşunu simüle etmek üzere tasarlanmış mimari esaslı bir video oyununun, gerçek bir hava aracının işlevlerinin kontrol etmek üzere geliştirilmiş görev için kritik sayılan (mission critical) yazılımlara göre hatalara daha toleranslı olması normaldir. SQA ile arasında yakın bağlantılar bulunmasına rağmen test ve kontrol birimi bağımsız olarak görev yapar ve bazı şirketlerde SQA işlevi bulunmaya da bilir.

### **2.6.8. Test Metotları**

Yazılım test metotları geleneksel olarak kara kutu testi (black box testing) ve beyaz kutu testi (white box testing) olmak üzere ikiye ayrılır. Bu iki yaklaşım bir test mühendisinin test meselelerinin tasarımını yaparken sahip olduğu bakış açısının tanımlanmasında kullanılır.

#### **2.6.8.1 Kara Kutu Testi (Black box testing)**

Bu test, yazılımın iç işlevlerini anlamaya çalışmaz ve onu içinde ne olduğu bilinmeyen kara bir kutu gibi değerlendirir. İşlevselliğin ihtiyaçlara göre test edilmesini amaçlar (Laycock 1993). Bu nedenle, testçi verileri girer ve sadece test nesnelерinin çıktılarını görür. Bu test seviyesi, genellikle sağlanan bir girdi ve alınan çıktı değerinin test meselesinde belirlenen beklenen değerle aynı olup olmadığının değerlendirilmesi amacıyla bütün test meselerinin test edene verilmesini gerektirir. Eşitlik Bölümlendirmesi (equivalence partitioning), Sınır Değeri Analizi (boundary value analysis), tüm parçaların testi (all-pairs testing), Basitleştirerek Test Etme



(fuzz testing), Model Esaslı Test (model-based testing), Ayrıntılı Tanımlama Matrisi (traceability matrix) vb. kara kutu test metotlarından bazılarıdır.

### **2.6.8.2 Beyaz Kutu Testi (White box testing)**

Beyaz Kutu Testi, test edici iç veri yapılarına, kod ve algoritmalara ulaşabildiğinde yapılır. Çeşitleri;

Bütün Kodların Kullanımı (code coverage) – Bütün kodların kullanılmasını sağlayacak bazı kriterlerin yerine getirilmesi için testlerin oluşturulması faaliyetidir. Örneğin test tasarımcısı programdaki tüm cümleciklerin en az bir kez uygulanmasını sağlayacak testler geliştirebilir.

- Değişime Uğratarak Test Etme Metotları (mutation testing methods).
- Hatalı Ara Durum Verme Metotları (fault injection methods)
- Statik Test (static testing) – Beyaz kutu testi tüm statik testleri içerir.

### **2.6.8.3 Kod Bütünlüğü Değerlendirmesi**

Beyaz kutu test metotları, kara kutu test metodu ile oluşturulan test meselesinin tamam olup olmadığının değerlendirilmesinde kullanılabilir. Bu, yazılım timinin sistemin nadiren test edilen bir bölümünü incelemesine imkân tanır ve en önemli işlev noktalarının (function points) test edilmesini garanti eder.

Bütün kodların kullanımı aşağıda açıklanan iki yöntemle gerçekleştirilebilir ve her ikisi de yüzde olarak ölçülebilen bir tamamıyla kullanım ölçütü verir.

- İşlevlerin yerine getirildiği rapor edilerek “işlevlerin tamamının kullanımı (function coverage)”
- Testin tamamlanması için çalıştırılan satırların sayısı rapor edilerek cümleciklerin tamamının kullanımı (statement coverage),

### **2.6.8.4 Gri Kutu Testi (Grey Box Testing)**

Son yıllarda Gri Kutu Testi terimi ortak olarak kullanılmaya başlanmıştır. Bu metotla test meselelerinin tasarlanması maksadıyla iç veri yapıları ve algoritmalara ulaşılmasına imkân tanınarak testin kullanıcı tarafından veya kara kutu seviyesinde gerçekleştirilmesi sağlanmıştır.

Girdi verileri üzerinde oynanması veya belirli formatta çıktı alınmasının sağlanması faaliyetleri yazılım test edilirken girdi ve çıktı değerleri belirgin şekilde kara kutu dışında olduğundan gri kutu testi olarak değerlendirilmez. Bu özellik,

sadece ara yüzler teste tabi tutulduğunda farklı geliştiriciler tarafından yazılan kodların ait olduğu iki modül arasında entegrasyon testi (integration testing) yapıldığında önem arz eder. Gri kutu testi sınır değerlerinin belirlenmesi maksadıyla ters mühendislik (reverse engineering) yöntemini de içerir.

#### **2.6.8.5 İşlevsel Olmayan Yazılım Testi**

Yazılımın işlevsel olmayan özelliklerinin test edilmesi maksadıyla aşağıdaki özel metotlar kullanılır.

- Performans Testi (Performance testing) Yazılımın yüksek oranda verinin elleçlenip elleçlenemeyeceğini ya da çok sayıda kullanıcının aynı anda sistemi kullanıp kullanamayacağını kontrol eder
- Kullanılabilirlik Testi (Usability testing) kullanıcı ara yüzünün kolay anlaşılabilir olup olmadığını veya kullanıcı kolaylığı sağlanıp sağlanmadığını test eder.
- Güvenlik Testi (Security testing); bir sistemde gizlilik dereceli verilerin işlenebilmesi ve sisteme kontrol dışı girmelerin engellenmesinin gerektiği yazılımlar kapsamında gerçekleştirilir.
- Genelleştirme ve Yerelleştirme (internationalization and localization); yazılımın belirli bir lokal ağ veya internet bağlantılı sistemde çalışabilmesi hususunun test edilmesi esasına dayanır. Sahte yerelleştirme (pseudolocalization) metodu bu noktada kullanılabilir.

#### **2.6.9. Test Süreci**

Yazılım test ve kontrolünün yazılım daha kullanıcıya teslim edilmeden önce ve işlevler geliştirildikten sonra bağımsız bir test ekibi tarafından gerçekleştirilmesi sürecidir. Bu uygulama genellikle test safhasının proje erteleme maliyetlerinin karşılanması ve bu sayede teste ayrılan zamanın makul değerler içinde tutulmasında bir proje tamponu görevi yapar (Myers, 1979:145-146).

Başka bir uygulama yazılım testinin projenin başlandığı anda başlaması ve proje bitene kadar sürekli bir süreç olarak devam ettirilmesini öngörmektedir (Dustin, Elfriede 2002:3).

Öte yandan, “test esaslı yazılım geliştirme (test-driven software development)” modeline bağlı yoğun programlama (extreme programming) ve hızlı yazılım geliştirme (agile software development) gibi bazı disiplinler ortaya çıkmıştır.

Bu süreçte yazılım mühendislerince ilk olarak birim testleri (unit tests) yazılır (yoğun programlama metodolojisinde genellikle çift programlama (pair programming) yöntemiyle). Hiç şüphesiz, bu testler ilk başta beklendiği şekilde başarısızlıkla sonuçlanır. Müteakiben kod yazıldıkça, git gide büyüyen test meselelerinin kullanımına başlanır. Test meseleleri yeni başarısızlıklar ve olumsuz durumlar keşfedildikçe ve sürekli olarak güncellenir ve geliştirilen regresyon testleri ile entegre edilirler. Birim testleri, genellikle yapım sürecine entegre edilir ve geri kalan yazılım kaynak kodlarıyla uyumlu şekilde yapısında bulunan ve kısmen manuel olarak yapım kabul sürecinde uygulanacak etkileşimli testler kullanılarak muhafaza edilir.

Test ve kontrol aşağıdaki seviyelerde gerçekleştirilir.

- Birim Testleri (Unit testing), en küçük yazılım bileşen veya modülünü test eder. Yazılımın her bir birimi (temel bileşen), birim için belirlenen ayrıntılı tasarımın doğru şekilde uygulanmasının doğrulanması maksadıyla test edilir. Nesne tabanlı bir ortamda, en düşük sınıfta ve en küçük birim testlerinde yapıcılar ve yok ediciler bulunur (Binder 1999:45).

- Entegrasyon Testleri (Integration testing), entegre edilmiş bileşen veya modüller arasındaki ara yüz ve etkileşimlerdeki hataları ortaya çıkarır. Yazılım bir sistem olarak işleyebilir hale gelene dek, mimari tasarım elemanlarına ilişkin test edilmiş daha büyük bileşen grupları entegre ve test edilir (Beizer 1990:21,430).

- Sistem Testi (System testing) sistemin bütün ihtiyaçlarının karşılandığının doğrulanması maksadıyla tamamen entegre edilmiş bir sistemi test eder (IEEE 1990).

- Sistem Entegrasyon Testi (System integration testing), sistem ihtiyaçlarında tanımlanmış her hangi bir dış veya üçüncü taraf sisteme sistemin entegre edilip edilemediğinin doğrulanması için kullanılır.

Yazılımın son versiyonu kullanıma sunulmadan önce, ilaveten alfa ve beta testleri yapılır.

- Alfa testi, sistemin simüle edilmiş veya gerçek ortamda muhtemel kullanıcılar veya geliştirici tarafında görev yapan bağımsız bir test timi tarafından test edilmesini öngörür. Alfa testi, henüz vitrine çıkmamış yazılımlar için, Beta testi öncesi bir iç kabul testi gibi gerçekleştirilir.

- Beta testi, alfa testinden sonra yapılır. Yazılımın versiyonları (beta versiyonları olarak bilinir) programlama timi dışında kısıtlı bir kullanıcı kesiminin

kullanımına sunulur. Yazılım konuya hakim belli bir kullanıcı grubuna denenmesi, incelenmesi ve test edilmesi maksadıyla verilir ve müteakiben ortaya çıkması muhtemel küçük hataların varsa ayıklanması sağlanır. Bazen, beta versiyonları daha fazla geri besleme sağlanması ve gelecekteki kullanıcıların tasarım aşamasında öngörülemeyen görüş ve isteklerinden de faydalanılması sağlanır.

Son olarak, kabul testi (acceptance testing) uç kullanıcı, müşteri, vb. tarafından ürünün kabul edilip edilmeyeceğinin belirlenmesi maksadıyla gerçekleştirilebilir. Kabul testi, geliştirme sürecinin her hangi iki aşaması arasında el değmeden üretim sürecinin bir bölümü olarak uygulanabilir.

#### **2.6.10. Regresyon Testi**

Yazılımın işlevsel değişiklik veya hata giderme maksadıyla düzenlenmesinden sonra, yapılan değişikliklerin istemeden önceki yanlış bir işlevin tekrarına neden olup olmadığını garanti etmek maksadıyla, değiştirilen yazılım üzerinde daha önce uygulanmış testler bir regresyon (geriye dönük değerlendirme) testi (regression test) ile yinelenir. Regresyon testi yukarıda açıklanan test seviyelerinin birinde veya tamamında uygulanabilir. Bu testler genellikle otomatikçe bağlanır.

Regresyon testinin daha özel bir şekli olan “Makuliyet Testi (sanity testing)” sistemin garip bir davranışta bulunduğu durumların kontrol edilmesi ve “Duman Testi (smoke testing)” temel işlevselliklerin test edilmesinde kullanılır.

#### **2.6.11. Yanlışlıkların Önceden Tespiti**

Bir yanlış ne kadar erken farkedilirse, o kadar daha ucuza düzeltileceği ortak bir inanıştır (Bach ve Pettichord 2001). Aşağıdaki tablo tespit edildiği aşamaya bağlı olarak bir hatanın giderilme maliyetini göstermektedir (McConnell 2004). Örneğin; ihtiyaç olarak ortaya çıkan bir problem sadece son sürümde bulunuyorsa, bu ihtiyaç gözden geçirmesi esnasında ortaya çıktığı zamanki giderilme maliyetine göre 10 ila 100 kat daha fazla maliyet getirecektir. (Bknz. Tablo-E2.1)

**Tablo-E2.1: Aşamalara Göre Hataların Giderilme Maliyetleri**

| Sistem Geliştirme Aşamaları | Tespit Edildiği Aşama |                |                                |                |           |
|-----------------------------|-----------------------|----------------|--------------------------------|----------------|-----------|
|                             | İhtiyaç Analizi       | Mimari Tasarım | Yapım (Kodlama ve Entegrasyon) | Sistemin Testi | Son Sürüm |
| İhtiyaç Analizi             | 1                     | 3              | 5-10                           | 10             | 10-100    |
| Mimari Tasarım              | -                     | 1              | 10                             | 15             | 25-100    |
| Yapım                       | -                     | -              | 1                              | 10             | 10-25     |

(McConnell 2004)

### 2.6.12. Yazılım Testinin Ölçülmesi

Genellikle kalite; doğruluk (correctness), tamamiyet (completeness) ve güvenlik (security) gibi ana başlıklarla sınırlı olsa da, ISO 9126 standardı ile tanımlanan yapabilme yeteneği (capability), güvenilirlik (reliability), etkinlik (efficiency), taşınabilirlik (portability), bakım ve idame edilebilirlik (maintainability), uyumluluk (compatibility) ve kullanılabilirlik (usability) gibi daha teknik ihtiyaçları da içerebilir.

“Ölçme Kriteri (metrics)” adı verilen ve yapılan testin yeterliliğini veya yazılımın durumunu ölçmede kullanılan genel yazılım ölçütleri mevcuttur.

### 2.6.13. Test Olayı (Test Case)

Test olayı; olay, faaliyet, girdi, çıktı, beklenen sonuç ve elde edilen sonuç bilgilerini içeren bir test dokümanıdır. Klinik olarak tanımlanan bir test olayı girdi (sebebe) ve beklenen sonuçtan oluşur (IEEE 1998). Bazı test olaylarında girdi senaryo ve beklenen sonuç bakımından daha ayrıntılı tanımlama yapılmış olabileceği gibi, sebebe sonuç ilişkisi “X durumu için elde edilen sonuç Y’dir” cümlecisindeki gibi basitçe tanımlanmış da olabilir. Duruma göre değişebilmekle birlikte, sadece bir beklenen sonuç veya beklenen çıktıya sahip olan adımlar (ancak bu adımlar genellikle ekonomik bir sorun şeklinde çok sayıda test olayı karşısında denenen bağımsız bir test usulü içinde yer alabilirler) serisi de olabilir. Opsiyonel veri alanları; “olay kodu, test adımı veya icra numarası, ilişkili ihtiyaç(lar), derinlik, test kategorisi, yazarı ve testin otomatikçe bağlanabilir olup olmadığını ve

otomasyona bağlandığını gösterir seçim düğmeleri” şeklinde olabilir. Daha büyük test olayları, önceden gerekli olan durumları veya adımları ve tanımları içerebilir. Bir test olayı, gerçek sonuç için bir yer de tanımlamalıdır. Bu adımlar, bir kelime işlem dokümanında boş sayfada, veri tabanında veya diğer genel bir depolama alanında muhafaza edilebilir. Bir veri tabanı sisteminde, geçmiş testlerin sonuçları, sonuçları kimin aldığı ve bahse konu sonuçları oluşturmada kullanılan sistem konfigürasyonları da görülebilir. Geçmiş bu sonuçlar genellikle ayrı bir tabloda gösterilir.

#### **2.6.14. Test Yazıtları (Test Scripts)**

Test yazıtları, test olayı, test usulleri ve test verilerinden oluşur. Test yazıtları terimi ilk olarak otomatik regresyon test araçları tarafından oluşturulan iş ürünlerinden çıkarılmıştır. Günümüzde ise; bu araçlar manuel, otomatik veya ikisinin bir kombinasyonu şeklinde olabilmektedir.

#### **2.6.15. Test Meselesi (Test Suite)**

Test meselesi terimi, bir araya getirilmiş test olaylarına verilen genel bir tabirdir. Test meselesi, genellikle her bir test olayları seti için daha detaylı talimatlar ve hedefler de içerir. Test edicinin test esnasında kullanılan sistem konfigürasyonunu tanımladığı olmazsa olmaz bir bölümü de vardır. Bir test olayları grubu içinde, önceden gerekli olan durumlar veya adımlar ve aşağıdaki testlerin tanımlarını da bulundurabilir.

#### **2.6.16. Test Planı ve Test Kontrol (Test Harness) Elemanları**

Geliştiricilere, hangi test planının uygulanacağı kullanıcılardan tarafından bildirildiğinden, kod geliştirirken daha çok temkinli olurlar. Bu, aynı zamanda geliştirilen kodların her hangi bir sürpriz test olayına veya test planına maruz kalmayacağını da garanti eder.

Yazılım araçları, veri girdi ve çıktı örnekleri ve konfigürasyonlar bir bütün olarak test kontrolü elemanları olarak adlandırılır.

#### **2.6.17. Test Döngüsü Genel Esasları**

Kuruluşlar arasında farklı uygulamaları olmasına rağmen, test için tipik aşağıdaki esasları içeren bir döngü mevcuttur.

- **İhtiyaç Analizi:** Test, yazılım geliştirme ömür devrinin ihtiyaçların belirlenmesi aşamasında başlamalıdır. Tasarım safhasında, testçiler bir tasarımın

hangi unsurlarının test edilebilir olduğunu ve testlerin hangi parametrelere göre çalışması gerektiğini belirlemede geliştiriciler ile birlikte çalışırlar.

- **Test Planlaması:** Test stratejisi, test planı ve test zemininin ortaya konmasıdır. Test esnasında çok sayıda faaliyet yürütüldüğünden ve her faaliyet de farklı birimler arasında koordinasyon gerektirdiğinden her teste ait bir plan bulunması gereklidir.

- **Test Geliştirme:** Yazılımın test edilebilmesi için gerekli test usulleri, test senaryoları, test olayları ve test yazıtlarının geliştirildiği faaliyetler bütünüdür.

- **Testin İcrası:** Testçilerin, yazılımı test planı ve test olaylarını kullanarak sistemi test ettiği ve bulunduğu hataları geliştiricilere rapor ettiği aşamadır.

- **Test Raporlama:** Testin tamamlanmasını müteakip, testçiler ölçme kriterleri (metrics) belirler, kendi test faaliyetlerine (test effort) yönelik olarak son raporlarını hazırlar ve test edilen yazılımın kullanıma sunulmaya hazır olup olmadığını ortaya koyarlar.

- **Test Sonuç Analizi:** Hata analizi olarak da bilinen bu analiz, geliştirici tim tarafından kullanıcı ile birlikte hangi hataların elleçleneceğinin, hangilerinin giderileceğinin, hangilerinin reddedileceğinin (Örn.: yazılımın hata yapmadığı anlaşıldığında) veya hangilerinin daha sonra ilgilenilmek üzere başka bir zamana öteleneyeceğinin belirlenmesi amacıyla gerçekleştirilir.

- **Giderilen Hataların Yeniden Test Edilmesi:** Geliştirme timi tarafından bir hata giderildikten sonra test timi tarafından tekrar test edilir.

- **Regresyon Testi:** en son sürümün her hangi bir şeye zarara vermediğinin ve yazılım ürününün bir bütün olarak doğru çalıştığına garanti edilmesi amacıyla, yeni hazırlanmış, üzerinde değişiklik yapılmış veya hataları ayıklanmış yazılımın her bir entegrasyonu için, alt testlerden oluşan küçük bir test programına sahip olunması olağandır.

#### **2.6.18. Test ve Sertifikasyon**

Yazılım test ediciler ve kalite sağlama uzmanlarının profesyonel isteklerinin desteklenmesi amacıyla geliştirilmiş sertifikasyon programları mevcuttur. Hali hazırda sunulan hiç bir sertifikasyon gerçekte yazılımın test edilebilirliğini gösterecek uygulayıcılara ihtiyaç duymaz. Sertifikasyon konusunda genel kabul görmüş bir uygulama bulunmamaktadır. Bu bazılarının test alanının sertifikasyona hazır

olmadığını bildirmelerine neden olabilir. Sertifikasyon tek başına bir bireyin verimliliğini, yeteneklerini veya pratik bilgisini ölçemez ve bir test edici unsur olarak yeterli veya profesyonel olup olmadıklarını da belirleyemez.

#### 2.6.19. Yazılım Test Sertifikasyon Çeşitleri

Sertifikasyon sınav esaslı ve eğitim esaslı olmak üzere iki gruba ayrılabilir.

- **Sınav Esaslı Sertifikasyonlar:** kendi kendine çalışarak öğrenilebileceği gibi, bunlar için bir sınavı geçme ihtiyacı da olabilir. Örn.: Uluslararası Yazılım Test Kalifikasyon Kurulu (ISTQB, International Software Testing Qualification Board) veya Kalite Garantileme Kurumu (QAI, Quality Assurance Institute) için böyle bir zorunluluk mevcuttur.

- **Eğitim Esaslı Sertifikasyonlar:** Bu sertifikasyon grubu eğitimcilerce yönlendirilen ve her bir dersten geçme zorunluluğu bulunan seanslardır. Örn.: Uluslararası Yazılım Test Enstitüsü (IIST, International Institute for Software Testing).

#### 2.6.20. Sertifikasyonların Test ve Kontrolü

- Kalite Garantileme Kurumu (QAI, Quality Assurance Institute) tarafından hazırlanan Sertifikalı Yazılım Test Edici (CSTE, Certified Software Tester).

- Uluslararası Yazılım Test Enstitüsü (IIST, International Institute for Software Testing) tarafından hazırlanan Sertifikalı Profesyonel Yazılım Testi (CSTP, Certified Software Test Professional).

- *K. J. Ross & Associates* tarafından hazırlanan Sertifikalı Profesyonel Yazılım Testi (CSTP, Certified Software Test Professional) (Avustralya Versiyonu).

- Uluslararası Yazılım Test Enstitüsü (IIST, International Institute for Software Testing) tarafından hazırlanan CATe.

- Bilgi Sistemleri Sınavları Kurulu (ISEB, Information Systems Examinations Board) tarafından hazırlanan ISEB.

- Uluslararası Yazılım Test Kalifikasyon Kurulu (ISTQB, International Software Testing Qualification Board) tarafından hazırlanan Kurum Sertifikası.



- Uluslararası Yazılım Test Kalifikasyon Kurulu (ISTQB, International Software Testing Qualification Board) tarafından hazırlanan İleri Seviyede Sertifikalı Test Edici (CTFL, Certified Tester, Advanced Level).

- Brezilya Yazılım Test Sertifikasyon Kurumu (ALATS, *Brazilian Certification of Software Testing*) tarafından hazırlanan CBTS.

Kalite Garantileme Sertifikasyonları

- Amerikan Kalite Kurumu (ASQ, American Society for Quality) tarafından hazırlanan CSQE.

- Kalite Garantileme Kurumu (QAI, Quality Assurance Institute) tarafından hazırlanan CSQA.

## **2.7. Kullanıma Sunma (Deployment)**

Bu aşama test ve muayenesi uygun şekilde yapılmış yazılımın, piyasaya sürümü/satılması veya diğer üretim birimlerine dağıtım konularıyla ilgilidir.

Yazılımın kullanıma sunulma süreci, aralarında muhtemel geçişleri öngören birbirleriyle ilişkili bir çok faaliyetten oluşur. Bu faaliyetler, üretici, kullanıcı veya her ikisinin tarafında da ortaya çıkabilir. Her yazılım sistemi birbirinden farklı olduğundan, her bir faaliyet içindeki kısa süreç veya usullerin tanımlanması neredeyse imkansızdır. Bu nedenle, kullanıma sunma özel ihtiyaç ve niteliklere göre özelleştirilmiş genel bir süreç olarak yorumlanmalıdır.

### **2.7.1. Kullanıma Sunma Faaliyetleri**

#### **2.7.1.1 Sürüm (Release)**

Sürüm faaliyeti, geliştirme sürecinin tamamlanmasını müteakip gerçekleştirilir. Sistemin birleştirilmesi (assembly) ve kullanıcıya verilmesine ilişkin tüm işlemleri kapsar. Bu nedenle, yazılımın çalıştırılmasında kullanıcı tarafında ihtiyaç duyulabilecek kaynakları belirlemeli ve kullanıma sunma süreci alt faaliyetlerinin gerçekleştirilebilmesi için bilgi toplamalıdır.

#### **2.7.1.2 Kurulum**

Yazılımın kullanım öncesi kullanıcı bilgisayarlarına ve kullanıcı yerel veya özel ağına kurulmasına ilişkin faaliyetlerdir.

### **2.7.1.3 Faaliyete Geçirme (Aktivasyon)**

Faaliyete geçirme, yazılımın işletilebilir bileşeninin başlatılması faaliyetidir. Bu işlem basit bir sistemde icra için bazı komuta yapıları tesis ederken, karmaşık sistemlerde tüm destek sistemlerini kullanıma hazır duruma getirir.

Daha büyük yazılım kullanıma sunulduğunda, üretim ortamındaki bir üretim sunumcusu üzerine yazılımın çalışan bir kopyası kurulmuş olmalıdır. Kullanıma sunulan yazılımın diğer versiyonları test ortamında, geliştirme ortamında ve büyük hata düzeltim ortamında tesis edilebilir.

### **2.7.1.4 Faaliyetini Durdurma**

Faaliyet Durdurma, faaliyete geçirme işleminin tersi bir işlemdir ve yazılımın her hangi bir icra bileşeninin kapatılması işlemini yürütür. Herhangi bir güncelleme yapılmadan önce sistem faaliyetlerinin durdurulmasına ihtiyaç duyulabilmesi örneğinde olduğu gibi, genellikle diğer kullanıma sunma faaliyetlerinin gerçekleştirilmesi için faaliyet durdurma işlemi uygulanır.

### **2.7.1.5 Uyum Sağlama**

Uyum Sağlama (adaptation) işlemi daha önce kurulmuş bir yazılım sisteminin düzenlenmesi/değiştirilmesi için gerçekleştirilen bir süreçtir. Güncelleme ile uyum sağlama arasındaki temel farklılık güncellemenin çoğunlukla uzaktaki bir yazılım geliştirici tarafından ve uyum sağlama faaliyetinin ise kullanıcı tarafında ortam (environment) değişikliği gibi yerel olaylar tarafından başlatılmasıdır.

### **2.7.1.6 Güncelleme**

Güncelleme, bir yazılım sisteminin daha yeni bir sürümünün yazılımın kurulu eski versiyonunun tamamının veya bir kısmının yerini alma sürecidir.

### **2.7.1.7 Gömülü Alt Sistem Kullanımı (Built-in)**

Bazı yazılım sistemleri içinde güncellemelerin sistem kapatılmadan gerçekleştirilebilmesini sağlayan iç mekanizmalar oluşturulmuştur. Bu güncelleme süreçleri, otomasyonu tamamen otomatikten, kullanıcı tarafından başlatılan ve kontrol edilenlere kadar değişiklik gösterir. Bazı antivirüs yazılımları hem virüs tanımlarına ve hem de sistemin diğer bileşenlerine ilişkin güncellemelerin indirilmesi ve kurulması için yarı otomatik bir metot kullanır. Bazıları ise yeni güncellemelerin kullanıma sunulup sunulmadığının belirlenebilmesi için sorgulama mekanizmaları sunarlar.

### **2.7.1.8 Versiyon Takibi**

Versiyon takip sistemleri, PC'lerde veya yerel ağlarda kurulu yazılım sistemlerinin güncellemelerinin kullanıcı tarafından bulunması ve kurulmasına yardım eder.

Web tabanlı versiyon takip sistemleri yerel bir ağ üzerinde kurulu yazılım sistemi için yeni güncellemeleri takip eder/bulur ve kullanıcıyı bilgilendirir. Bazı versiyon takip sistemleri önce kullanıcının kendi bilgisayarında kurulu versiyonu araştırır ve müteakiben güncellemelerinin bulunup bulunmadığını internet üzerinden kontrol eder.

Yerel versiyon takip sistemi, yerel bir ağ üzerinde kurulu yazılım sistemleri için yeni güncellemeler ana sunumcуда mevcut olduğunda kullanıcıyı bilgilendirir. Bazı yerel versiyon takip sistemleri yerel bir ağda kurulu her bir yazılım paketi için versiyon ve diğer bilgileri depolar. Tek bir düğmeye basılarak uygulama güncellemesinin bulunduğu pencere açılır ve şifreli siteler için kullanıcı bilgilerini (kullanıcı adı ve şifresini) otomatik olarak (istendiğinde) doldurulur. Bu sayede ulaşılan güncelleme dosyası ile yazılım güncellenir.

Browser tabanlı versiyon takip sistemi yerel bir ağda kurulu yazılımlara ilişkin yeni yazılım güncelleme paketleri ağ üzerinde mevcut olduğunda kullanıcıyı uyarmak üzere geliştirilmiştir.

### **2.7.1.9 Yazılımın Kaldırılması (Uninstallation)**

Yazılımın kaldırılması, kurulumun tersi bir işlev olup, ihtiyaç duyulmayan bir yazılımın veya onun bir bölümünün kullanımdan kaldırılmasının esas alır. Aynı zamanda diğer yazılım sistemlerinin yeniden konfigürasyonu ile kaldırılan yazılımın ortamda bıraktığı izlerin (dosya, vb.) tamamen temizlenmesi de sağlanabilir.

### **2.7.1.10 Kullanım Dışı Kalma**

Son olarak, yazılım ürünü ömür devrinin (life cycle of a software product) sonu olarak da nitelendirilen sistemin kullanımının gereksiz (obsolete) olduğunun tespiti ve üretici desteğinin durdurulmasıyla süreç sonlandırılır.

## **2.8. Kopyalama, Sevk ve Yükleme Aşaması**

Yazılımın gerekli kalite değerlendirmenin yapılmasından sonra donanımdaki seri üretime benzer olarak kopyalarının çıkarılması gerekmektedir. Yazılımın kopyalanması yani aynı konfigürasyonda olan birçok yazılımın üretilmesi donanım

gibi karmaşık ve uzun süre alacak süreçlerden geçerek değil basit ve kısa süre alan bir süreç sonucu gerçekleştirilebilmektedir. Bu süreç işletim sisteminin kopyalama için kullanılan komutuyla gerçekleştirilebilir. Kopyalanan yazılımın kopyalandığı ortamında ayırt edilebilirliği ve izlenebilirliği sağlanmalıdır. Bu durum eğer kopyalanan ortam disket, disk, manyetik şerit gibi harici bir ortam ise bu disketin veya diskin üzerine yazılımın konfigürasyon numarası revizyonu yazılabilir. Kopyalanan yazılımın doğru olarak kopyalanıp kopyalanmadığı araştırılmalı ve bu faaliyete ilişkin kayıtlar tutulmalıdır.

Kopyalanan yazılımın bulunduğu taşınabilir ortamın yükleneceği ortama taşınması esnasında yazılıma dolayısıyla yüklü olduğu ortama herhangi bir zarar gelmemesi için taşınma koşulları dikkate alınarak gerekli önlemler alınmalı ve uygulanmalıdır.

Yazılımın hedef ortama taşınmasını müteakip bu yazılımın hedef ortama yüklenmesi ve işletilmesi gerekecektir. Hazırlanacak bir yazılım yükleme planı ile yükleme işlemindeki yetki ve sorumluluklar, yükleme sürecinin adımları, yüklenecek ortamın özellikleri verilmelidir. Yükleme işlemi bu plana göre gerçekleştirildikten sonra yükleme işleminin doğru olarak gerçekleştirilip gerçekleştirilmediği yükleme planında verilen kabul ve ret kriterleri göz önüne alınarak araştırılmalı ve bu incelemenin sonuçları kayıt altına alınmalıdır. Sürecin gerçekleştirilmesi esnasında çıkabilecek problemler gerekli onay mekanizmalarından geçerek çözülmelidir.

## **2.9. Kullanım ve İdame Aşaması**

Bu aşamada ortaya çıkabilecek sorunlar ve çözüm önerileri aşağıda verilmiştir.

- Sistemin (yazılımın) kullanımı esnasında çıkabilecek kısıtlamalar. Bu kısıtlamalara örnek olarak kullanım ihtiyaçlarından, hedef donanım karakteristiğinden, kullanıcıdan kaynaklanan kısıtlamalar verilebilir.
- Kullanım esnasında çıkabilecek problemler verilebilir. Bu problemler yazılım veya donanımdan kaynaklanan ve sistemin tamamen veya bir kısmının kullanımını engelleyecek sorunlar olabilir.
- Tedarikçi şikayetleri olabilir.

Bu sorunların çözümlenebilmesi amacıyla ;

- Öncelikle tespit edilen veya bildirilen problem veya öneri yazılı hale getirilmelidir. Yazılı hale getirilmiş olan problem veya önerinin kalite planı veya prosedürlerde belirtilen yetkili ve sorumlu personel tarafından analiz edilmesi/değerlendirilmesi gerekmektedir.

- Yapılan analiz sonucunda oluşturulan analiz raporu yetkili personel tarafından gözden geçirildikten sonra (tedarikçinin de bu süreçte söz sahibi olması beklenmektedir) çözüm önerileri oluşturulmalı ve yetkili birimlerin onayına sunulmalıdır. Bu çözüm önerisi uygulanmadan önce, sistemin güvenliği, güvenilirliği gibi temel ihtiyaçlarını ve yazılımın (sistemin) diğer birimlerinin performansını olumsuz yönde etkilemeyeceği veya karşılaşılabilecek olumsuzlukların belirlenerek kullanıcı ve tedarikçinin bu konuda mutabakata varmış olması, gerekmektedir.

- Yetkili birimler ve tedarikçi tarafından da onaylanan çözüm önerisi oluşturulacak uygulama planı ile uygulamaya geçirilmelidir. Bu çözüm önerisi yeni yazılım/donanımın birim veya öğelerinin oluşturulması veya mevcut yazılım/donanım öğelerinde değişiklik yapılmasını gerektirebilir.

- Kabul edilen çözüm önerisi uygulandıktan sonra, bu çözümün hedefleri sağlayıp sağlamadığı araştırılmalı gerekli test ve değerlendirme faaliyetleri icra edilmeli ve sonucun olumlu olması durumunda gerekli dokümanlar (operasyon kitapları, bakım idame kitapları, kullanıcı el kitapları) değiştirilmelidir.

## **2.10. Proje Dosyası ve Yazılım Dokümantasyonu**

Yazılımın, iç tasarımının gelecekteki bakım ve geliştirme faaliyetlerinin gerçekleştirilebilmesi amacıyla yazılı hale getirilmesi (dokümantasyonu) geliştirme süreci boyunca gerçekleştirilir.

Proje dosyası, proje ile ilgili belgeleri düzenli biçimde tutmak için oluşturulur. Böylelikle, proje ile ilgili temel bilgiler tek bir yerde ve erişilebilir durumda olur. Bu dosya tek bir kişi tarafından oluşturulmalı, ancak, proje ile ilgili herkesin kullanımına açık olmalıdır. Böylelikle bilgilerin bütünlüğü sağlanır. Olası bir proje yöneticisi değişiminde ise proje dosyası yeni yöneticinin en önemli başvuru kaynağı olacaktır.

### **2.10.1. Dokümantasyon**

Proje yöneticisi, projenin yönetimi, izlenmesi ve kontrolü sürecinde bazı belgelerden yararlanır. Önceden belirlenmiş çalışma belgeleri, projenin akışı içinde yaralan yazışmalar ve özel raporlar burada yer alabilir.

Projelerin yürütülmesi sürecinde proje yöneticisi ve ekibi için önem taşıyan belgeler aşağıda ana başlıklar altında verilmiştir:

- **Proje Başlatma Belgesi**, projenin planlanmasına, bütçelenmesine ve proje planının hazırlanmasına temel oluşturur. Bu belgenin projeye ilişkin ilk resmi belge olma özelliğinden dolayı proje dosyasında bulunması önem taşımaktadır.
- **Sözleşme**, proje başlangıç belgelerinin yasal başlangıç ve referans belgesidir.
- **Proje Planı**, proje kapsamında yapılacak işlerin ayrıntılı hale getirilmesini, zamanlamasını, sıralamasını ve kaynakların tanımlanmasını gösterir. Proje planı içeriğinde;
  - **"Sınıflandırılmış İş Listesi" (WBS)** Nelerin yapılacağına ilişkin tüm işlerin listesini oluşturan bu yapının asli dosyaya konmalıdır. Daha fazla ayrıntı için nelere başvurulması gerektiği belirtilmelidir.
  - **Proje Zamanlaması**; Gantt, PERT çizgeleri ya da ikisi birden kullanılabilir.
  - **Proje Bütçesi**, tümüyle ya da "sınıflandırılmış iş listesi" ile birlikte özet olarak verilebilir.
  - **Risk Yönetim Planı**, projenin asil uygulama planından ayrı olarak proje ekibinden ya da ilgili diğer kişilerden gelen riskler ile ilgili raporlar konur. Bu plana ek olarak, yönetimden gelen ilgili raporlar da kullanılabilir.
  - **Sistem tanımı**, sistem ihtiyaçları analizi aşamasında hazırlanan, proje kapsamını, kullanıcı ihtiyaçlarını ve önerilen sisteme ilişkin ön tasarım bilgilerini içeren bir rehber belge niteliğindedir.
  - **Sistem Tasarımı**, geliştirilen sisteme ilişkin işlem ve veri tanımlarını, tasarım ve işlem prosedürlerini, detaylı sına planlarını içeren belgedir.
  - **Sistem Teslim Özellikleri**, teslim edilecek sistemin özellikleri, projenin daha sonraki gerçekleştirme, kurma ve sına aktivitelerinin oluşturulmasına ilişkin tüm detay bilgileri ve tasarım modelini içermektedir.
  - **Yazışmalar**, proje yöneticisinin proje ile ilgili yaptığı tüm yazışmalar dosyada bulundurulmalıdır. Bu tür yazışmalar, hukuksal önem taşıdığından düzenli bir biçimde bulundurulmaları önemlidir.

Ayrıca, yukarıda belirtilen belgelerin yanı sıra

- Konfigürasyon Yönetimi
- Sınama ve Kabul Raporları
- Toplantı Notları
- Fatura Takibi
- Proje Özeti
- Destekleyen Diğer Planlar proje dosyasında yer almalıdır.

### 2.10.2. Raporlama

Proje ile ilgili bilgi akışı, projenin hem yönetimini, hem de denetimini etkileyecektir. Doğal olarak, yönetimin her durumda güncel ve sağlıklı bilgi akışına ihtiyacı vardır. Ancak bu ihtiyaç, proje söz konusu olduğunda, daha büyük bir önem kazanmaktadır. Bu kapsamda, raporlama, temel bir yönetim aracı işlevini görür. Raporlar, proje sürecinin her aşamasını kayda geçirmek ve belgelemekle kalmayarak, uygulama sırasında taraflar arasındaki iletişimi güçlendirir, uygulamanın izlenmesine ve denetlenmesine yardımcı olur. Ayrıca, projenin geliştirilmesi sırasında önceden öngörülmesi mümkün olmayan koşulların ortaya çıkıp çıkmadığının belirlenmesini sağlar.

Proje yönetiminin en önemli faaliyetlerinden birisi, projenin başlangıçta belirlenen plana uygun olarak gidip gitmediğinin izlenmesidir. Proje yöneticisine bu konuda yardımcı olacak en önemli araç, düzenli bir raporlama sistemi çerçevesinde, belirlenen nitelikte, çeşitli sıklıkta ve içerikte sunulan raporlardır. En yaygın kullanılan raporlar şunlardır:

- **Durum Raporu**, genellikle proje planında öngörülen belirli bir aşamanın tamamlanmasından sonra hazırlanır. Projenin güncel durumu hakkında detaylı bilgi sağlar.

- **Gelişme Raporu**, projenin yakından izlenmesini sağlar. Durum Raporu 'dan dan daha sık aralıklarla ve düzenli olarak hazırlanır. Projenin, proje planına uygun olarak ilerleyip ilerlemediğini, sorunları, projede ortaya çıkabilecek değişiklikleri ve diğer önemli noktaları gösterir. Bu raporun haftalık olarak hazırlanması projenin izlenmesi ve kontrolü açısından daha uygundur.

- **Özel Amaçlı Raporlar**, belirli bir soruna ya da konuya dikkati çekmek ve zamanında önlem alınmasını sağlamak amacıyla, gerek görüldükçe hazırlanır. Bu tür raporlar, Durum ve Gelimse raporları gibi düzenli bir biçimde hazırlanmadığından, bunların gerek biçimi ve gerekse içeriği, raporun hazırlanma amacına göre, farklılık gösterir.

- **Toplantı Tutanaqları**, yapılan toplantıların özetini, alınan kararları, dile getirilen sorunları ve önerilen çözümleri kayda geçirerek, bilgi akısını ve bilginin yaygın hale getirilmesini sağlar. Daha sonraki uygulamalar için bir başvuru kaynağı oluşturur.

### 2.11. Yazılım Eğitim Desteği

Yazılım projelerinin büyük çoğunluğunun başarısızlığa uğramasının altında, yüklenicilerin kullanıcının aslında kullanmayacağı abartılı ve gereksiz özellikler içeren ve arzuladığı üründen farklı yapıda bir yazılım için geliştirme timinin harcadığı zamanın ve yaptığı planlama miktarının önemsiz olduğunu anlayamamaları yatmaktadır. İnsanlar genellikle değişikliklere karşı direnç gösterirler ve bilmedikleri bir alanda çalışmak istemezler. Kullanıma sunma aşamasında da durum böyledir. Bu nedenle hazırlanan yeni yazılıma yönelik eğitim verilmesi ve kullanıcıların her seviyede eğitimi büyük önem arz etmektedir. Eğitim Planı dokümanı EK-14'de sunulmuştur.

### 2.12. Bakım

Yeni ortaya çıkan problemlerin giderilmesi için ya sistemin bakım ve iyileştirmeleri yapılacak ya da yeni ihtiyaçların karşılanması için yazılımın ilk geliştirme aşamasından daha fazla zaman harcanması gerekecektir. Öngörülemeyen bir problemin giderilmesi için orjinal kod tasarımının içine uymayan bir kodun ithali veya kullanıcının istediği ilave işlevlerin gerçekleştirecek bir kodun eklenmesi gerekebilir. Bakım aşamasında aslında sadece öngörülemeyen eksikliklerin giderilmesine çalışılmaz, aynı zamanda sistemin herhangi bir nedenle çökmesi, yeniden yüklenme ihtiyacının ortaya çıkması, sistemin kendi yarattığı kayıt dosyalarının yer yer temizlenmesi, verilerin gözden geçirilmesi, sistem işlevlerini hızlandırıcı tedbirlerin alınması vb. gibi kabul sonrası faaliyetlerini içerir.

Yazılım mühendisliğinde, yazılım bakım ve idamesi kullanıcıya teslimi müteakip yazılım ürünüde ortaya çıkabilecek yanlışlıkların düzeltilmesi, performansın artırılması, diğer özelliklerin iyileştirilmesi veyadeğişiklik yapılmış bir



ortama ürünün uyum sağlaması için yazılım üzerinde değişiklik/düzenleme yapılması sürecidir. Uluslararası ISO/IEC 14764 standardı aşağıda sunulan altı yazılım bakım ve idame sürecini tanımlar.

- Uygulama süreci; bakım planının tasarlanması ve oluşturulması, geliştirme sırasında ve müteakiben ürün konfigürasyon yönetiminde ortaya çıkan problemlerin çözümlenmesi için hazırlık yapılması gibi yazılım hazırlama ve geçiş faaliyetlerini içerir.

- Problem ve değişiklik analiz süreci; uygulama bakım ve idame grubunun sorumluluğuna bırakıldığında icra edilir. Bakım ve idame programcısı her bir isteği analiz etmeli, tekrar uygulayıp doğrularak geçerliliğini kontrol etmeli, inceleyerek çözüm önerisi sunmalı, isteği ve çözüm önerisini yazılı hale getirmeli ve son olarak da değişiklikleri uygulayabilmek için ihtiyaç duyacağı yetkileri almalıdır.

- Süreç değişikliği kendi kendine uygulayabilecek şekilde düzenlenmelidir.

- Süreç değişikliklerinin kabulü, yapılacak değişikliğin gerçekten bir çözüm getirip getirmediğinden emin olunması maksadıyla isteği ileten kullanıcı ile birlikte kontrol edilmesini müteakip gerçekleştirilir.

- Göç süreci (migration process) özel bir durumdur ve günlük bakımın bir parçası değildir. Bu süreç, yazılımın işlevlerinde her hangi bir değişiklik yapılmadan başka bir platforma taşınması gerektiğinde kullanılabilir ve bir bakım proje timi bu görevi yerine getirmekle görevlendirilebilir.

- Son süreç, son bakım süreci günlük bazda gerçekleştirilmeyen yazılımın bir kısmının kullanımdan kaldırılmasıdır.

Aşağıda örnek olarak sunulan süreç, faaliyet ve uygulamalar bakım ve idame faaliyetini yürütenler için vazgeçilmezdir.

- **Geçiş (Transition):** Bir sistemin, geliştiriciden bakım ve idameciye aktarılması esnasında kontrol ve koordine edilen birbirini takip eden faaliyetlerdir.

- Ana odağa özel (domain-specific) bakım ve idame sözleşmeleri ve Sözleşme Hizmet Seviyelerinin (SLAs-Service Level Agreements) bakım ve idameciler tarafından gerçekleştirilmesi,

- **Değişiklik İstek ve Problem Raporlama Yardım Masası (Modification Request and Problem Report Help Desk):** Bakım ve idameciler

tarafından, aldıkları deęişiklik isteklerinin önceliklendirilmesi, yazılı hale getirilmesi ve yönlendirilmesinde (kime havale edileceğinin belirlenmesi) kullanılan bir problem çözme sürecidir.

- **Deęişiklik İsteęi Kabulü/Reddi (Modification Request Acceptance/Rejection):** deęişiklik isteklerinin büyüklük, çaba ve karmaşıklık açısından bakım ve idamecilerce üzerinde çalışılarak reddedilebilir ve geliştiricilere geri gönderilebilir.

Ortak bir bakım ve idame algısı, bakım ve idamenin sadece sistemdeki küçük hataların giderilmesi için yapıldığı şeklindedir. Ancak yıllarca süren araştırma ve uygulamalar, % 80'in üzerinde bir oranda bakım ve idame faaliyetlerinin hata düzeltme dışındaki olaylarla ilgilendiğini ortaya koymuştur (Pigosky, 1996). Bu algı, gerçekte birer sistem işlev geliştirme/iyileştirme olan problem raporlarını ileten kullanıcılarca sürekli hale getirilmesi yadrganmayacak kadar çok karşılaşılan bir durumdur.

Yazılım bakım ve idamesi ve sistemin evrimi ilk defa 1969 yılında Lehman tarafından ortaya atılmıştır. Yirmi yıldan uzun süren çalışmaları "Evrimin Sekiz Kanununun (eight Laws of Evolution) formülasyonu ile sonuçlanmıştır. Bakım ve idamenin aslında evrimsel geliştirmeler olduğu ve bakım ve idame kararlarının zaman geçtikçe sisteme (yazılıma) neler olduğunun anlaşılmasına yardım ettiğine ilişkin elde ettiği sonuçlar, araştırmalarının anahtar bulgularını oluşturur. Lehman, yazılım sistemlerinin zamanla evrim geçirdiğini ortaya koymuştur. Karmaşıklığın giderilmesi için önceden önlem alınmadığı takdirde, evrimleştikçe daha karmaşık bir yapıya bürünürler.

E.B. Swanson ilk başlarda, yazılım bakım ve idamesini düzeltici, uyarlayıcı ve kusursuzlaştırıcı olmak üzere üç kategoriye ayırmıştır. Günümüzde ise yapılan güncellemeler sonucunda oluşturulan ISO/IEC 14764 standardı ile bu kategoriler aşağıda açıklandığı şekilde dörde çıkarılmıştır.

- **Düzeltilici Bakım ve İdame:** Yazılımın kullanıcıya tesliminden sonra tespit edilen problemlerin anında düzeltilerek sistemde deęişiklik yapılması esasına dayanır.

- **Uyarlayıcı Bakım ve İdame:** Yazılımın kullanıcıya tesliminden sonra, deęiştirilmiş veya deęişen bir ortamda bir yazılımın kullanılabilir olarak idamesi için sistemde deęişiklik yapılması esasına dayanır

- **Kusursuzlaştırıcı Bakım ve İdame:** Yazılımın kullanıcıya tesliminden sonra, yazılım sisteminin performans ve idame yeteneğinin iyileştirilmesi için sistemde değişiklik yapılması esasına dayanır.

- **Önleyici Bakım ve İdame:** Yazılımın kullanıcıya tesliminden sonra, etkin hatalar durumuna gelmeden önce yazılımda gözükmeyen hataların tespit edilmesi ve düzeltilmesi maksadıyla sistemde değişiklik yapılması esasına dayanır.

Yazılım bakım ve idamesine ilişkin anahtar esaslar hem teknik hem de yönetsel olabilir. Anahtar yönetim esasları: kullanıcı önceliklerine göre düzenleme yapılması, personel görevlendirilmesi, bakım ve idamenin nasıl bir organizasyonla gerçekleştirileceğinin belirlenmesi ve maliyetlerin tespiti. Anahtar teknik esaslar ise; anlamaların sınırlandırılması (yazılımdan beklenen hususların somutlaştırılması ve dondurulması), etki analizi yapılması, test ve değerlendirmeye yapılması ve bakım ve idamenin ölçülebilmesinin sağlanması (bakım ve idame için ne ölçüde bir organizasyon kullanılması ve faaliyet sahalarının ayrıntılarıyla belirlenmesi).

Yazılım bakım ve idamecilerinin izleyeceği vazgeçilmez süreçleri belirleyen Yazılım Bakım ve İdame Yeteneği Olgunluk modelleri; Yazılım Bakım ve İdame Olgunluk Modelinde (S3M-Software Maintenance Maturity Model), düzeltici bakım ve idame olgunluk modelinde (corrective maintenance maturity model) ve Bilgi Teknolojileri Hizmeti Yetenek Olgunluk Modelinde (IT Service CMM) açıklanmıştır.

## KAYNAKLAR

(ABD Askeri Standardı (MIL-STD) 498, Software Development and Documentation, 5 Aralık 1994, s: DI-IPC-81431)

Bach, James ve Pettichord, Bret (2001). Lessons Learned in Software Testing: A Context-Driven Approach. John Wiley and Sons s.4. ISBN 0-471-08112-4.

Bass, Len; Paul Clements, Rick Kazman (2003). Software Architecture In Practice, Second Edition. Boston: Addison-Wesley, p. 21-24. ISBN 0-321-15495-9.

Beizer, Boris (1990). Software Testing Techniques, Second Edition, s..21,430. ISBN 0-442-20672-0.

Binder, Robert V. (1999). Testing Object-Oriented Systems: Objects, Patterns, and Tools. Addison-Wesley Professional, s.45. ISBN 0-201-80938-9.

Boehm, B.W. and Papaccio, P.N., 1988, Understanding and controlling software costs, IEEE Trans of Software Engineering, 14(10), 1462-1477

Bridges, W., 1995, Managing Transitions, Making the most of change, Nicholas Brealey Publishing, UK.

Brinkkemper, S., 1996, Method engineering: engineering of information systems development methods and tools, Inf. Software Technol., 38(4), 275-280.

Clements, Paul; Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Robert Nord, Judith Stafford (2003). Documenting Software Architectures: Views and Beyond. Boston: Addison-Wesley, s. 13-15. ISBN 0-201-70372-6

Davis, A.M., 1993, Software Requirements: Analysis and Specification, Prentice Hall, second Edition, 1993.

Dustin, Elfriede (2002). Effective Software Testing. Addison Wesley, s.3. ISBN 0-20179-429-2.

Falk, Jack and Nguyen, Hung Quoc (1999). Testing Computer Software, 2nd Ed.. New York, et al: John Wiley and Sons, Inc., ISBN 0-471-35846-0.

Frakes, W.B. And Kyo Kang, (2005), "Software Reuse Research: Status And Future", IEEE Transactions On Software Engineering, 31(7), July, S. 529-536

IEEE (1990), IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York: IEEE. ISBN 1559370793.

IEEE (1998), IEEE standard for software test documentation. New York: IEEE. ISBN 0-7381-1443-X.

IEEE (2006). Transactions on Software Engineering, "Introduction to the Special Issue on Software Architecture". IEEE internet sitesinden 23.09.2009 tarihinde alıntılanmıştır.

International Software Testing Qualifications Board Section 1.1.2, Certified Tester Foundation Level Syllabus, Principle 2, Section 1.3, Certified Tester Foundation Level Syllabus ISTBQ internet sitesinden 25.04.2009 tarihinde alıntılanmıştır.

Iverson K.E., "Notation as a tool of thought", Communications of the ACM, 23, s. 444-465 (August 1980)

Kaner, Cem (Kasım 2006) "Exploratory Testing" Florida Institute of Technology, Quality Assurance Institute Worldwide Annual Software Testing Conference, Orlando, FL,

Laycock, G. T. (1993). "The Theory and Practice of Specification Based Software Testing" (PostScript). Dept of Computer Science, Sheffield University, UK. Üniversitenin internet sitesinden 19.01.2009 tarihinde alıntılanmıştır.

McConnell, Steve (1996). Rapid Development: Taming Wild Software Schedules, 1st ed., Redmond, WA: Microsoft Press. ISBN 1-55615-900-5.

McConnell, Steve (2004). Code Complete, 2nd edition, Microsoft Press, 960. ISBN 0-7356-1967-0.

Myers, Glenford J. (1979). The Art of Software Testing. John Wiley and Sons, 145-146. ISBN 0-471-04328-1.

NIST Raporu (2002). Software errors cost U.S. economy \$59.5 billion annually.

Pigosky T.M. (1996). Practical Software Maintenance. New York: John Wiley & Sons. ISBN 0-471-17001-1.

SoftwareArchitectures.com (2006). "Intro to Software Quality Attributes". İnternet sitesinden 23.09.2009 tarihinde alıntılanmıştır.

SEI (2006). "How do you define Software Architecture?". SEI internet sitesinden 23.09.2009 tarihinde alıntılanmıştır.

Stellman, A. ve Greene, J. (2005). Applied Software Project Management. Cambridge, MA: O'Reilly Media. ISBN 0-596-00948-8

Sommerville, Ian ve Sawyer, Pete (1997). Requirements Engineering A good practice guide, John Wiley and Sons.

Tran, Eushuan (1999). "Verification/Validation/Certification", in Koopman, P.: Topics in Dependable Embedded Systems. USA: Carnegie Mellon University. Üniversitenin internet sitesinden 19.01.2009 tarihinde alıntılanmıştır.

University of Waterloo (2006). "A Very Brief History of Computer Science". Üniversitenin internet sitesinden 23.09.2009 tarihinde alıntılanmıştır.

Wieggers, Karl E. (2003). Software Requirements 2: Practical techniques for gathering and managing requirements throughout the product development cycle, 2nd ed., Redmond: Microsoft Press. ISBN 0-7356-1879-8

Young, Ralph R. Effective Requirements Practices. Boston: Addison-Wesley, 2001

## EK-3: Yazılım Proje Risk Yönetimi Adımları

### YAZILIM PROJE RİSK YÖNETİMİ ADIMLARI

- I. Adım: RİSKLERİ TANIMLA**
- Fikir yaratma araçları
  - Eşgüdüm araçları
  - İnsan yönetimi araçları
- II. Adım: RİSKLERİ ANALİZ ET**
- Fikir yaratma araçları
  - Karar verme araçları
  - Proje yönetimi araçları
- III. Adım: RİSKLERİ ÖNCELİKLENDİR**
- Fikir yaratma araçları
  - İnsan yönetimi araçları
  - Karar verme araçları
- IV. Adım: RİSKE KARŞI STRATEJİ PLANLAMA**
- Fikir yaratma araçları
  - İnsan yönetimi araçları
  - Karar verme araçları
- V. Adım: RİSK YÖNETİM PLANININ YÜRÜTÜLMESİ**
- İnsan yönetimi araçları
  - Proje yönetimi araçları
- VI. Adım: SONUÇLARIN DEĞERLENDİRİLMESİ**
- İnsan yönetimi araçları
  - Karar verme araçları
  - Proje yönetimi araçları
- VII. Adım: SONUÇLARIN RAPORLANMASI**
- İnsan yönetimi araçları
  - Eşgüdüm araçları

## **EK-4: Yazılım Projelerinde Doğru Eleman Seçimi ve Yazılım Proje Ekibinin Oluşumu ve Gelişimi Aşamaları (TUCKMAN Modeli)**

### **1. YAZILIM PROJELERİNDE DOĞRU ELEMAN SEÇİMİ**

Yazılım projelerinin 3 temel bileşeni vardır, Goldenberg üçlüsü de denen bu bileşenler aslında insan-süreç-teknoloji üçgenidir. Teknoloji, bu üçlü içinden en kolay temin edilebilen unsur iken, bir firmanın gayri maddi aktifleri olan insan ve sürecin kalitesini arttırmak zaman almaktadır. Bilişim projelerinde konvansiyonel proje yönetimine ek olarak aslında ekstra bir şey yapmaya gerek bulunmamaktadır. En önemli etmen doğru proje yöneticisidir. Yazılım projelerindeki başarı tamamen kapsam belirlenmesine bağlıdır, olayı teknolojik şova döndürüp süreç ve değişim yönetimi atlanırsa sonuç başarısız bir yazılım projesi olur.

**1.1. Proje Yöneticisi:** Proje Yöneticisi, bir projeyi yönetmek üzere görevlendirilen ve projeyi, mümkün olan en yüksek üretkenlik, en düşük belirsizlik ve risk ile yürütmekten sorumlu kişidir. Proje yöneticisi, projenin her şeyinden sorumlu olduğundan gerekli ve yeterli yetkilerle donatılmalıdır. Proje yöneticisi hem proje ekibini kurmak, hem de projeyi gerçekleştirirken yönlendirmekten sorumludur. Bu sorumluluklar şunları içerir:

- Proje ekibinin hangi niteliklere sahip kişilerden oluşacağını belirlemek,
- Bu kişileri ekibe dâhil etmek,
- Görevlerini ve sorumluluklarını bildirmek,
- Ekibin gerekli kaynaklara ulaşmasını ve gerekli eğitimi almasını sağlamak,
- Ekibin proje ile ilgili görevlerini yerine getirmelerini sağlamak.

Proje yöneticisi, proje ekibinin günlük işleri ile doğrudan ilgilenmelidir. Burada, liderlik, proje yöneticisinin en önemli aracıdır ve proje yöneticisi tarafından her zaman açık ve net biçimde ortaya konulmalıdır. Proje yöneticisi, proje ekibinin, proje hedeflerine ulaşması için gerekli olan isteklendirme (motivasyon), cesaret ve yönlendirmeyi sağlamalıdır. Bunu yaparken, proje ekibinin insanlardan oluştuğu ve insan gruplarının bir arada çalışmaları ve birbirleri ile etkileşimde bulunurken hangi evrelerden geçmekte olduklarının bilinmesi önemlidir.



Proje yöneticisinin en temel görevi, projenin planlanması ve bu planı uygulayarak projenin başarıyla bitirilmesidir. Bu kapsamda, proje yöneticisi tarafından yapılması zorunlu işler, yapısal proje yönetimine göre on temel adımda (Tablo-E...1) toplanmaktadır.

Tablo-E4.1: Yapısal Proje Yönetimine Göre On Temel Adım

|   |    |  |
|---|----|--|
| Projenin Planlanması                    | 1  | Amacın kaybedilmemesi, hedefe odaklanması  |
|   | 2  | Yapılacak işlerin listelenmesi   |
|   | 3  | Tek bir proje yöneticisiyle projenin yönetilmesi   |
|   | 4  | İşi yapacak kişilerin ve iş dağılımının belirlenmesi.                                      |
|   | 5  | Beklentilerin yönetilmesi, hata payı bırakılması, geri adım atacak imkânların yaratılması. |
| Planın Uygulanması ve Hedefe Ulaşılması | 6  | Uygun bir liderlik biçimi seçilmesi  |
|   | 7  | Projede etkin takip ve kontrolün sağlanması  |
|   | 8  | Projede gelinene durum konusunda proje elemanlarının bilgilendirilmesi                     |
|   | 9  | 10'uncu adıma ulaşana kadar 1' den 8' e kadar olan adımların tekrarlanması                 |
|   | 10 | Projenin belirlenen hedeflere uygun olarak tamamlanması                                    |

(10 Adım, Fergus O'Connell, Silver Bullet, 1996)

Proje Yöneticisi, projeyi uygularken; planlama, yürütme, İzleme ve değerlendirme biçiminde gerçekleşen döngüsel bir süreç yaşayacaktır. Bu süreç, Proje Yöneticisi Denetim Döngüsü olarak da adlandırılabilir. Bu döngünün resmi olarak haftalık, gayri resmi olarak günlük bazda uygulanması, projenin başarı ile tamamlanmasını sağlayacaktır. Proje Yöneticisi olarak görevlendirilecek kişide aşağıdaki özellikler aranmalıdır:

- Örgütlenme ve liderlik deneyimi,
- Gerekli kaynaklara başvurma (araştırmacılık),
- Değişik kaynakları uyumlu olarak kullanabilme becerisi,
- İletişim ve çeşitli yöntemleri uygulama,

- Elemanlara sorumluluk verme ve onları izleme becerisi,
- Güvenilirlik,
- İnisiyatif sahibi olma,
- Hızlı öğrenebilme,
- İnsanlarla ilişkilerde yüksek niteliklere sahip olma,
- Ağacı değil de ormanı görebilen geniş ufuklu görüş açısına sahip olma,
- Birebir tartışmalarda iyi konuşabilme,
- İyi sunuş yeteneđi,
- Söze ve yazıya dayalı iletişimi iyi yapabilme,
- Dili iyi kullanabilme,
- Deđişik seviyeden yönetim kadroları ile iş yapması gerektiğinde her seviyede eşit derecede rahat davranabilme,
- Kendine güvenme,
- Hayata iyimser bakma;
- Planlama yapabilme,
- Problemleri anında fark etme ve yaratıcı çözümler bulma.

İyi bir proje yöneticisi olabilmek için bu görevi üstlenecek kişiler tarafından aşağıdaki noktalara dikkat edilmesinde büyük yarar görölmektedir.

- Sağduyu sahibi olunması,
- Stres altında denetimin kaybedilmemesi,
- Aynı anda beliren, birbiri ile karşılıklı ilişki içinde olan etkinliklerin düzenlenebilmesi,
- Geleceđe odaklanması,
- Bađımsız olunması,
- Proje başarısı için birden fazla alternatif yönelim içinden dođru yaklaşımın seçilmesi,
- Etkili liderlik becerilerinin uygulanması,

- Sözlü ve yazılı iletişim yeteneğinin geliştirilmesi ve etkinlikle kullanılması,
- Ekip içinde kişiler arası iletişimin geliştirilmesi,
- Toplantı yönetme, sunuş ve tartışma (müzakere) konusundaki becerilerin geliştirilmesi.

## 1.2. Proje Ekibinin Seçimi Kriterleri

Üstlenilen bilişim projesinin başarıyla uygulanması tüm uzman personelin ve kaynakların istenilen zamanda sağlanması ile mümkündür, Ancak gerekli elemanların seçimi sanıldığından daha zor olabilir. Çünkü verilen işi başarıyla yerine getirdiği bilinen personel sıkça istenildiğinden onları ekibe katmak zor olabilir.

Projeye başlamadan önce nasıl bir bilgi ve beceri karışımının gerektiği belirlenmelidir. Projeye olumlu bakan, bu isten kazanacakları beceri ve deneyime inanan kişilerin katılımı çok daha yapıcı olacaktır. Bu kapsamda proje ekibinin seçiminde göz önünde bulundurulması gereken faktörler şunlardır:

- Projenin hangi grup ve birimleri etkilediğinin belirlenmesi,
- Proje hakkında bilgili, deneyimli elemanların tespiti,
- Farklı elemanların proje grubuna katılımlarının sağlayacağı yararlar ya da engellerin ortaya konması,
- Performanslarını proje sonuna kadar koruyabilecek elemanların tespiti,
- Proje ekibine katılmanın kişilerin fayda ya da zararlarına olup olmadığının netleştirilmesi,
- Seçilen ekibin grup çalışması yapmaya uygunluğunun değerlendirilmesi,
- Farklı elemanlardan proje ekibine katmadan belirli konularda uzman şahısların projeye katkı sağlayıp sağlayamayacağının belirlenmesi

Proje grubu elemanlarının sorumlulukları şunlardır:

- Proje kalitesini olumsuz etkileyebilecek, kullanıcı istekleri ile uyumsuzluk gösterebilecek problemleri çıkmadan önce engelleyecek önlemleri almak,
- Karşılaşılan problemleri çözmek ve kaydını tutmak,

- Problemlere proje çerçevesinde çözüm yolları önermek ve uygulamaya sokmak,
- Probleme neden olan aksaklık giderilinceye kadar, bu sorundan etkilenecek olan işleri denetlemek.

Projenin genel yapısında, matris örgütlenmesindeki proje ekibine ek olarak, bir yönlendirme kurulu bulunmalıdır. Bu kurul; genelde üst düzey yönetim, çalışma ürünleri ortaya çıkıncaya kadar, projelere gereken önemi ve dikkati ayırmamasına rağmen, ürünler ortaya çıktığında da projeye etkileri çok sınırlı olmaktadır. Çünkü planlama, tasarım ve maliyetler daha önceden belirlenmiş ve kararlaştırılmıştır. Proje yöneticisi ve ekibi, teknik açıdan ne denli yetkin ve konularının uzmanı olurlarsa olsunlar, bir projenin arzu edilen biçimde gerçekleşmesi için, yönetimin üst kademesinin kesintisiz desteğine ihtiyaç duyacaktır.

Üst düzey yöneticiler, projeye ve proje yöneticisine kesintisiz destek sağlamalı ve katkılarını görünür bir biçimde sürdürmelidir. Bunu sağlamak üzere Proje Yönlendirme Kurulunun oluşturulması önerilir.

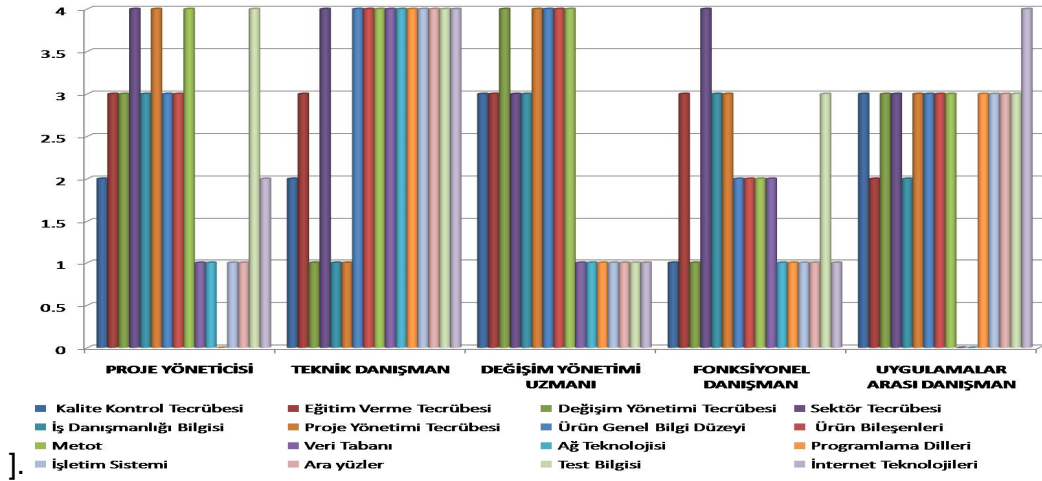
Proje Yönlendirme Kurulu, projenin ilerleyişini gözleyen, gerektiğinde yardım sağlayan ve projenin başarısını değerlendiren gözlemci bir gruptur. Proje ile ilgili birimlerden temsilciler ile oluşturulur. Bunlar:

- Proje yöneticisi,
- Müşterinin proje yöneticisi,
- Proje destekleyicisi,
- Müşteriden temsilciler,
- Diğer üst düzey yöneticiler (gerektiğinde davet edilmek üzere) olabilir.

Günlük proje çalışmalarında yer almasa da büyük çaplı "proje gözden geçirme" toplantılarında, proje ile ilgili değişiklik isteklerini değerlendirme toplantılarında, büyük kaynak ihtiyaçlarını sağlamaya çalışırken ve sorunları çözmede bu kurulun katılımına ve onayına başvurulmalıdır.

Proje yöneticisi, proje yaşam döngüsünün en basında bu kurulun oluşumu için uğraş vermeli ve bu kurulun projeye katkıda bulunmasını düzenli bir biçimde sağlamalıdır.

Proje yöneticisi ve anahtar proje elemanları disiplinler arası çalışabilecek düzeyde olmalıdır. Günümüzde olaya bir bütün olarak bakıp paralel süreçleri kesintisizce ilişkilendirebildiklerinden endüstri ve işletme mühendisleri bu role en çok uyan uzmanlardır (Cardoso 2005:192-212).



Şekil-E4.1: Yazılım Projelerinde Doğru Eleman Seçimi

Şekil-E...1:'deki grafik incelendiğinde; üç genel özelliğin altına diğer alt özelliklerin sıralandığı, “Hiç (0)”, “Temel (1)”, “Orta (2)”, “Uzman (3)”, “İleri Derece Uzman (4)” olarak derecelendirildiği görülecektir. Bu özellik/niteliklere sahip personel atandığı takdirde projede başarıdan söz edilebilir.

Uluslararası uygulamalarda sektör, değişim yönetimi, kalite yönetimi, iş danışmanlığı önem kazanırken, ilk başlarda ne yazık ki proje yöneticileri neyi neden yaptığını bilmeyen teknik uzmanlardan seçildiğinden, projelerin yönetilememesi yanında, zaman, işgücü ve donanım yanlış yer ve zamanda kullanılarak istenilen sonuca da ulaşılamamıştır (Macvitte 2001:97).

- **Teknik Danışman:** Teknik bilgisi ileri derecede uzman seviyesindedir, ürün işlevini de iyi bilen, proje metodolojisine hâkim, sektör bilgisi ve eğitim verme yeteneği üst düzeyde olan bir proje katılımcısıdır. Ülkemizde teknik uzman eksikliği kendini burada da belli etmiş ve bilgisayar teknisyenlerine teknik danışmanlık payesi verilmiş ve eksik yabancı dilleri yetersiz teknik bilgileri ile projelerin korkulu rüyaları olmuşlardır (Erkan 2007).

- **Değişim Yönetimi Uzmanı:** projenin proje yöneticisinden sonraki, hatta en önemli bireyidir. Genel iş idaresi bilgisi ve sektör bilgisi üst düzeyde olan bu danışmanlar, ürün bilgisi ile mevcut durum analizinden sonra firmanın değişim rotasını çizerler. Firmanın stratejik plan gereği seçilen ürün ile harmonizasyonunu ve sektör tecrübeleri sayesinde kıyaslama (benchmarking) analizleri sonucu firma yönlendirmesinin kusursuz yapılmasını sağlarlar (Piturro 1999:41).

- **Fonksiyonel Danışman:** Bu proje temsilcisi ürünün kusursuz olarak firmaya kurulumunda en büyük rolü oynar. Sektör bilgisi gereği, firmanın ihtiyaçlarına göre süreçlerin belli senaryolar dâhilinde simülasyonunu gerçekleştirir, teknik bilgisi gereği de teknik danışmanla uyum içerisinde çalışarak proje uygulama süresini en aza indirir (Ross ve Vitale 2000:233-241). Ülkemizde bu danışmanlık tipinde daha çok genel iş idaresi ve sektör bilgileri yetersiz ve tek dertleri ürünü çalışır şekilde teslim etmek olan yeni mezunlar çalışmaktadır. İş süreçleri, teknik uyum, yapılması gereken testler onlar için bir şey ifade etmediğinden böylesi personelin görevlendirildiği projelerin bitirilmesi çok zordur. Daha kötüsü, proje kapanma şartları bile netleşmemesine rağmen fonksiyonel danışmana göre bitmiş, teknik danışmana göre başlamış süreçlere rastlamak mümkündür.

- **Uygulamalar Arası Uzman Danışman:** Aynı anda birden fazla sayıda uygulama kurulduğunda, gerçek zamanlı olarak bütünleştirilmeleri (integration) gerektiğinden, teknik ve uygulama danışmanlığı nitelikleri yüksek ve aynı zamanda genel bilgi yetenekleri de uzman seviyesinde birine ihtiyaç gösterir. Bu çerçeveden bakıldığında uygulamalar arası danışmanlar, firmanın bütün iş süreçlerine hâkim, ürün ve ürün teknik bilgisi üst düzeyde personel oldukları görülmektedir. Bu tip danışmanlık nitelikleri, genellikle küçük projelerde proje yöneticiliği ve uygulama danışmanlığı yapma şartı altında teknik eğitim alınması sonucunda kazanılmaktadır. Bu görev çoğunlukla, fonksiyonel danışman görevi yaparken, teknik ekip ile anlaşmazlığa düşüp teknik yeteneklerini artırmış personel tarafından yerine getirilmektedir. Aslında yetersiz teknik danışmanlar farkında olmadan böyle bir katkı sağlamışlardır. Bahse konu nitelikleri, genel iş idaresi eğitim ve yetenekleri ile birleştirilebilen personel de genelde firmaların teknik birim başkanı olurlar.

## 2. YAZILIM PROJE EKİBİNİN OLUŞUMU VE GELİŞİMİ AŞAMALARI

Proje yöneticisinin temel sorumluluklarından birisi proje ekibini oluşturmaktır. Proje ekibi oluşturma, projenin ilk safhasından itibaren gerekli ve önemlidir. Ancak,

proje sonuçlanıncaya kadar sona ermeyecek bir süreçtir. Proje ekibi, projenin gerektirdiği teknik bilgi ve beceri ile donatılmış, eksik kalan bilgi ve becerileri için proje öncesinde ve/veya proje süresince kısa süreli eğitim programlarından geçirilmiş kişilerden oluşturulur. Bu ekipteki elemanların başlangıçta proje kapsamına giren konularda asgarî bilgi birikimine sahip olmaları projenin başarısı açısından önemlidir. Proje süresince proje ekibine konu ile ilgili kişiler eklenebilir, görevleri tamamlanan kişilerse ekipten ayrılırlar. Ayrıca, gerekli görülen durumlarda, kurum dışı danışmanlık hizmetlerinden de yararlanılabilir.

Proje yöneticisi, proje planını kullanarak ekipteki elemanlar için kişisel faaliyet listeleri çıkarmalıdır. Bu listeler, proje ekibindeki elemanların üstlenecekleri görev ve sorumlulukları, diğer ekip elemanları ile ortak yürütülecek işleri ve iş takvimlerini içerir. Proje yöneticisi bu listeleri kullanarak görev eşgüdümü yapabilir ve ekip elemanları arasındaki iş yükü, dağılımını gözlemleyebilir. Ekip elemanları ise yapmaları gereken tüm işleri bir arada görür. Kişisel faaliyet listelerinin dağıtım sırasında proje yöneticisi her ekip elemanı ile ayrı konuşarak görevlendirmeye ilgili konulara açıklık getirmeli, ortak yürütülecek çalışmalarda inisiyatifin kimde olduğunu belirtmeli ve bu sırada kişilerin üstleneceği görevler konusunda oluşabilecek kaygıları da gidermelidir.

Proje ekibi içinde, projenin sahibi olan birimin, gerekli bilgi ve deneyime sahip ve yeterli sayıda elemanın "Kullanıcı Temsilcisi" olarak görevlendirilmeleri şarttır. Projelerin bütünlüğünün korunabilmesi ve başarıyla sonuçlandırılması için, özellikle projelerin başlangıcında hemen her konuda katılımı gerekli olan kullanıcı temsilcilerinin, süreklilik arz eden personel arasından seçilmeleri yerinde olur. Kullanıcı temsilcilerinin, projeye etkin katılımının ve projeyi sahiplenmelerinin önemi, proje başlangıcında proje sahibine anlatılmalı ve tüm proje boyunca tam bir işbirliği içinde çalışılması sağlanmalıdır.

Bir proje yöneticisi, C. Kormanski tarafından tanımlanan "Tuckman Modeli" adı ile bilinen grup dinamikleri yöntemi ile ekibine bilinçle yaklaşabilir ve uygun liderlik tarzını uygulayabilir. Bu modelde, grup dinamikleri sıra ile incelenir. Grubun olgunluk seviyesi, görevler ve davranış ilişkisine göre değerlendirilerek grup performansı aşamalara bölünür. "Tuckman modeli" beş ayrı aşamayı tanımlar:

- **Oluşum (Forming):** Grubun girdiği ilk aşamadır. Grup üyeleri birbirleri ile tanışır, projenin genel amacını ve ekip yapısının nasıl olacağını öğrenirler. Bu aşamada, proje yöneticisinin amacı, ekip üyelerinin, proje örgütü,

iletişim yapısı ve temel sorumluluklarını anlamalarını sağlamaktır. Proje yöneticisi, basit ekip oluşturma aktiviteleri düzenleyebilir.

- **Karışma (Storming):** Ekip hızla, basit görevler için verilen duygusal yanıtların ve grup içi çelişkilerin ortaya çıktığı karışma aşamasına girer. Karışma davranışları ekip üyelerinin verilen görevleri anlamaya başlamaları ve proje liderlik yapısını sınıadıkları tipik bir dönemdir. En çok görülen tepkilerin nedenleri, görevin büyüklüğü, verilen zamanının az olduğunun düşünülmesi ve bu asamaya kadar yapılmış olan planlama ve izleme çalışmalarının göreceli olarak az olmasıdır. Proje yöneticisinin görevi bu normal aşamanın geçirilmesinde ekip üyelerine yardım etmek ve çelişkilerin kabul edilebilir bir düzeyde çözümlenmesini sağlamaktır.

- **Düzenlenme (Norming):** Bu dönemin özelliği, görevlerle ilgili düşüncelerin açıkça ifade edilmesi, iletişimin düzenli olması, ekip üyelerinin daha işbirlikçi olmaları ve uyum göstermeleri ile tanımlanır. Bu safhada ekip, ekip üyelerinin performanslarından beklenenleri oluşturur ve bireyler ekip içindeki rollerini tamamlarlar. Proje yöneticisi, bilginin etkin biçimde dağıtılmasını özendirerek ve ekip içinde birlik, yüksek moral ve üretkenliğin artırılmasını sağlayarak çok etkin olur.

- **Çalışma (Performing):** Ekip, beklentileri yerine getirmek için çalışır. Ortaya çıkan sorunlar ve çelişkiler normal seviyelerdedir. Ekip gerekli teknikleri ve prosedürleri kullanarak bunları halletmek için çalışır. Ekip üyeleri bağımsız hale gelirler ve işlevsel ilişkiler bu durumu devam ettirir. Bu aşamada, proje yöneticisinin ana rolü, ekip performansını ve birbirleri ile etkileşimlerini kolaylaştırmak ve sorunlar çıktıkça çözmektir.

- **Çözülme(Adjourning):** Bu hem en kolay hem de en zor aşamadır. Projenin sonunda ulaşılan yüksek performans seviyeleri çok büyük kişisel tatmin verir. Bu durumda, sıkı çalışma temposu ve yoğun katılımın ardından gelen çözülme ile büyük bir boşluk duygusu ve bir sonraki proje için endişe oluşabilir. Proje yöneticisi, olumlu bir ortamın yaşanması için uğraş vermelidir. Ekip ve ekip üyelerinin performansları ile ilgili tanıtım ve kutlamaların yapılması bu aşamada yapılacak ana işlerdendir.

Ekip gelişmesinde, bu beş aşama sıralı bir gelişim gösteriyor gibi olmasına rağmen, genelde böyle seyretmez. İki ya da daha fazla safhanın aynı anda olması sıkça görülür. Proje yöneticileri bu safhaları anlamalı ve gerekli uygulamaları yaparak ekibin çalışmalarını düzenlemelidir.



## **KAYNAKLAR**

Cardoso J. Workflow Handbook 2005, chapter How to Measure the Control-flow Complexity of Web Processes and Workflows, 2005, s. 192-212)

Macvitte, L. "Buckle Up: Implementing an ERP Takes Time and Patience," Network Computing, (2001), C.12. S.6. s.97.

Erkan, T. Erman, Akademik Bilişim 2008, Çanakkale Onsekiz Mart Üniversitesi, Çanakkale, 30 Ocak - 01 Şubat 2007.

Pitirro, M., "How Midsize Companies Are Buying ERP," Journal of Accountancy, C.188. S.3. 1999, s.41.

Ross, J. W. ve Vitale, M. R. The ERP Revolution, Surviving vs. Thriving. Information Systems Frontiers; special issue of on The Future of Enterprise Resource Planning Systems, 2(2), 2000, s. 233-241.

## EK-5: Yazılım Geliştirme Model, Metodoloji ve Teknikleri

### YAZILIM GELİŞTİRME MODEL, METODOLOJİ VE TEKNİKLERİ

#### 1. YAZILIM GELİŞTİRME MODELLERİ

##### 1.1. Şelale (Waterfall) Modeli

En çok bilinen ve en eski süreç olan şelale modeli (waterfall model), içinde geliştirme faaliyetlerinin ihtiyaç analizi (requirements analysis), tasarım (design), uygulama (implementation), test ve geçерleme [testing (validation)], entegrasyon (integration) ve bakım ve idame (maintenance) adımlarını takiben bir şelale gibi sürekli olarak yukarıdan aşağı aktığı aşamalı yazılım geliştirme modelidir.

Şelale terimi 1970'de Winston W. Royce (1929–1995) tarafından yayınlanan bir makaleye (makalede şelale terimi kullanılmamasına rağmen) dayandırılarak literatüre kazandırılmıştır (Rerych 2009). Royce makalesinde şu an şelale modeli olarak bilinen yaklaşımın her ne kadar hatalı olduğunu vurgulasa da bir başlangıç konsepti olduğunu ileri sürmüştür. Makalesinde, başlangıç modelinin her aşamadan elde edilen ve müteakip veya alt aşamaları etkileyen geri beslemelere dayanarak tekrarlı bir modele nasıl dönüştürülebileceğini araştırmıştır. Makalesinde sadece başlangıç modeli olarak sunduğu ve eleştirdiği model dikkate alınmış, sonuçta sunduğu tekrarlı model tasarımıyla ilgilenilmemiştir ve şelale modeli için makalesi kaynak olarak gösterilmiştir.

Royce'un orjinal şelale modelinde aşamalar aşağıdaki şekilde sıralanmıştır.

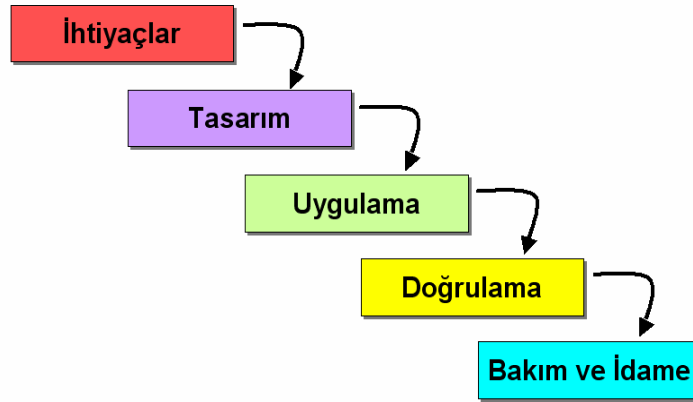
- İhtiyaçların Belirlenmesi (Requirements specification)
- Tasarım (Design)
- Yapım (uygulaması veya kodlama) (implementation or coding)]
- Entegrasyon (Integration)
- Test ve Hataların Ayıklanması (Testing and debugging)
- Kurulum (Installation)
- Bakım ve İdame (Maintenance)

Müteahhitlerin binanın iskeleti yapılmadan inşasına müsaade etmemeleri örneğinde olduğu gibi, her bir adımın tamamlanmasını müteakip sonraki adıma geçilir. Başka bir ifade ile bir aşama tamalanmadan diğerine geçilme (Şekil-E..1).

Hazırlanması planlan yazılım, tasarlanır (bu tasarım verilen ihtiyaçların uygulanması için hazırlanmış bir plan niteliğinde olmalıdır) ve kodlayıcıların takip edebilmesi için mavi kopyası çizilir. Tasarım bütünüyle tamamlandığında, kodlayıcılar tarafından kodlamaya geçilir. Bu uygulama safhasının müteakip aşamalarına doğru farklı timler tarafından üretilen birbirinden farklı sistem bileşenleri birbirine entegre edilir. Kodlama ve entegrasyon safhaları tamamlandıktan sonra, yazılım test edilir ve hata ayıklaması yapılır. Önceki safhalarda karşılaşılan hatalar burada düzeltilir. Müteakiben yazılımın kurulumuna geçilir. Yazılıma yerni yetenekler kazandırılması ve bu yeteneklerin test ve hata ayıklaması da bakım ve idame safhasında yapılır.

Sürecin ilk aşamalardaki (ihtiyaçların belirlenmesi gibi) hataların düzeltilmesine müsaade etmediğine dair yanlış bilgiler mevcuttur. Aslında bu faaliyet, içinde değişim yönetiminin yer aldığı ihtiyaçlar yönetiminin ana odak noktasında (domain) yapılmaktadır.

Bu süreç kapsamında, Royce tarafından geliştirilen nihai model de dahil bir çok küçük veya büyük oranda değiştirilmiş şelale modeli (modified waterfall models) ortaya çıkmıştır.



**Şekil-E5.1:** Şelale Modeli

Bu model ve getirdiği yaklaşım özellikle büyük savunma sözleşmeleri gibi yüksek riskli projelerde kullanılmaktadır. Bu modeldeki problemler “olgunlaştırılmamış mühendislik” çalışmalarından değil, özellikle ihtiyaç analizi ve ihtiyaç yönetiminden kaynaklanmaktadır. ABD savunma bakanlığı standart dokümanında (DOD-STD-2167) başarısızlık oranları üzerine yaptığı şelale modelini destekleyen çalışmalar, bir projenin süreçlerine sadık kaldığı sürece (özellikle

ihtiyaların toparlanmasında), yazılımın son srmnde kullanılmayacak bazı zellikler bulunması ihtimalinin de o kadar artacađını gstermiřtir.

Kullanıcı ve yklenici arasında sık sık mřterek gzden geirme ařamaları uygulanır. Yklenici aslında risk altında geliřtirme faaliyetini yrtr ve tasarımı geliřtirir, ancak Kritik Tasarım Gzden Geirmesi (CDR-Critical Design Review) adı verilen anahtar tasarımı tasfiye etmek zorunda kalabilir. Bu da mhendislik problemlerinin mhendislerden alarak bařka becerilere sahip kullanıcılara kaydırılmasına neden olabilir.

Yazılım geliřtirilirken ne kadar zaman harcanırsa, ileriki safhalarda ve yazılımın mr devrinde o oranda ekonomiklik sađlanmaktadır. Bir hata, retim dngsnn ihtiya belirleme ya da tasarım gibi ilk safhalarında tespit edildiđinde; ayıklanması/giderilmesi ileriki safhalara kıyasla zaman, gayret ve para anlamında daha ucuza gelmektedir. Bazı tahminlere gre; bakım ya da retim sonuna kadar bir ihtiya hatalı/eksik belirlenmiř olduđu takdirde 50 ila 200 kata kadar daha pahalıya gelebilmektedir.

Bu aslında řelale modeli ve Byk Tasarım nermesi (Big Design Up Front - BDUF) altında yatan ana dřncedir (İhtiyalar ve tasarımın tamamen dođruluđu konusunda emin olma bakımından ne kadar zaman harcanırsa, ileride o oranda zaman ve gayret tasarrufu sađlar. Bu nedenle, řelale sreci boyunca programın mteakip ařamasına geilmeden nce mevcut ařamanın % 100 oranında dođru olarak tamamlandıđından emin olunması gereklidir. Program ihtiyaları, tasarımlama bařlamadan nce eksiksiz ve dođru olarak tanımlanmiř ve sabitlenmiř olmalıdır. Aynı řekilde tasarımın uygulanmasına geilmeden nce de tasarımın tam manasıyla kusursuz olması gereklidir).

řelale modeli iin diđer nemli bir husus dokmantasyona da (ihtiya dokmantasyonu, tasarım dokmantasyonu vb.) kaynak kodu kadar nem verilmesidir. Tasarım ve dokmantasyonu nispeten az kullanan metodolojilerde ekip yelerinden biri ayrıldıđında, elde edilmiř bilgilerin byk kısmı kaybolabilir ve bu da projede onulmaz bi durum yaratabilir. İyi bir tasarım dokmantasyonunun varlıđı [řelale Modeli ve Byk Tasarım nermesinde (BDUF) olduđu gibi] yeni ekip yeleri veya tamamen yeni teřkil edilmiř bir ekip bahse konu dokmantasyonu okuyarak projeye kolayca uyum sađlayabilirler.

Yukarıda aıklanan hususlar yanında kimileri řelale modelini basit ve tartıřılır derecede disiplinli bir yaklařım oldu iin tercih ederler. řelale modeli yapısal bir yaklařım sunduđundan, dođrusal olarak ilerlediđinden, kolayca anlařılabilir ve

açıklanabilir aşamalar içerdiğinden ve geliştirme sürecinde kolayca belirlenebilen kilometretaşları sağladığından bu modelin savunucuları muhtemel bir kaostan korunulduğunu ifade ederler. Şelale modeli ve BDUF genelde ihtiyaçları hiç değişmeyen, büyük değişiklikler gerektirmeyen ya da başka bir ifade ile sabit seyir izleyen projelere daha çok uyduğu ve bu özelliği sayesinde de ihtiyaç duyulduğunda tasarımcılara sistemde problem sahalarını eksiksiz tespit etme ve tasarımı kodlanmaya başlamadan önce düzeltme imkânının sağlandığı söylenebilir.

Şelale modeli aynı zamanda kodlamayı gerçekleştirecek personelin, iyi düzenlenmiş eksiksiz bir tasarımı doğru olarak takip etmelerini ve sistem bütünleştirmesinin kusursuz şekilde ilerlemesini sağlar.

Şelale modelinin uygulamada kötü bir araç olduğu çoğu yazılımcı tarafından dile getirilmektedir. Bu aslında, herhangi bir proje için ürün ömür devrinin bir sonraki aşamalara geçmeden önce kusursuz hale getirilmesi ve bundan ders çıkarılmasının mümkün olmadığı düşüncesinden kaynaklanmaktadır. Örneğin, müşteriler, çalışan bir prototip görmeden önce aslında ne istediklerinin farkına varamaz ve değerlendirmesini yapamazlar. İhtiyaçları sürekli olarak değiştirebilirlerken program tasarımcı ve kodlayıcılarının bunun üzerinde sınırlı etkileri olabilir. Müşteriler ihtiyaçları tasarım tamamlandıktan sonra değiştirmeye kalktıklarında tasarım için BDUF kapsamında ne kadar zaman ve gayret harcanmış olursa olsun yeni ihtiyaçları karşılayacak şekilde düzenlenmesi gerekir. Tasarımcılar, tasarımlarını gerçekleştirirken gelecekte ortaya çıkabilecek kodlama zorluklarının farkına varamayabilirler Bu durumda, ihtiyaçları tam olarak karşılamayan, hatalı tahminler içeren ve yeni belirlenen problem sahalarına cevap veremeyecek bir tasarımın kodlanması konusunda ısrarcı olmaktansa tasarımın yenilenmesi her zaman için daha olumlu sonuç verir.

Steve McConnell, şelale modelinin kullanım alanlarına ilişkin değerlendirmeler sunan "Tamamlanmış Kod (Code Complete)" adlı kitabında tasarımı "ihtiyaç sınırları tamamlanmadan önce kesin olarak bilinmeyen problem ya da büyü problem (wicked problem)" olarak tanımlamaktadır. Bu, yukarıda da açıklanan bir safha kursuz olarak tamamlanmadan diğer safhalara geçilmesini engelleyen şelale modeli zorluklarını ifade eder niteliktedir. David Parnas, "Rasyonel bir Tasarım Süreci: Nasıl ve Niçin Ondan Kurtulmalıyız (A Rational Design Process: How and Why to Fake It)" başlıklı makalesinde aşağıdaki ifadeyi kullanmıştır:

*“Çoğu sistem detayının, sistemin kodlanmasına geçtikten sonra farkına varırız. Tasarımımızda hatalı olduğunu değerlendirdiğimiz hususlarda geri dönme ya da baştan başlamaktan çekinmemeliyiz.”*

Şelale modelinin diğer belirgin özellikleri aşağıda sunulmuştur:

- Üzerinde çalışılan program için ihtiyaçları belirleyen ve yazılım sistemini tasarımılayanlar konularında üst seviyede bilgi ve beceri sahibi olmadıkları takdirde, izleyen safhada belli bir zaman harcanmadan yazılım sürecinin her bir safhada neye ihtiyaç duyulduğunun tam olarak anlaşılması zordur. Bu sayede müteakip safhalardan gelen geri beslemeler önceki safhaların tatminkâr şekilde tamamlanmasını sağlamaktadır (tasarım aşamasının kodlama aşamasında gelen geri bildirimlerle tasarım aşamasındaki hataların farkına varılması gibi). Şelale modelini destekleyen bir görüşe göre; daha önce benzer projelerde çalışmış tecrübeli tasarımcıların prototipleme ya da kodlamaya gerek kalmadan sezgisel olarak problem sahalarını tahmin edebilecekleri değerlendirilmektedir.
- Tasarımın, kodlamanın ve doğrulama aşamalarının sürekli test edilmesi kendilerinden önce gelen aşamaların doğrulanabilmesi için gereklidir. İhtiyaçların çatışmaması ve karşılanabilmesi bakımından prototip tasarımının, tasarım sürecinin bilgilendirilmesi ve hatalı hususların tespiti bakımından kodlamanın, uygulamamının planlandığı ilerlediğinden emin olunması bakımından da kodlamanın doğrulanması ve bütünleştirilmesinin sürekli olması gereklidir.
- Geliştirilmiş yeni sürümlerin yayımlanması yazılım geliştirme ekibi ve müşterilerinde güven yaratılması bakımından sıkça başvurulan bir yöntemdir.
- Üzerinde çalışılan yazılım konusunda yeterli tecrübeye sahip kişilerce gerçekleştirilmediği sürece, maliyet ve zaman tahminlerinin yapılabilmesi için her bir yazılım geliştirme aşamasının sık sık gözden geçirilmesi gereklidir.
- Şelale modeli, bir proje üzerinde hiçbir resmi tatbik yönetimi getirmemesi yanında proje ve planlama kontrolü ile risk yönetimi de model içinde yer almamaktadır.
- Her bir aşamada çok özel yetenek setlerine ihtiyaç duyulduğundan, kaynak kullanımının optimizasyonu için çoklu proje işletilmesine ihtiyaç vardır.



Tekrarlamalar önceden de planlandığı gibi 6 ay ila 2 yıl sürer. Her bir safha bir tasarım hedefi ile başlar ve kullanıcı tarafından sürecin o ana kadarki işleyişi gözden geçirilir. Projenin nihai hedefi göz önünde tutularak, analiz ve mühendislik faaliyetleri projenin her safhasında uygulanır.

Spiral model kullanılarak;

- Önemli konular daha önce belirlendiğinden, tahminlerin (bütçe, program, vb.) iş yürüdükçe daha gerçekçi olması sağlanır.
- Yazılım geliştirmenin genellikle kaçınılmaz bir sonucu olan değişiklik yapılması hususuyla daha kolay başa çıkılabilir.
- Yazılım mühendislerinin taşın altına ellerini daha erken sokmaları ve projenin merkezinde çalışmaya daha erken başlamaları sağlanabilir.

Spiral modele ilişkin aşamalar aşağıda açıklanmıştır.

- Yeni sistem ihtiyaçları tanımlanabileceği en ince detayda tanımlanması (Bu, mevcut sistemin iç ve dış kullanıcılarını ve diğer özelliklerini temsil eden belli sayıda kullanıcıyla görüşmeler yapılarak sağlanır).
- Sistem için ön bir tasarım oluşturulması.
- Oluşturulan ön tasarımdan faydalanılarak yeni sistemin ilk prototipinin yapılması. Bu genellikle sistemin küçültülmüş ölçekli bir modelidir ve sistemin nihai haline ve sahip olacağı özelliklere ilişkin tahminleri gösterir.
- Aşağıdaki dört aşamalı süreç kullanılarak ikinci prototip oluşturulur.
  - İlk prototipin kuvvetli/zayıf tarafları ve riskleri açısından değerlendirilmesi,
  - İkinci prototipin ihtiyaçlarının tanımlanması,
  - İkinci prototipin planlanması ve tasarlanması,
  - İkinci prototipin yapımı ve test edilmesi.
- Kullanıcı tercihiyle bağlı olarak, risk çok büyük olduğu düşünüldüğünde tüm projenin iptal edilmesi [Risk faktörleri (işletim maliyetinin yanlış hesaplanması vb.), kullanıcı değerlendirmelerine göre tatminkârdan daha kötü seviyede bir nihai ürün elde edilmesine neden olarak geliştirme maliyetini çok büyük oranda yükseltebilir.



- Mevcut prototipin kendisinden önceki prototipte uygulanan değerlendirme yöntemiyle değerlendirilmesi ve mümkünse yukarıda açıklanan dört aşamalı süreç kullanılarak yeni bir prototip geliştirilmesi.
- Yukarıdaki aşamaların kullanıcının istediği nihai ürünü temsil eden ve müşteri tamini sağlayan bir prototip geliştirilinceye kadar tekrarlanması.
- Nihai sistemin, kabul gören son prototip kullanılarak yapılması.
- Nihai sistemin test ve değerlendirmesinin yapılması. Rutin bakım ve idame faaliyetleri büyük çaplı başarısızlıkların önlenmesi ve bitiş zamanının minimize edilmesi maksadıyla süreklilik arz eden tarzda gerçekleştirilir.

### **1.3. Ani Yazılım Geliştirme Modeli (RAD-Rapid application development)**

Ani Yazılım Geliştirme, Yapısal Sistem Analiz ve Tasarım Metodu ve diğer Şelale modelleri gibi 1970 ve 1980'lerde geliştirilmiş hızlı olmayan süreçlere tepki olarak ortaya çıkmıştır. Önceki metodolojilerde karşılan problemlerden biri, yazılımların geliştirilmesi çok uzun zaman aldığından yazılım tamamlanmadan önce ihtiyaçların değişmesi ve bunun sonucu olarak da yetersiz ya da kullanılmayan bir sistemin ortaya çıkmasıydı. Başka bir problem, sistematik ihtiyaç analizi safhasının tek başına bütün kritik ihtiyaçları tanımlamasının beklenmesiydi.

James Martin; Ani Yazılım Geliştirme yaklaşımını Brian Gallagher, Barry Boehm ve Scott Shultz'un fikirlerinden hareketle IBM'de 1980'lerde geliştirdi ve 1991'de bunu "Ani Yazılım Geliştirme" isimli bir kitapta yayınladı (Martin 1991).

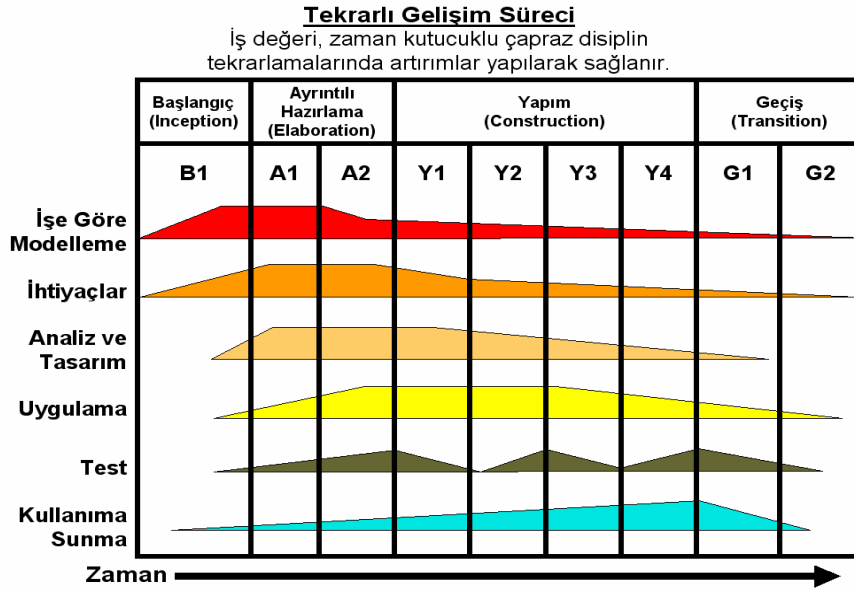
Martin'in metodolojisi tekrarlı geliştirme ve prototipler yapılması esasına dayanır. Son zamanlarda, bu terim ve onun kısaltması ağ uygulama çatıları (web application frameworks) ve diğer yazılım çatı (software frameworks) tipleri gibi yazılımın geliştirilmesinin hızlandırılmasını amaçlayan bir çok tekniği içine alan daha geniş bir bakış açısı ile kullanılmaya başlanmıştır. RAD yaklaşımı, yazılımın bakım ve idamesinin kolaylaştırılması ve daha hızlı geliştirme yapılabilmesi karşılığında performans ve işlevsellik arasında bir denge oluşturulmasını dikte etmektedir.

Ani yazılım geliştirme tekrarlı ve artırılmış bir süreç olması sebebiyle asla tatminkâr bir ürünle sonuçlanamayabilecek prototiplerin oluşturulmasına yönlendirebilir. Bu gibi başarısızlıklardan, yazılım geliştirme araçları güçlü, esnek ve doğru kullanım sağlayacak yapıda olduğu takdirde kaçınılabilir.

### **1.4. Tekrarlı (Aşamalı) Geliştirme Modeli (Iterative Development)**

Tekrarlı Geliştirme faaliyetleri, problem veya hatalı değerlendirmelerin bir felakete sebep olmadan önce, önemli olduğu kabul edilen hususları belirleyenlere

yardımcı olmak maksadıyla, önceleri küçük ancak sonraları yazılım projelerinin daha büyük dilimlerinin inşasına ilişkin reçetelerin sunulması olarak ifade edilebilir. Ne istediğini tanımlamayı bilmeyen kullanıcıların potansiyel tasarım hedeflerine ulaşılmasını sağlayan bu süreçler, ticari yazılım geliştirme şirketleri tarafından tercih edilmektedir.



**Şekil-E5.3: Tekrarlı Gelişim Süreci**

Tekrarlı gelişim, sağlanabilir iş değerini (sistem işlevi) tekrarlar şeklinde dilimlere ayırır. Her bir tekrarda bir işlev dilimi modelleme/ihtiyaçlardan başlayan ve test etme/kullanıma sunmaya kadar olan çapraz disiplinli bir çalışma ile sağlanır. Birleştirilmiş süreç grupları tekrarlama safhaları; başlangıç, ayrıntılı tanımlama, yapım ve geçiş olmak üzere dörde ayrılır. Başlangıç (Inception) safhası, projenin kapsamını, riskleri ve ihtiyaçları (işlevsel olan ve işlevsel olmayan) yüksek seviyede ancak yapılacak işin büyüklüğünü tespit edebilecek ayrıntıda tanımlar. Ayrıntılı tanımlama (Elaboration) en büyük riskleri azaltan ve işlevsel olmayan ihtiyaçları karşılayan bir çalışma mimarisi sağlar. Yapım (Construction); analiz, tasarım, uygulama ve işlevsel ihtiyaçların testi ile üretilen üretime hazır kodlarla giderek artan oranda mimariyi tamamlar. Geçiş (Transition), sistemin ürün işletim ortamına gönderilmesi aşamasıdır. Her bir safha, genellikle özellik kutucuklu olmaktan çok, zaman kutucuklu bir veya daha fazla tekrarlama alanına bölünür. Yazılım mimar ve analistleri, kendi iş ürün kayıtlarını dolu tutmak maksadıyla geliştiriciler ve test edicilerden bir tekrar ileride giderler.

### 1.5. Hızlı Yazılım Geliştirme (Agile Software Development)

Hızlı Yazılım Geliştirme faaliyetleri, tekrarlı geliştirme süreçlerinin oluşturduğu ana yapı üzerine kurulur ve bu süreçlere geleneksel yaklaşımlardan daha hafif ve daha insan-merkezli bakış açılarını ilave eder. Hızlı süreçler planlamadan ziyade geri beslemeleri kendi ilk kontrol mekanizması olarak kullanır. Geri beslemeler geliştirilen yazılımın belirli aralıklarla test edilmesi ve sürümleri oluşturularak elde edilir.

Araştırmaların şelale metodu kullanılarak önemli oranda verimlilik sağlanabileceğini göstermesi ilginçtir. Ağustos 2006'da VersionOne ve Agile Alliance tarafından yayınlanan ve 700'den fazla şirketin hızlı yazılım geliştirme yaklaşımı için aşağıdaki anket değerlerini sunan araştırması buna örnektir. Araştırma, Ağustos 2007'de 1700 kişinin katılımıyla tekrarlanmıştır.

Bazı araştırma sonuçları aşağıda sıralanmıştır.

- Artırılmış Verimlilik:
  - 2006: katılımcıların %87'si tarafından %10 veya daha yüksek iyileşme rapor edilmiştir.
  - 2007: katılımcıların %83'ü tarafından %10 veya daha yüksek iyileşme rapor edilmiştir.
  - 2006: katılımcıların %55'i tarafından %25 veya daha yüksek iyileşme rapor edilmiştir.
  - 2007: katılımcıların %55'i tarafından %25 veya daha yüksek iyileşme rapor edilmiştir.
- Yazılım Hatalarının Azaltılması:
  - 2006: katılımcıların %86'sı tarafından %10 veya daha yüksek iyileşme rapor edilmiştir.
  - 2007: katılımcıların %85'i tarafından %10 veya daha yüksek iyileşme rapor edilmiştir
  - 2006: katılımcıların %55'i tarafından %25 veya daha yüksek iyileşme rapor edilmiştir
  - 2007: katılımcıların %54'ü tarafından %25 veya daha yüksek iyileşme rapor edilmiştir
- Maliyet Düşürme:

▫ 2006: katılımcıların %63'ü tarafından %10 veya daha yüksek iyileşme rapor edilmiştir.

▫ 2007: katılımcıların %28'i tarafından %10 veya daha yüksek iyileşme rapor edilmiştir.

▫ 2006: katılımcıların %26'sı tarafından %25 veya daha yüksek iyileşme rapor edilmiştir.

▫ 2007: katılımcıların %28'i tarafından %25 veya daha yüksek iyileşme rapor edilmiştir.

Rapor edilen diğer ilginç iyileşmeler aşağıda ifade edilmiştir:

- Değişiklik önceliklerinin yönetiminde yetenek geliştirilmesi,
- Bilgi teknolojileri (IT) ile iş hedefleri arasında paralellik sağlanması,
- Artırılmış Tim Morali,
- Azaltılmış Proje Riskleri.

#### **1.6. Tekrarlı Geliştirme ile Hızlı Yazılım Geliştirme Yaklaşımlarının Karşılaştırılması**

Mevcut hızlı yazılım geliştirme yöntemi ile tekrarlı iyileştirme arasındaki ana farklılıklardan biri analiz ve ölçmenin çıkış noktası/girdi olarak kullanılmasıdır. Hızlı geliştirme yaklaşımı, yazılım ürününün kalitesinin ve süreçlerinin etkililiğinin belirlenmesinde destek sağlar ve geliştiricilerin özel ortamlardaki süreçler üzerinde çalışmalarına ve bu sayede iyileştirme ve ihtiyaca göre biçimlendirme yapmalarına imkân tanır. Bu ölçme ve analiz faaliyeti mevcut hızlı geliştirme metodlarına eklenebilir.

Çoğu yazılım sistemleri, aslında kullanıcıya geliştirme döngüsünün ilk safhalarında çalışan bir model sunulmasını öngören ihtiyaçların yer aldığı bu model kullanılarak geliştirilmektedir. Yeni özellikler eklendikçe, daha az hata içeren ve eskisinden daha fazla özelliğe sahip yeni bir versiyon denemek üzere hazırlanır. Yahoo Messenger, Azureus, Cyber Sitter, Net Meter, PC Security, Limewire, P2P, vb. bu tip modellere örnek olarak gösterilebilir.

#### **1.7. Test Kaynaklı Geliştirme Modeli (TDD-Test Driven Development)**

Bu modelde bir bölüm yazılmadan önce o bölüme ilişkin bir birim test hazırlanır. Bu, bir anlamda ilk olarak bölüme ilişkin araştırma yapılması ve müteakiben de TDD'nin kullandığı "testi yaz ve bölüm testi geçene kadar kodlamaya

devam et" modelini kullanacak şekilde yeterli detayda tanımlama yapılması olarak ifade edilebilir. Yazılımcıların hâlen erken kodlama ve hafif tasarım konusunda cesaretlendirildikleri hızlı geliştirme yaklaşımlarına zıt bir durum oluşturmasına rağmen, TDD'nin bahse konu faydalarının elde edilebilmesi için sorumlulukların en alt seviyeye kadar tam olarak tasarlanmasına gerek yoktur. Ancak bu, her ne kadar tekrarlı geliştirme süreci içinde kabul görse de üzerinde anlaşılmış, ihtiyaçları sabitlenmiş ve yenilenmeyen bir tasarım söz konusu olduğunda, olası değişim ihtiyaçlarının zorunlu kılacağı yeniden değerlendirme ve düzenleme faaliyetleri de dikkate alındığında TDD'yi etkisiz kılabilir.

### **1.8. Temiz Oda Yazılım Mühendisliği Modeli (Cleanroom Software Engineering Model)**

Yazılım güvenilirliğinin belgelenebilir seviyede olması için geliştirilmiş bir yazılım geliştirme sürecidir. Üretim hatalarının giderilmesinden çok önlenmesine odaklanmış bu süreç, ilk olarak Harlan Mills ve çok sayıda meslektaşı tarafından IBM'de geliştirildi (Dyer ve Linger 1987:19-25). Temiz Oda, elektronik endüstrisinde entegre devrelerin üretimi esnasında hatayla karşılaşılmasının önlenmesinde kullanılan temiz odalar (cleanrooms) terimi kullanılarak türetilmiştir. Bu modelin ilk kullanımı 1980'lerin sonunda olmuş ve uygulama projeleri askeri ortamda 1990'ların başında kullanılmaya başlanmıştır (Foreman 2005). Bu sürece ilişkin son çalışmada, temiz oda konsepti ile süreç içinde tanımlanan özelliklerin gerçekleştirdiği otomatik doğrulama yeteneklerinin kaynaştırılmasına çalışılmaktadır (Broadfoot ve Hopcroft 2005).

Temiz oda temel süreç prensipleri aşağıda açıklanmıştır.

- **Resmi Metotlara (Formal Methods) Dayalı Yazılım Geliştirilmesi:**

Temiz oda geliştirmeleri, bir yazılım ürününün özelliklerinin ve tasarımının belirlenmesinde kullanılır. Doğrulama, takımca gözden geçirilen özelliklerin tasarım tarafından doğru şekilde uygulanmasıyla sağlanır.

- **İstatistiksel Kalite Kontrolü (Statistical Quality Control)**

**Uygulanarak Artırımlı Uygulama Yapılması:** Temiz oda geliştirmeleri, içinde ürünün uygulanan işlevi adım adım yükseltecek şekilde artırımlar kullanmasını sağlayan tekrarlı bir yaklaşım (iterative approach) sunar. Her bir artırımın kalitesi, yazılımın arzu edilen şekilde geliştirildiğini doğrulayan önceden tanımlanmış standartlara göre ölçülür. Kalite standartlarının karşılanamaması, hâlihazırdaki artırım için testin durması ve tasarım safhasına geri dönülmesi ile sonuçlanır.

- **İstatistiksel Ağırlıklı Test Yapılması:** Temiz oda sürecinde yazılım testi, istatistiksel bir deney gibi gerçekleştirilir. Resmi özellikler bağlamında, yazılım girdi/çıkıktı değerlerini temsil eden bir alt küme seçilir ve test edilir. Bu örnek, müteakiben yazılımın güvenilirliğine ilişkin bir tahmin ve tahmin içinde de bir güven seviyesi üretilmesi maksadıyla istatistiksel olarak analiz edilir.

#### 1.9. Tekrarlı ve Artırmalı Geliştirme Modeli (Şekil-E5.4)

Bu süreç, şelale modelinin eksikliklerini ve cevap veremediği durumların ortadan kaldırılması maksadıyla geliştirilmiş döngüsel bir yazılım geliştirme sürecidir. Rational Firması Tekleştirilmiş Süreci (Rational Unified Process), Dinamik Sistem Geliştirme Metodu (Dynamic Systems Development Method), Yoğun Programlama (Extreme Programming) ve genellikle de Hızlı Yazılım Geliştirme (Agile Software Development) çatısının vazgeçilmez bir parçasıdır.

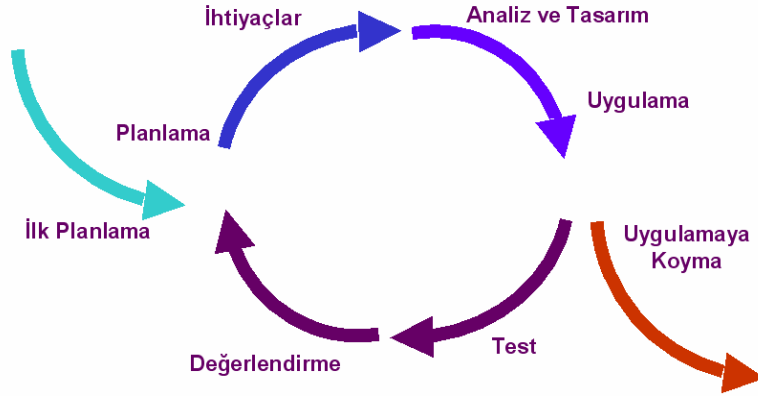
Artırmalı geliştirme, sistemin farklı parçalarının (modüller) farklı zamanlarda geliştirildiği ve geliştirmeyi müteakip ana sisteme ve birbirlerine entegre edildiği bir programlama ve safhalandırma stratejisidir. Her ikisi de yeniden çalışma stratejileri olan şelale ve tekrarlı geliştirmelere vurgu yapmaz, ihtiyaç duymaz ve bunlara ilişkin özellikleri içermez. Buna alternatif olarak kullanılacak bir yöntem de sistemin tamamının büyük parça bütünleştirilmesi (big bang integration) yöntemiyle gerçekleştirilmesidir. Tekrarlı geliştirme ise, sistemin parçalarının yenilenmesi ve iyileştirmesi maksadıyla zamanın bir kenara bırakıldığı yeniden çalışma programlama stratejisidir. Artırmalı geliştirmeyi önermez ama beraber iyi çalışır. Artırmalı ve tekrarlı geliştirme arasındaki farklardan biri; artırımdan elde edilen çıktıların müteakip iyileştirmelerde kullanılmaması ve testlerin veya kullanıcı geri beslemelerinin artırım plan veya özelliklerini yenileyici girdiler olarak değerlendirilmemesidir.

Bu iki yöntem, 1990'ların ortalarında birleştirilerek kullanılmaya başlandı. Tekleştirilmiş Sürecin (UP-Unified Process) ve Rational Firması Tekleştirilmiş Sürecinin (RUP-Rational Unified Process) yaratıcıları “tekrarlı geliştirme (iterative development)” ve “tekrar (iteration)” terimlerini genellikle artırmalı ve tekrarlı geliştirmenin her hangi bir kombinasyonu anlamında kullanmışlardır. Çoğu kimse bu nedenle tekrarlı (iterative) terimini artırmalı ve tekrarlı anlamında anlamakta ve kullanılmaktadırlar. Bazı proje timleri farkında olmadan sadece birini yapıp diğerini yapmadığında sorunla karşılaşmaktadırlar.

Tekrarlı iyileştirmenin altında yatan ana düşünce, sistemin önceki, atırımlı ve dağıtıma hazır versiyonlarının geliştirilmesi esnasında öğrenilen hususların sağlayacağı avantajlardan geliştiricinin faydalanmasına imkân tanıyan artırımlı bir yazılım sistemi geliştirilmesidir. Öğrenme, sistemin hem geliştirilmesi hem de kullanılması sayesinde sağlanır. Süreçteki anahtar adımlar, yazılım ihtiyaçları alt kümesinin basit bir uygulaması ile başlamak ve sistem tamamen uygulanabilir hale gelene kadar birbirini takip eden versiyonlar kullanılarak iyileştirme faaliyetlerini tekrarlamaktır. Her bir tekrarda, tasarım düzenlemeleri yapılır ve yeni işlevsel yetenekler eklenir.

Yöntem, başlatma adımı, tekrar adımı ve proje kontrol listesinden oluşur. Başlatma adımı sistemin temel versiyonunun oluşturulduğu adımdır. İlk adımla kullanıcının tepki vereceği bir ürün yaratılması hedeflenmektedir. Bu adımda; hazırlanacak olan temel yazılım, problemin anahtar hususlarına ilişkin bir örnek sunulmalı ve anlaşılması ve uygulanması kolay bir çözüm sağlanmalıdır. Tekrar sürecine rehberlik etmek maksadıyla, gerçekleştirilmesi gerekli bütün görevlerin kaydedildiği bir proje kontrol listesi oluşturulur. Listede, mevcut çözümün yeniden tasarlanan alanlar ve uygulanacak yeni özellikler gibi hususlar yer alır ve analiz safhasının bir sonucu olarak sürekli olarak değişir.

Tekrarlama, sistemin mevcut versiyonunun analizi ve proje kontrol listesindeki bir görevin yeniden tasarlanması ve uygulanması faaliyetidir. Her hangi bir tekrarlamanın tasarım ve uygulanması sayesinde, bulunan aşamadaki yeniden tasarımın desteklenmesi veya proje kontrol listesine yeni bir görev eklenerek basitliğin, açıklığın ve modülerliğin (tek başına çalışabilirlik) sağlanması amaçlanmaktadır. Tasarım detayının seviyesi tekrar yaklaşımında dile getirilmez. Hafif ağırlıklı bir tekrar projesinde, kaynak kodları sistem dokümantasyonunun ana kaynağı gibi görülebilmesine rağmen, kritik bir görev tekrar projesinde resmi bir Yazılım Tasarım Dokümanı (Software Design Document) kullanılabilir. Uygulanan tekrarın analizi, kullanıcıdan alınan geri beslemelere ve mevcut program analiz yeteneklerine dayandırılır. Yapısalılık, tek başına çalışabilirlik, kullanılabilirlik, güvenilirlik, etkinlik ve hedeflere ulaşıp ulaşılmadığı konularında her bir tekrar faaliyetinden sonra analiz yapılır ve her defasında proje kontrol listesi elde edilen analiz sonuçlarına göre yeniden düzenlenir.



**Şekil-E5.4:** Tekrarlı ve Artırımlı Geliştirme Modeli Döngüsü

Analiz ve uygulamaya rehberlik edecek hususlar aşağıda açıklanmıştır.

- Bir değişikliğin tasarımı, kodlanması ve testinde karşılaşılan her hangi bir zorluk ya da engel yeniden tasarım ve yeniden kodlama gerektirir.
- Değişiklikler bağımsız ve ulaşılması kolay modüllere kolayca uymalıdır. Uymadığında yeniden tasarlanması gerekir.
- Tablolarda yapılan değişikliklerin özellikle uygulanması kolay nitelikte olması sağlanmalıdır. Her hangi bir tablo değişikliği kolayca ve çabukça yapılamadığında yeniden tasarım yapılması zorunludur.
- Değişiklikler, tekrarlar gerçekleştikçe kolaylaşmalıdır. Bu sağlanamadığında, tasarım defosu veya çok sayıda yama (patches) yapılması gibi temel bir problem ortaya çıkar.
- Normal olarak bir veya iki tekrarlama için yamaların varlığına müsaade edilebilir. Yamalar bir uygulama esnasında yeniden tasarımdan kaçınmanın gerekli olduğu durumlarda kullanılmalıdır.
- Mevcut uygulamanın, proje hedeflerine ne kadar uyduğunun belirlenmesi maksadıyla sık sık analizler gerçekleştirilmelidir.
- Program analiz yetenekleri, kısmi uygulamaların analizine yardım edilebildiği durumlarda kullanılmalıdır.
- Kullanıcıların yazılıma verdikleri tepkiler istenmeli ve mevcut uygulamadaki yetersizliklerin görülmesi maksadıyla analiz edilmelidir.

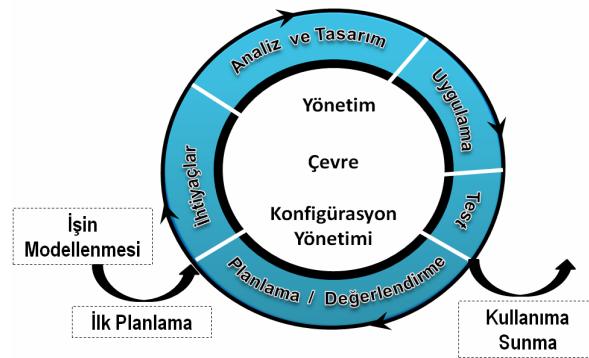


## 2. YAZILIM GELİŞTİRME METODOLOJİLERİ

Günümüzde yazılımlar için birçok proje yönetim metodolojisi ve tekniği geliştirilmiştir. Aşağıda en çok kullanılan metodoloji ve tekniklere yer verilmiş, anahtar özellikleri yanında avantaj ve dezavantajları üzerinde durulmuş, açıklanan diğer metodolojilerle birlikte kullanılabilirliğine ilişkin değerlendirmeler sunulmuştur. Kalın ve ince metodolojiler açısından değerlendirme yapılarak, kalın metodolojiler kapsamında RUP, ince metodolojiler kapsamında ise SSADM, PRINCE2, XP, SCRUM and Crystal Clear metodolojilerine ilişkin açıklamalara yer verilmiştir. Her bir metodoloji açıklanırken yönetim ve iş bakış açılarına odaklanılmıştır.

### 2.1. Rational Firması Tekleştirilmiş/Birleştirilmiş Süreci (Rational Unified Process, RUP)

RUP yazılım tasarım metodolojisi olarak "The Rational Software Company" firması tarafından ortaya konmuştur. RUP, bütün yazılım tasarımının yüksek detayda tanımlandığı kalın bir metodolojidir. Özellikle büyük kapsamlı yazılım projelerinde uygulanabilen, tasarımı ve dokümantasyonu Tekleştirilmiş/Birleştirilmiş Modelleme Dili (UML) ile yapılmış olan RUP, ihtiyaç duyulan hemen tüm proje adımlarını içermekte ve bu şekliyle de tek bir proje firmasının özel ihtiyaçlarına uyarlanabilir bir yapı sunmaktadır. RUP ve SSADM (Bkz. 3.2) gibi metodolojiler arasındaki en önemli farklıklardan biri RUP'un yazılım geliştirme şelale yaklaşımını kullanmamasıdır. İhtiyaçların ortaya konması, analiz, tasarım, uygulama, bütünleştirme (integration) ve test aşamaları kesin veya zorunlu bir sıra izlemektedir. RUP içinde yazılımın tasarım ve kodlanmasında her aşamada tekrarlar içeren aşamalı bir yaklaşım kullanılmaktadır. Kat edilen her bir aşama çoğu yazılım geliştirme disiplini (ihtiyaç tanımı, analiz, tasarım, uygulama, test vb.) içermektedir. Şekil-E5.5'te bir RUP proje aşamasının grafik gösterimi sunulmuştur (Kruchten 2004).



Şekil-E5.5: Tekrarlı Yazılım Geliştirme Yaklaşımı (Scharbert 2009)

- **Uygulama Alanı**

RUP'un modüler yapısı dikkate alındığında her türlü yazılım projesinde hatta yazılım projesi olmayan projelerde de kullanılabileceği ifade edilebilir. Ancak, RUP metodolojisi karmaşık yapısı nedeniyle çoğunlukla büyük kapsamlı yazılım projelerinde kullanılmaktadır.

- **Avantajları**

Sadece yazılım proje ömür devri sonunda değil, her bir aşamada test yapılmasını öngören aşamalı yaklaşım muhtemel problemlerin önceden fark edilmesini ve daha kolay ve ucuz çözüm getirilmesi bakımından yüksek etkinlik sağlamaktadır.

Şelale yaklaşımı kullanıldığında, yazılım programcılar, kodlama ve tasarımı bütünleştirmeye (integration) başlamadan önce, tasarım aşamasının tamamlanmasını beklemek zorunda kalırlar. Oysa aşamalı yaklaşım kapsamında bütünleştirme ve uygulama sadece projenin sonunda gerçekleştirilmediğinden proje ekibine zaman kazandırmakta ve bu bağlamda gerçekleştirilecek tasarımlama ve kodlama ile bahse konu problemi ortadan kaldırmaktadır.

Yazılım ihtiyaçlarında meydana gelebilecek değişikliklerin yönetilmesi RUP ile daha kolaydır. Küçük projeler haricinde bütün yazılım ihtiyaçlarının daha projenin başında tanımlanması neredeyse imkânsızdır. Gerek yazılım müşterisi (Kullanıcı) ve de gerekse proje ekibi için yazılımın son halinin neye benzeyeceğinin belirlenebilmesi genellikle birden fazla aşamanın gerçekleştirilmesinden sonra ortaya çıkmaktadır. Aşamalı geliştirme, gözden kaçan hususların ortaya çıkmasına veya müşteri tatminini engelleyebilecek ihtiyaçların değişimi "kırılması (creeping)" sürecini kolaylaştırır.

Kendisi de bir yazılım olan RUP, internet bağlantılı elektronik bir formla dağıtılmaktadır. Proje ekibinin RUP ile ilgili faaliyetleri gerçekleştirirken bu nedenle bilgisayarlarının başından kalkmalarına gerek yoktur. Yazılım geliştirme metodolojisi kapsamındaki her türlü bilgi bu sayede proje ekibinin parmak uçlarındadır. Bundan daha önemlisi, tüm proje ekibinin aynı RUP sürümünü kullanıp kullanmadıklarının kolayca belirlenmesi de bu sayede sağlanmaktadır. RUP, UML kullanılarak nesne tabanlı şekilde tasarımı ve dokümantasyonu sağlandığından, tek bir proje ya da organizasyonun özel ihtiyaçlarına göre şekillendirilmesi daha kolaydır.

- **Dezavantajları**

RUP ticari bir ürün olduğundan açık ve kullanımı serbest standartları yoktur. RUP kullanılmadan önce kullanım haklarının IBM firmasından elektronik yazılım ve dokümantasyon paketi şeklinde satın alınması gereklidir (IBM'in internet sitesinden deneme sürümü indirilebilmektedir). RUP'a sadece elektronik form şeklinde ulaşılabildiğinden bazen kullanımı sınırlı olabilmektedir.

RUP, önceden de açıklandığı gibi tüm yazılım tasarım sürecini yüksek detayda tanımladığından hem proje yöneticileri hem de proje ekibi tarafından anlaşılması zor çok karmaşık bir metodolojidir. Bu nedenle, küçük projelerde kullanılmaya uygun değildir.

RUP'un bir yazılım geliştirme metodolojisi olarak kullanılmaya başlanması zordur. Projede yer alan tüm personel tarafından RUP kullanımının öğrenilmesi zorunludur (Projelerde RUP uygulamaları konusunda detaylı bilgi için (<http://www-106.ibm.com/developerworks/rational/library/4243.html>) (Kroll 2009).

- **Diğer Metodolojilerle Birlikte Kullanımı**

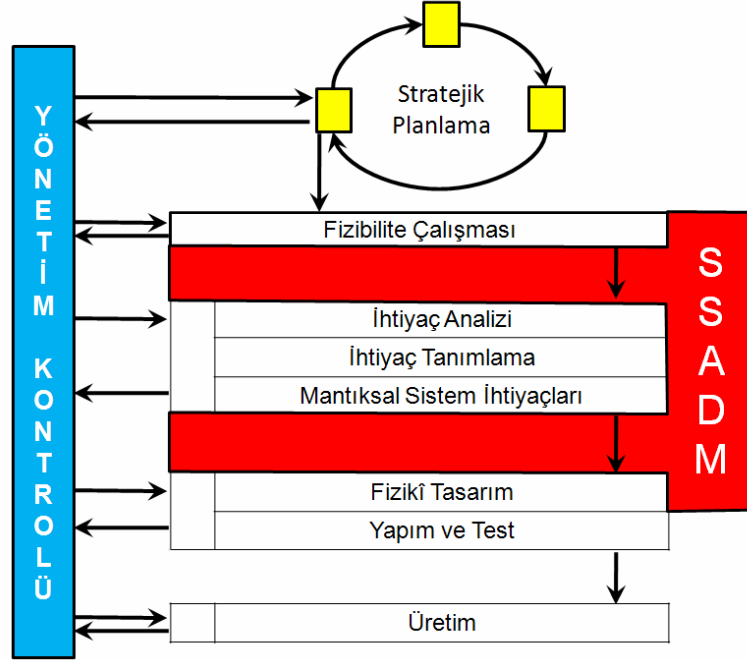
RUP kalın bir metodoloji olduğundan diğer metodolojilerle birlikte kullanımı her zaman mümkün olamamaktadır. "Crystal Clear" metodolojisiyle birlikte kullanımı için bir yol mevcuttur. (Bkz. 2.6)

## **2.2. Yapısal Sistem Analizi ve Tasarımı Metodolojisi (SSADM-Structural System Analysis & Design Methodology)**

SSADM, en çok İngiltere'de kullanılan bilgisayar uygulama geliştirme metodudur. Kullanımı PRINCE (Bkz. 2.3) gibi, hükümet yazılım projelerinde bir ihtiyaç olarak belirtilmektedir. Günümüzde Avrupa'da kamu sektöründe giderek artan oranda kullanılmaktadır. SSADM, 1981 yılında ilk kullanıma sunulduğundan bu yana hükümetler tarafından yazılım projelerinde kullanılmaktadır. İngiliz Merkezi Yazılım ve İletişim Ajansı (CCTA) tarafından bakanlıklar arasında geliştirilecek çok sayıda ve farklı şekillerdeki bilgi teknolojilerinin (IT) standart hale getirilmesi kapsamında onaylanmıştır. CCTA, bu onay öncesinde Learmonth & Burchett Management Systems firmasıyla birçok yaklaşımın uygulanabilirliği konusunda araştırma yapmıştır (Malcolm 2006).

1981'den sonra SSADM geliştirilerek/iyileştirilerek 1990 yılında 4'üncü sürümü kullanıma sunulmuştur. SSADM açık bir standart olup, endüstride ve onun için destek, eğitim ve Bilgisayar Destekli Yazılım Mühendisliği (CASE) araçları sağlayan firmalar tarafından ücretsiz olarak kullanılmaktadır.

SSADM, şelale yaklaşımıyla sistemin geliştirildiği ve her biri bir sonraki adıma girdi olacak bir dizi adımların kullanılmasını öngörür (Bkz. Şekil-E5.6).



Şekil-E5.6: SSADM Süreç Modeli (Godland ve Riha 2009)

SSADM adımları (Hutchings 2009);

- **Fizibilite Çalışması:**

Fizibilite çalışması bir sistemin iş ihtiyaçlarını maliyet etkin şekilde karşılayıp karşılamadığını belirlemek amacıyla bir iş alanının üst seviyede analizinin gerçekleştirildiği tek bir süreçtir. Bu çalışmada, yüksek seviye Mantıksal Veri Yapısının (Logical Data Structure - LDS) ve Veri Akış Grafiğinin (Data Flow Diagram - DFD) gözden geçirilmesi sağlanır. Bu aşamada, DFD mevcut sistemi temsil etmekle birlikte LDS daha tam olarak tamamlanmamış ve tam olarak çözümlenmemiş ilişkiler içerebilir.

- **İhtiyaç Analizi:**

- **Mevcut Ortamın Araştırılması:** Bu aşamada, sistem ihtiyaçları tanımlanır ve mevcut iş ortamı, uygulanacak süreçler ve veri yapıları bakımından modellenir. Burada, DFD ve LDS'ler mevcut sistemin detaylı mantıksal modellerinin oluşturulmasında kullanılır.

▫ **İş Sistemi Seçenekleri (Business System Options - BSO):**

Bu aşmada altı iş sistem seçeneği oluşturulur ve kullanıcıya sunulur. Sonuçta da bu seçeneklerden birine göre çalışmaya başlanır ve seçenek düzenlenir. DFD ve LDS'ler her bir sistem seçeneğinin ve seçilen seçeneğin desteklenmesi için hazırlanır. Önceki aşamadan bu aşamaya geçiş, SSADM'in mevcut sistemin mantıksal modelinden ihtiyaç duyulan sistemin mantıksal modeline geçişte kilit rol oynayan bölümdür. Bu da yeni ve/veya değişen ihtiyaçların karşılanması kapsamında DFD ve LDS'lerin yenilenmesi/düzeltilmesi anlamına gelmektedir.

▫ **İhtiyaç Özelliklerinin Belirlenmesi:** İhtiyaç özelliklerinin

belirlenmesi, ihtiyaç duyulan veri ve süreçlerin tanımlanması maksadıyla detaylı işlevsel ve işlevsel olmayan ihtiyaçların belirlendiği ve yeni tekniklerin kullanılmaya başlandığı ihtiyaç analizinde gerçekleştirilen işlemlerin detaylandırılması/geliştirilmesi kapsamındaki faaliyetleri içeren tek bir süreçtir.

▫ **Mantıksal Sistem Özelliklerinin Belirlenmesi:**

◆ **Teknik Sistem Seçenekleri:** Bu aşamada, geliştirme

ve uygulama ortamını belirleyen altı teknik seçenek belirlenir ve biri bunlardan seçilir.

◆ **Mantıksal Tasarım:** Bu aşamada, güncelleme/erişim

süreçlerinin ve sistem diyaloglarının (menüler vb.) mantıksal tasarımı gerçekleştirilir.

◆ **Fiziksel Tasarım:** Fiziksel tasarım, mantıksal ve teknik

sistem özelliklerinin fiziksel veri tabanı tasarımının ve yazılım özelliklerinin oluşturulmasında kullanıldığı tek bir süreçtir.

- SSADM, üç anahtar teknik kullanılarak biçimlendirilir:

▫ **Mantıksal Veri Modellemesi (Logical Data Modeling-LDM):**

Bir iş bilgi sisteminin veri ihtiyaçlarını tanımlayan, modelleyen ve dokümanlarını hazırlayan bir süreçtir. LDM, LDS ve ilgili dokümanlardan oluşur. LDS'ler, hakkında bilgi toplanacak/kaydedilecek birimleri (entities) ve birimler arasındaki ilişkileri temsil eder.

▫ **Veri Akış Modellemesi (Data Flow Modelling-DFM):** Bir iş

bilgi sistemi içinde verilerin nasıl dolaştığının ortaya konmasını sağlayan ve bunun tanımlama, modelleme ve dokümantasyonunun yapıldığı süreçtir. DFM, bir araya getirilmiş ve uygun dokümanlarla desteklenmiş Veri akış Grafiklerinden (Data Flow Diagram-DFD) oluşur. DFD'ler verilerin birinden diğerine aktarıldığı

faaliyetleri/süreçleri, veri depolarını, sisteme dışarıdan veri gönderen ve/veya sistemden veri alan dış birimleri (entities) ve son olarak da veri dolaşımını gösteren grafiklerdir.

▫ **Birim/Faaliyet Modelleme (Entity/Event Modeling, EM):** Her bir birimi ve faaliyetlerin gerçekleşme sırasını etkileyen iş faaliyetlerini tanımlayan, modelleyen ve dokümantasyonunu sağlayan süreçtir. EM içinde her bir birim (entity) için bir Birim (Entity) Hayat Döngüsü ve uygun destek dokümantasyonları yer alır.

• **Kullanım Alanları**

SSADM İngiltere’de bakanlıklar arasında geliştirilen çok sayıda ve farklı yapıda bilgi teknolojisi (IT) projelerinin standart hale getirilmesi için geliştirilmiştir. Günümüzde SSADM Sürüm 4 her türlü analizde ve sistem geliştirme tasarım aşamalarında kullanılmaktadır. SSADM, her büyüklükteki [küçük (1-2 personel, yılda bir kişi), orta (4-10 personel, yılda 1-20 personel) ve büyük (10’dan fazla, yılda 20’den fazla)] projelerde uygulanabilmektedir. Bunun da ötesinde, SSADM yeni projelerin geliştirilmesinde de mevcut sistemlerin idame ettirilmesinde de kullanılabilir (Godland ve Riha 2009).

• **Avantajları**

▫ SSADM açık bir standart olup, endüstride ve onun için destek, eğitim ve Bilgisayar Destekli Yazılım Mühendisliği (CASE) araçları sağlayan firmalar tarafından ücretsiz olarak kullanılabilir.

▫ SSADM uygulama geliştirme projelerini modüllere, adımlara ve görevlere böler ve projelerin proje yönetimine uygun şekilde tanımlanması bakımından bir çatı oluşturur.

▫ SSADM’in hedefleri:

- ◆ Proje yönetim ve kontrolünü geliştirmek,
- ◆ Tecrübeli ve tecrübesiz proje ekibi personelinin daha etkin kullanımını sağlamak,
- ◆ Daha kaliteli sistemler geliştirmek,
- ◆ Projelerin CASE gibi bilgisayar tabanlı araçlarla desteklenmesini sağlamak,
- ◆ Proje paydaşları arasında iyi bir iletişim alt yapısı oluşturmaktır.

- SSADM ilk ihtiyaların yanlış anlaşılması ve sistem işlevlerinin yetersiz analiz ve tasarım teknikleri kullanımı sonucu ihtiyalardan farklı olması olasılığını azaltmaktadır.

- **Dezavantajları**

- SSADM, ihtiyaç belirleme sürecini ihtiyaların yanlış anlaşılması olasılığının azaltılması bakımından resmileştirdiğinden ve en iyi uygulama tekniklerini analiz ve tasarım sürecinde kullandığından yapılaşmış sıkı bir süreç öngörür.

- İlk ihtiyaların yanlış anlaşılması ve sistem işlevlerinin yetersiz analiz ve tasarım teknikleri kullanımı sonucu ihtiyalardan farklı olması olasılığını azaltmasına rağmen SSADM, ihtiyaların projenin geliştirilmesi esnasında değişmeyeceğini farz ve kabul eder.

- Sıkı sıkıya SSADM adımlarının takip edilmesi zaman alır ve hazırlama ve dağıtım aşamalarında dikkate değer miktarda gecikmeler yaşanabilir. Bunun sonucu olarak da her ne kadar daha fazla zaman harcadığında sistem ihtiyaç özellikleri daha iyi karşılanırsa da dağıtım anında iş ihtiyaları karşılanamayabilir.

- **Diğer Metodolojilerle Birlikte Kullanımı**

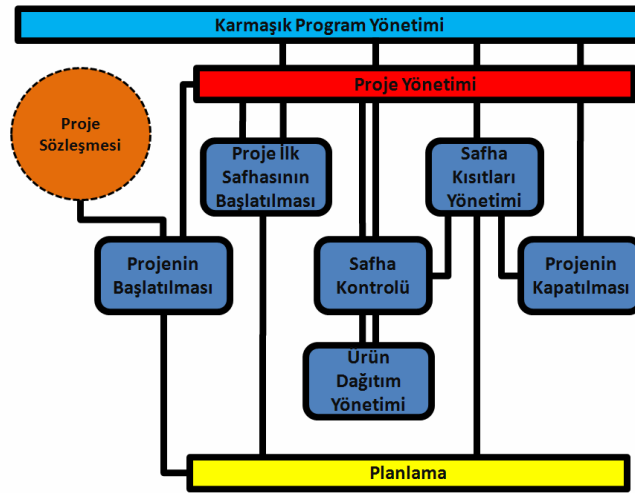
SSADM, fizibilite çalışma aşamasından fiziksel tasarımın üretimine kadar bir sistemin ömür devrine ilişkin tüm hususları dikkate alır. Genellikle, proje yönetimini daha kapsamlı ele alan diğer proje yönetim metot ve teknikleriyle bağlantılı olarak kullanılır. Bu nedenle de SSADM bu dokümanda yer alan hemen her metot ya da proje tekniğı ile birleştirilerek kullanılabilir (www.techtarget.com 2009).

### **2.3. PRINCE2**

Kontrol Edilebilir Ortamda Proje Yönetimi (PRINCE) projelerin düzenlenmesi, yönetimi ve kontrolü konularını ele alan bir proje yönetim metodolojisidir. Bir projede genellikle belirgin bir başlangı, gelişme ve sonuç yanında belirgin organizasyonel yapı ve tanımlanmış hedefler vardır. PRINCE gibi bir yönetim metodolojisi projenin başarısı (zamanında bitirilmesi, belirlenen büte dâhilinde gerçekleştirilmesi ve kullanıcının istediğı şekilde geliştirilmiş olması) konusunda emin olunmasını sağlar.

PRINCE, ilk olarak şimdilerde Hükümet Ticaret Ofisi tarafından İngiliz Hükümeti bilgi teknolojileri (IT) proje yönetim standardı olarak 1989'da geliştirilmiştir. Geliştirilmesinden bu yana kamu ve özel sektör tarafından büyük oranda kullanılmaya başlanmış ve günümüzde İngiltere'de proje yönetimi için vazgeçilmez bir standart haline almıştır. Bilgi teknolojileri projeleri için geliştirilmiş olmasına rağmen başka projelerde de kullanılabilir. Metodolojinin son sürümü olan PRINCE2, mevcut kullanıcıların ihtiyaçlarının karşılanması ve sanal bir en iyi uygulama yaklaşımı ile tüm projelerin yönetimi için eski metodolojinin yeniden düzenlenmesi için tasarlanmıştır.

PRINCE2, tüm proje çeşitlerinin yönetiminde projeye göre kolayca şekillendirilebilen ve ölçülebilir süreç bazlı bir yaklaşım öngörmektedir. Her süreç icra edilecek faaliyetler ve ulaşılmaları planlanan hedeflerle yanında anahtar girdi ve çıktıları ile birlikte tanımlanmıştır. Metodoloji bir projenin proje boyunca ilerleme durumu takibini ve etkin kaynak kontrolünü sağlayan yönetilebilir aşamalara nasıl bölünebileceğini tanımlamaktadır. Proje yönetimi değişik rol ve sorumlulukların tüm hatlarıyla tanımlanabilir ve projenin büyüklüğü ve karmaşıklığına ve organizasyonun esaslarına uyması maksadıyla uyarlanabilir yapıda olması sağlanır (<http://www.ogc.gov.uk/prince>).



Şekil-E5.7: PRINCE2 Süreç Modeli (<http://www.ogc.gov.uk/prince>).

PRINCE2, bir projenin başarıyla yürütülebilmesi için gerekli olan sekiz süreci ortaya koyar (Bkz. Şekil-E5.7). Bunlar (Spence 2009);

- **Projeye Başlanması:** Projenin açık ve anlaşılır şekilde başladığından emin olunması maksadıyla, bu süreç daha proje gerçek anlamda



başlamadan uygulanır. Tüm karar vericiler, bu süreçte bir araya gelir ve bir Proje Yöneticisi (Project Manager-PM) tayin ederler. Bunun yanında projeye başlanma nedenleri açıklanır ve projenin nasıl yürütüleceğine karar verilir. Bütün bu bilgiler “Proje Özeti” nde bir araya getirilir.

- **Projenin Başlatılması:** Bir proje “Projenin Yönlendirilmesi” sürecinde onaylanmadan önce hedefleri karşıladığından emin olunabilmesi bakımından dikkatle planlanmalıdır. Detaylı maliyet tahminleri, ihtiyaç duyulan zaman ve diğer kaynaklar değerlendirilmeli ve Proje Kurulu (PB) tarafından onaylanmak üzere Proje Yöneticisince Proje Başlatma Dokümanında (Project Initiation Document-PID) bir araya getirilir.

- **Projenin Yönetilmesi:** Proje Özeti hazırlandıktan ve PID oluşturulduktan sonra Proje Kurulu (Project Board-PB) adı verilen üst düzey yöneticiler tarafından onaylanması gereklidir. Projenin geri kalan bölümünde PB, projenin başarısı konusunda tüm sorumlulukları üstüne alırken PM ise günlük tüm faaliyetlerden sorumludur. Günlük raporlarla PB'nin proje ilerleme durumu konusunda bilgilendirilmesini sağlar.

- **Safha Kontrolü:** PRINCE2'nin avantajlarından biri projenin yönetilebilir olduğundan ve kontrol edilebildiğinden emin olunmasını sağlayan yönetilebilir aşamalara bölünmesidir. Ne kadar aşamanın kullanılacağı risk seviyesine ve projenin büyüklüğüne bağlıdır. PRINCE2'de her bir proje aşaması bir sonraki aşamadan önce tamamlanır ve her bir yeni aşama kendisinden önceki aşamada planlanır. Aşama Planları da projenin belirlenen bütçe sınırları içinde kalması ve istenen hedeflere ulaşılması bakımından PB tarafından onaylanır.

- **Safha Sınırlarının Yönetimi:** Bu süreçte mevcut aşama gözden geçirilir ve müteakip aşamaya hazırlık yapılır. PM, projenin iş hedeflerine ulaşabilme derecesine ilişkin tahminlerini ve iş durumu, proje planı, riskler ve dikkate değer diğer konularda meydana gelen her türlü değişiklikleri PB'ye iletir. Bir projede açık aşama sınırları bulunduğu kolayca kontrol edilebilir ve PB mevcut aşama ve planlanan müteakip aşama konusunda tatmin olmadıkça projenin ilerlemesine müsaade etmeyecek şekilde yönetilebilir.

- **Planlama:** Her bir proje planı, aşama planı ve ekip planı anahtar planlama durumlarını (üretilen ürünün ne olduğu, ürünün üretimi için gerekli faaliyetler, maliyet ve zamanı da içeren tahmin edilebilir kaynaklar, faaliyetlerin programlanması ve risklerin analizi) dikkate almalıdır. PRINCE2 planlama sürecini

takiben bütün bu noktalar bilinçli ve mantıksal bir sıra dâhilinde icra edilir. Uyumluluğun garanti edilmesi, planların karşılaştırılmasını ve planlama sürecinin yürütülmesini sağlar.

- **Ürün Dağıtım Yönetimi:** Bir PRINCE2 projesinin hedefi ürünün dağıtılmasıdır. Aslında PRINCE2' de üretilen her şey bir ürün olarak kabul görür. PM genellikle ürünü yaratan kişi değildir. Üçüncü taraf hizmet sağlayıcısı ve/veya meslektaşları işin bir kısmını veya tamamını yaparlar. Yapılacak işi tanımlayarak doğru zamanda doğru ürünün hizmet sağlayıcı tarafından üretilmesinin sağlanması PM'nin sorumluluğudur.

- **Projenin Sonlandırılması:** Projenin sonunda, ürünler dağıtıldıktan sonra PB tarafından onayı müteakip proje sonlandırılır. PM, Proje Sonrası Gözden Geçirmesi adı verilen ve proje çıktısının değerlendirilmesi için yapılması gerekenlerin ne olduğunu açıklayıcı işlemleri planlar. Kontrollü bir kapanış, gösterilebilir son PRINCE2 proje faaliyetinden sonra gerçekleşir. Alınan dersler kayıt edilir, kaynaklar dağıtılır ve Proje Sonrası Gözden Geçirme Planı (Post-Project Review plan-PPRP) oluşturulur.

- PRINCE2 için temel olan anahtar konseptler:
  - **Kontrol:** Projenin kontrol edilebilmesi projenin başarısında anahtar rol oynar. Bu nedenle, PRINCE2 projeyi yönetilmesi kolay aşamalara böler.
  - **Kalite:** Ürün ya da hizmetin proje planlandığında tanımlanarak ve üzerinde uzlaşılarak müşteri kalite beklentilerinden kesin olarak emin olunması gereklidir.

- **Planlama:** PRINCE2'de proje başlasa bile planlama bitmez. Sekiz PRINCE2 sürecinden son süreç de dâhil sadece bir tanesinde planlama faaliyeti yer almaz.

- **Alınan Dersler:** Her proje yürütüldüğünde bir şeyler öğrenilir. Tüm dersler, hatalar, fikirler ya da başarılar Alınan Dersler Kütüğüne kaydedilir. Projenin sonunda bu kayıtlar, organizasyonda çalışan diğer personelinde faydalanması maksadıyla Alınan Dersler Raporuna dönüştürülür.

- **Uygulama Alanları**

PRINCE2, değişik birçok endüstri alanı ve uygulama altyapısından gelen geri besleme ve en iyi uygulamaları özetlemektedir. PRINCE2, İngiliz Milli Sağlık Hizmetleri tarafından tercih edilen metodoloji olarak kabul edilmiş ve dünyada

birçok ülke tarafından da standart proje yönetim metodolojisi olarak kabul edilmesi beklenmektedir. PRINCE2, hem kamu hem de özel sektör tarafından kullanılan birkaç hükümet standardından biridir. PRINCE2'yi proje yönetim metodolojisi olarak kullanan organizasyonlardan bazıları; İngiliz Emniyet Genel Müdürlüğü, Rolls Royce, İngiliz Tıp Kurumu, Norwich Birliği, İngiliz Adalet Bakanlığı ve Londra Underground'dur (Spence 2009).

- **Avantajları**

PRINCE2 proje yönetimine standart bir yaklaşım getiren yapısal bir metodolojidir. Proje yönetiminde geçerliliği ispatlanmış ve tesis edilmiş en iyi uygulamaları bir araya getirmektedir. Genel kabul görmüş olan bu metodoloji, proje paydaşlarının ortak bir dil kullanmasını sağlar ve eğitici bir platform sağlar. PRINCE2 sağladığı anahtar konseptler yanında ilave bazı avantajlar da sunmaktadır.

PRINCE2 projelerin:

- Kontrollü ve organize bir başlangıç, gelişme ve sonuca sahip olmasını,
- İş Durumları ve plan kapsamında ilerleme durumunun rutin olarak gözden geçirilmesini,
- Esnek karar noktaları bulunmasını,
- Plandan olası sapmalarının otomatik yönetim kontrolünü,
- Yürütülmesi esnasında proje yönetim ekibi ile müşterilerin doğru zamanda ve doğru yerde müdahil olmalarını,
- Organizasyonun geri kalanı, proje ve proje yönetim ekibi arasında oldukça iyi iletişim kanallarına sahip olmasını sağlar.

PRINCE2 el kitabında test metodlarına ilişkin bir bölüm bulunmadığından her türlü test metodunun kısıtsız olarak kullanılabilmesi söylenebilir ancak, PID'nin bir bölümünü oluşturan proje kalite planı hazırlanırken bu konuda dikkatli olunmalıdır.

- **Dezavantajları**

- PRINCE2 projesinde çalışan herkes, oyunu kuralına göre oynamak manasında PRINCE2'nin hemen tüm özelliğini kullanabiliyor olmalıdır. Ancak, PRINCE2 eğitiminin herkese verilmesi oldukça pahalı olduğundan genellikle bu husus göz ardı edilmektedir.

- Sadece PB'yi projenin durumu konusunda bilgilendirmekten sorumlu olan PM'lerin işler sarp sardığında başkalarını suçlamasının önüne geçilmesi için PRINCE2'de birçok doküman ve listenin yazılması zorunluluğu vardır. Benzer şekilde bir proje ekibi olumsuz bir durumda diğer proje ekiplerini suçlayabilmektedir.

- PRINCE2 projesinin bölünmesi PM gibi sorumlu personelin proje konusunda yeterli bilgiye sahip olmasını engelleyebilir. İş sadece PB'nin bilgilendirilmesi ve yönlendirilmesi olan pahalı PM'nin kullanımı faydalı değildir.

- **Diğer Metodolojilerle Birlikte Kullanımı**

PRINCE2, tastamam bir proje yönetim metodolojisidir ve projenin en başından en sonuna kadar olan tüm hususları içerir. Bu nedenle de bu metodolojinin başka bir metodoloji ya da proje tekniği ile birlikte kullanılması söz konusu değildir. Genellikle SSADM'in eksiklik ve zaafalarının giderilmesi amacıyla bu metodolojiye ek olarak kullanılmaktadır.

#### **2.4. Yoğun Programlama (eXtreme Programming - XP)**

1990'lardaki çoğu yazılım geliştirme projesi; içte nesne tabanlı programlamanın (Object-Oriented Programming - OOP) işlem sıralı programlamanın (procedural programming) yerini alması ve dışta ise internetin hızlı yükselişinin daha hızlı pazarlama ve şirket büyümesini rekabetçi bir iş faktörü olarak ortaya çıkarması olarak ifade edilebilecek iki büyük etki altında kalmıştır. İhtiyaçların hızlı değişimi, daha kısa ürün ömür devirlerinin ortaya çıkmasına ve bu değişikliklerin yazılım geliştirmede geleneksel metotlarla karşılanamaz hale gelmesine neden olmuştur.

Yoğun Programlama (XP), Kent Beck tarafından maaş sistemini araştırma nesnesi olarak kabul ederek nesne tabanlı teknolojilerin kullanımında en iyi yöntemlerin belirlenmesi amacıyla başlatılan Chrysler Kapsamlı Ödeme Sistemi [Crysler Comprehensive Compensation (C3) System] projesi üzerinde çalışırken geliştirilmiştir. Yazılım dili olarak "Smalltalk" ve veri erişim katmanı (data access layer) olarak da "GemStone" paket programlarının kullanıldığı proje, Kent Beck'e performans iyileştirmesi (performance tuning) amacıyla ihale edilmiş ve Mart 1996'da C3 proje ekip lideri olarak görevlendirilmiştir. İlk olarak projede kullanılan geliştirme metodunun iyileştirilmesine başlayan Beck 1999'da bahse konu metodu konu alan bir kitap yazmıştır (Extreme Programming Explained). 2000 yılında Crysler projeden vazgeçmesine ve proje başarısız olmasına rağmen XP yazılım mühendisliği alanında kullanılan bir metot haline almıştır.

XP, aşağıda açıklanacak değerleri oluşturan ve teşvik eden günlük çalışma setlerini açıklayan bir yazılım mühendisliği metodolojisi olup hızlı yazılım geliştirme yönteminin özel bir şeklidir. XP taraftarları, geleneksel yazılım mühendisliği tekniklerini sözde “uç (extreme)” seviyelere taşıyan bu pratiklerin uygulanmasıyla, daha kaliteli bir yazılım oluşturulması yanında müşteri ihtiyaçlarına daha hızlı cevap veren geliştirme süreçlerinin kullanılmasına öncülük edildiğine inanmaktadırlar.

XP ve hızlı yazılım geliştirme metodolojisi taraftarları genelde süregelen ihtiyaç değişikliklerini yazılım geliştirme projelerinin doğal, kaçınılmaz ve arzu edilen/beklenen bir yanı olarak değerlendirirler. Bütün ihtiyaçların proje başlangıcında tanımlanması ve ihtiyaç değişikliklerinin kontrolü için gayret sarfedilmesinden, projenin herhangi bir anında ortaya çıkan ihtiyaç değişikliklerine uyum sağlamanın daha gerçekçi ve daha iyi bir yaklaşım olduğuna inanırlar.

Ancak XP’de; değişken ihtiyaçlar, yazılı hale getirilmemiş kullanıcı çatışmaları ve ana tasarım özellik ve dokümantasyonu eksikliği içeren daha doküman bazlı metodolojilere kıyasla birçok potansiyel ürün geri çekmeleri kaydedilmiştir.

- **XP’nin Amacı**

Beck’in Açıklamalı XP (XP Explained) isimli kitabı XP’yi:

- İnsanlık ile verimliliğin arasının bulunması için bir girişim,
- Bir sosyal değişim mekanizması,
- Gelişime giden bir yol,
- Bir geliştirme stili
- Bir yazılım geliştirme disiplini olarak tanımlamaktadır.

XP’nin ana amacı, değişim maliyetlerinin azaltılmasıdır. Geleneksel sistem geliştirme metotlarında (SSADM gibi), sistem ihtiyaçları daha projenin başlangıcında belirlenmekte ve sabitlenmektedir. Bu da yapılacak yazılım geliştirmenin kaçınılmaz özelliklerinden biri olan ileriki safhalarda değişiklik konusunda maliyetlerin yüksek olması anlamına gelmektedir.

XP, kendi içinde yer alan temel değer, prensip ve uygulamaları kullanarak değişim maliyetlerini azaltır. Bu sayede uygulandığı yazılım geliştirme projelerinde değişikliklere karşı daha esnek ve rahat bir duruş sergilenmesini sağlar.

- **XP Değerleri**

XP'nin dayandırıldığı beş temel değer mevcuttur. :

- **İletişim:** Yazılım sistemleri, sistemi geliştirenler arasındaki iletişim ihtiyaçlarının karşılanmasını öngörür. Resmi yazılım geliştirme metodolojilerinde bu görev dokümantasyon ile sağlanmaktadır. XP teknikleri, geliştirme timi arasında kurumsal bilgilerin süratle inşası ve yaygınlaştırılmasında kullanılan metotlardır. Hedef tüm geliştiricilere kullanıcı beklentileri ile örtüşen ortak bir sistemin anlayışının kazandırılmasıdır. XP, bu sayede basit tasarımlar, ortak hedefler, kullanıcı-proramcı ortak çalışmaları, sıkça gerçekleştirilen yüz yüze iletişim ve geri besleme sağlar.

- **Basitlik:** XP, en basit çözümlerle işe başlanması düşüncesini destekler. Ekstra işlevlerin dah sonra da kazandırılabilceğini savunur. Bu yaklaşımla diğer sistem geliştirme metotları arasındaki temel fark, tasarım ve kodlamanın gelecek yerine bugünün ihtiyaçlarına odaklanmasıdır. XP ile gerçekleştirilen yazılımların her ne kadar geleceğin ihtiyaçlarına cevap vermesi beklenmese de gelecekte olması beklenen özelliklerin bugünden tahmin edilerek geleceğe dönük olarak geliştirilmesi yerine, tahmin edilen değişiklikler meydana gelmeden ilk geliştirilen sisteme bahse konu özelliklerin kazandırılması öngörülmektedir. Kesin olarak belli olmayan gelecek ihtiyaçlarının tasarımı ve kodlanması kaynakların ihtiyaç duyulmayan özellikler için harcanması anlamına gelebilir. Tasarım ve kodlamadaki basitlik timdeki programcıların çoğunluğu tarafından kolay anlaşılabilirliği sağlar ve iletişim kalitesini artırır.

- **Geri Besleme:** XP içinde geri besleme sistem geliştirmenin farklı boyutlarına işaret eder. Bunlar:

- ♦ **Sistemden Gelen Geri Besleme:** Birim testleri yazılarak ya da periyodik bütünleştirme testleri uygulanarak, programcıların değişikliklerin uygulanmasından sonra sistemin durumuna ilişkin bir geri besleme almaları sağlanmış olur.

- ♦ **Kullanıcıdan Gelen Geri Besleme:** Müşteri ve test ediciler tarafından yazılan işlevsel testler (kabul testleri) sistemin mevcut durumu konusunda somut geri beslemeler sağlar. Bu test şeklindeki gözden geçirmeler, müşterinin gelişmeleri takip edebilmesi ve geliştirilen bölüme ilişkin müşteri görüş ve önerilerinin alınması için ya da üç ayda bir gerçekleştirilir.

◆ **Geliştirici Ekipten Gelen Geri Besleme:** Müşteriler yeni ihtiyaçların farkına vardıklarında planlama faaliyetinde yer alan ekip üyeleri bahse konu ihtiyaçların gerçekleştirilmesi için gerekli süreye ilişkin bir tahmin yürütürler.

Geri besleme aslında iletişim ve basitlik ile yakından ilgilidir. Sistemdeki hatalar, kodlanan bölüm içinde belli bir noktada hata olduğunu ispatlayan birim testleri yazılarak kolayca anlaşılabilir. Sistemden bu noktada gelen direkt geri besleme bahse konu bölümün yeniden kodlanması gerektiğini ifade eder. İşlevsel ihtiyaçlar çerçevesinde sistem periyodik olarak kullanıcı tarafından test edilebilir. Kent Beck tarafından bu husus “İyimserlik programlama meslekî açıdan programlamaya verilen bir zarardır, geri besleme ise onun tedavisi” şeklinde açıklanmıştır.

▫ **Cesaret:** Bu değer sayesinde, geliştiricilerin mümkün olduğunda kodlamanın yeniden yapılmasında bir sakınca görmemeleri ve kendilerini rahat hissetmelerini sağlar. Bu da mevcut sistemin gelecekte değişikliklerin kolayca uygulanmasını sağlayacak şekilde gözden geçirilmesi ve düzenlenmesi anlamına gelir. Cesaret konusunda başka bir örnek de, üzerinde ne kadar emek sarfedilmiş olursa olsun işe yaramayan kaynak kodunun silinmesidir.

▫ **Saygı:** XP’de programcılar derlemeyi önlemesi, birim testlerinde başarısızlık yaratması veya kendileri açısından işleri ertelemesi nedeniyle değişikliklere onay vermediklerinden, ekip üyeleri birbirlerine saygı gösterirler. Üyeler, yeniden değerlendirme sayesinde eldeki en iyi tasarım çözümünün aranması ve yüksek kalitenin sağlanması bakımından birbirlerinin işlerine saygı gösterirler.

İlk dört değerın sağlanması ekip içinde karşılıklı saygıyı geliştirir. Ekipteki hiç kimse kendine değer verilmediğini ve dışlandığını düşünmemelidir. Bu yüksek seviyede isteklendirme sağlar ve ekip içinde ve proje konusunda bağlılığı geliştirir

- **XP Prensipleri**

▫ XP, yazılım geliştirmede ortaya çıkan risklerin azaltılmasını amaçlar. Özellikle geciktirici tasarım kararlarının getirdiği maliyetlerin azaltılmasını hedeflemektedir. Beck, XP’de tasarım kararlarının zamanında alınabilmesi, özellik uyarlamalarının (kendi ifadesi ile opsiyonların) zamanında yapılabilmesi ve bu sayede de maliyetin azaltılması için bir reçete önermekte ve değer sağlayacağı belli

olmayan opsiyonlardan vazgeçilmesinin daha faydalı olacağını ifade etmektedir (Beck 1999:11-14).

- Geleneksel olarak, bir projenin geliştirilme safhasında verilen yeni kararların, başka bir ifade ile getirilen değişikliklerin maliyette üssel bir artış getirdiği bilinmektedir. Bu nedenle bazı opsiyonların göz ardı edilmesinin daha sonra tekrar uygulamaya konması gerektiği takdirde çok maliyetli olacağından (opsiyonun maliyetinden daha yüksek maliyete ulaşabilmektedir) ilave bir maliyet getirebileceği söylenebilir.

- XP, geliştirme esnasında değişiklik yapma maliyetlerini azaltır ve bu sırada üzerinde kesin bir karar verilinceye kadar yüksek risk taşıyan kararlardan vazgeçilmesini sağlar. Tüm XP uygulamaları bu hedefin gerçekleştirilmesi için çalışır.

- XP, yazılım geliştirme sürecini dört ana faaliyete/aşamaya böler. Bunlar:

- ◆ **Kodlama:** XP savunucuları, kodlama olmadan sistemden bahsedilemeyeceğini ve bu nedenle de kodlamanın sistem geliştirme süreci içindeki en önemli faaliyet olduğunu dile getirmektedirler. XP, bir programlama probleminde tüm çözümlerin kodlanması ve otomatik testlerle hangi çözümün en uygun olduğunun bulunması kapsamında birçok alternatifle karşılaşır. Kodlama, aynı zamanda programlama problemleri konusunda fikir alışverişine de imkân tanır. Karmaşık bir programlama problemi ile karşılaşan ve diğer programcılara çözümünün nasıl olacağını anlatabilme konusunda kodlamaya ihtiyaç duyar. Kod, sürekli özlü, açık ve anlaşılır olmalı ve sadece tek bir şekilde yorumlanabilmelidir. Diğer programcılarda bahse konu kod için kendi görüş ve önerilerini kodlama ile gerçekleştirirler.

- ◆ **Test Etme:** Test bir konuda emin olmak için gerçekleştirilir ve müşteri için anlaşılabilir ya da öncelikli bir ihtiyaç değildir. Günümüzde test edilmemiş olduğu halde kullanılan birçok yazılım mevcuttur. Yazılım geliştirmede, XP bu manada her türlü kodlamanın ya da faaliyetin test edilmesinden yanadır. Bu da kişilerin hangi konu(lar)da emin olup olmadıklarının belirlenmesi sorununu ortaya çıkarmaktadır.

Kodlananın ifade edilmek istenenden farklı olup olmadığından emin olunabilmesi için test edilmesi gereklidir. Bu maksatla XP birim



testlerini kullanmaktadır. Bu testler, kodu otomatik olarak test etmektedirler. Programcılar bu manada, kodu bölecek muhtemel durumların tespiti için mümkün olduğunca çok sayıda test yazmaya çalışırlar. Bütün testleri geçen kodun tamamlandığı kabul edilir.

XP, ne demek istenildiği ile ne denilmesi gerektiği arasında farklılık olup olmadığı konusunda emin olunmadığında bunun değerlendirilmesi için edilmesi için yayım planının araştırma safhasında müşteriden alınan ihtiyaç bildirimlerine dayanan kabul testlerini (acceptance tests) kullanmaktadır.

◆ **Dinleme:** Programcılar genellikle geliştirilen sistemin işsel özellikleri bakımından yeterli bilgiye sahip değillerdir. Sistemin işlevini belirleyen de aslında bu işsel yöndür. Bu nedenle de programcılar sistemin bu yönünü doğru (olması gerektiği gibi) algılayabilmek amacıyla “dinleme” yaparlar.

Programcılar, geniş manada müşteri ihtiyaçlarını dinlemek zorundadırlar. Bunun yanında işsel problemin ortaya konması ve müşteriye problem konusunda geri besleme sağlanması müşteri problemine ilişkin olarak müşterinin kendi anlayışının/farkındalığının geliştirilmesini sağlar. Müşteri ve programcı arasındaki iletişim Planlama Oyunu yönteminde ele alınacaktır.

◆ **Tasarımlama:** Basitlik bakış açısında, sistem geliştirmenin kodlama, test ve dinlemeden başka bir şeye ihtiyaç göstermediği düşünülse de uygulamada durum farklıdır. Tasarım olmadan çok yol alınabilse de sistem gittikçe karmaşıklaşacağından ve sistem içindeki bağımlılıklar git gide gözden kaybolacağından belli bir zamanda geliştirme tıkanabilecektir. Bundan kaçınılması için, sistemdeki mantığı ortaya koyacak bir tasarım yapısının ortaya konulması zorunludur. İyi bir tasarımda sistem içinde bağımlılıklar az olacağından sistemin bir bölümünde yapılacak değişiklikler diğer bölümleri etkilemeyecektir.

Bir XP projesi sürümlerden oluşur. İlk sürüm bir başlangıç yapmayı amaçlar. İzleyen sürümler projeye işlevsellik kazandırır, değişiklik yaratır ve hataları düzeltir. XP projesi uygulamanın bütün ömrü boyunca devam eder ve bu sayede yazılımın mümkün olduğunca faydalı olabilmesi için sürekli olarak güncellenir ve yenilenir. Bu durum müşteri tamam diyene kadar devam edebilir. Sürümler arasında bir ila üç ay zaman bulunabilir. Her bir sürüm bir ila üç haftalık aşamalara bölünmüştür.

Bir XP projesinde, ihtiyaçlar önceden belirlenmez. Projenin başında veya müşterinin istediği her hangi bir zamanda müşteri tarafından istenen özellik

(story/feature) yazılır ve tipik bir “Durum Çalışması (Use Case) yoluyla da açıkça anlaşılması sağlanır. Projenin başında bir sürüm planı hazırlanır. İlk olarak, müşteri tarafından istenen tüm özellikler yazılır. Geliştirme ekibi daha sonra her bir özellik için bir maliyet belirler. Bu maliyet tek bir geliştirici için bir, iki veya üç “ideal programlama haftası” olabilir. Maliyet bundan küçük olduğunda birden fazla özellik birleştirilir, büyük olduğunda ise özellik iki ve daha çok bölüme ayrılır. Müteakiben özellikler sürümlere dağıtılır. Sürüm tarihleri bu sayede her bir sürüme dağıtılan özellikler doğrultusunda hesaplanabilir ya da özellikler belirlenmiş sürüm tarihlerini karşılayacak şekilde dağıtılır. Bunun için de, ideal programlama haftalarının takvim haftalarına nasıl dönüştürüleceğinin tahmin edilmesi gerekir. İlk etapta bu çok zor olsa da proje ilerledikçe tahminler gittikçe iyileşir.

Müşteri her bir aşamanın başlangıcında, mevcut sürüm içinden aşama dâhilinde uygulanacak olan ve kendisi için en önemli olduğu değerlendirilen özellikleri seçer. Özellikler daha sonra görev (task) adı verilen küçük birimlere ayrılır. Her bir geliştirici istediği sayıda görevi seçerek sorumluluğunu üstlenir. Görevler daha sonra onları seçen geliştirici tarafından ideal planlama günleri şeklinde tahmin edilir ve mevcut aşamada gerçekleştirilir. Bir aşama ya da sürümden elde edilen tecrübe (ideal programlama zamanının doğrulanması gibi), izleyen sürüm veya aşama planlamasının daha iyi tahmin edilebilmesi bakımından dikkate alınır.

- **Uygulama Alanları**

“XP, net olmayan veya çabuk değişen ihtiyaçlar kapsamında yazılım geliştiren küçük-orta büyüklükteki proje ekipleri için ince/hafif ağırlıkta bir metodolojidir” (Beck 1999:15). XP, kullanıcının da aslında tam olarak ne istediğini bilmediği, net olmayan ihtiyaçların söz konusu olduğu ya da iş durumlarındaki değişiklikler veya dış etkenlerin bir sonucu olarak değişiklik gereken durumlarda iyi bir seçimdir. Çünkü XP’de ürünün geliştirilmesi birçok küçük döngüye bölündüğünden ve her bir döngü de ayrı ayrı planlandığından planlama değişiklikleri sürekli, hızlı ve kolayca gerçekleştirilir. XP kullanıldığında proje ekibinin büyüklüğü önemlidir. XP’de proje ekibi küçük-orta büyüklükte olmalıdır. Pratikte, bu ekibin en çok on kişi olması anlamına gelir. Bu sayı belki birkaç kişi fazla olabilse de yirmi çok yüksek bir rakamdır (Beck 1999:157).

Olumsuz yönlerine rağmen XP halen daha bazı projeler için esas çözüm olma özelliğini korumaktadır. XP;

▪ İhtiyaçların hızla değiştiği ya da öngörülemeyen uygulama problemlerinin belirlenmesini ve yeni ya da prototip teknolojisinin kullanılmasını gerektiren projelerde,

▪ Çalışma sonucu yazılım ürününün kendisi olmayıp odak bilgi olan araştırma projelerinde,

▪ Hızlı metotlar kullanılan küçük ve kolayca yönetilen projelerde,

▪ Hızla değişen İşsel İhtiyaçlar/Talimatlara uyum sağlanmasını gerektiren projelerde,

▪ Özel nitelikli mali sistem projelerinde kullanıldığında başarı getirmektedir.

XP'nin yazılım mühendisliğinin en iyi uygulamalarından türetilmiş ve dört grupta toplanmış 12 uygulaması mevcuttur:

▪ İyi Derecede Geri Besleme Sağlayan Uygulamaları

- ◆ Çift Yönlü Programlama (Pair programming)
- ◆ Planlama Oyunu (Planning game)
- ◆ Test Esaslı Geliştirme (Test driven development)
- ◆ Bütüncül Tim (Whole team)

▪ Sürekli Süreçler Sunan Uygulamaları

- ◆ Sürekli Bütünleştirme (Continuous integration)
- ◆ Yeniden Tanımlama ya da Tasarım İyileştirme

(Refactoring or design improvement)

- ◆ Küçük Sürümler (Small releases)

▪ Ortak Anlayış (Shared understanding)

- ◆ Kodlama Standartları (Coding standards)
- ◆ Toplayıcı Kod İyeliği (Collective code ownership)
- ◆ Basit Tasarım (Simple design)
- ◆ Sistem Metaforu (System metaphor)

▪ Programcı Kolaylıkları (Programmer welfare)

- ◆ İdame Ettirilebilir Adım (Sustainable pace)

- **Avantajları**

XP projeleri oldukça kolay şekillendirilebilir niteliktedir. Kullanılabilir bir ürünün piyasaya sürümü çok hızlı olabilir. Bu sayede hem iş çevresi ürünün sağlayacağı avantajlardan faydalanmaya başlar, hem de gerçek ortamda kullanım sayesinde sağlanacak geri besleme sonucunda ürünün müteakiben sürekli olarak geliştirilmesi sağlanmış olur. Özellikle, proje müşteri için de bir keşif niteliği taşıdığına gerçek ortamda kullanımdan gelecek geri beslemeler ve değişen fikir, istek ve durumlara uyum sağlanabilmesi hiçbir değerle kıyaslanamaz. İlaveten sürecin çok şeffaf olması, proje yönetim ekibini mutlu edecek şekilde projenin ilerleme, durum ve yönetiminin de çok şeffaf olmasını sağlar

- **Dezavantajları**

Belirli bir ortamda XP kullanımının önündeki en büyük engel müşterinin ta kendisidir. Bu nedenle müşteri geliştirme ekibinin bütünleşik bir parçası olmak zorundadır. Bu da müşterinin proje ekibi çevresinde her zaman ulaşılabilir durumda olmasını gerektirir. XP, bunun uygulanmasının mümkün olmadığı ya da müşterinin bir çalışanını projeye sürekli olarak tahsis etmek istemediği durumlarda düzgün çalışamayacağından metodolojinin kullanımından vazgeçilmelidir. Başlıca dezavantajları aşağıda sunulmuştur:

- **İhtiyaçların Değişkenliği:** XP taraftarları, müşterinin sürekli yazılım geliştiren ekibin bulunduğu yerde temsilci bulduğunda değişiklik isteklerinin gayri resmi olacağı ve bunun da süreci daha esnek hale getirerek resmi yazışma zorunluluklarından kurtaracağını dile getirmektedirler. Ancak, XP karşıtlarınca bu durum, yüksek maliyetli yeniden çalışmalar ve proje kapsamının önceden üzerinde anlaşılmış ve kaynağı belirlenmiş hususlardan uzaklaşabileceğini vurgulamaktadırlar.

- **Kullanıcı Çatışmaları:** Değişiklik kontrol kurullarının varlığı, iki veya daha fazla kullanıcı arasında proje hedef ve kısıtları konusunda potansiyel çatışmaların olabileceğine işaretler. XP metodolojisi, programcının üzerinde uzlaşılan kısıt ve hedeflerin dokümantasyonundan çok kodlama üzerinde yoğunlaşmasını sağlayan birleştirilmiş bir müşteri bakış açısını kabul edebilmelerine bağlıdır. Bu aynı zamanda birden çok programlama şirketince üstlenilen projelerde muhtemel rekabetten de kaynaklanabilmektedir.

- İhtiyaçların özellik dokümanlarından çok otomatik kabul testleriyle vurgulanması,

- İhtiyaçların hepsinin aynı anda önceden alınmak yerine peyder pey tanımlanması,
- Programcılarının ikiyeşerli gruplar halinde çalışması,
- İhtiyaç değişimi nedeniyle yeni baştan tasarılmanın daha az tasarım gayreti gerektirmesi: Tasarım faaliyetlerinden çoğu basitten zora doğru gerçekleştirilir. Çalışabilecek en basit tasarımla başlanır ve karmaşıklıklar müteakiben yapılan testler sonucunda eklenir. XP karşıtları, bu durumun ihtiyaçlar değiştikçe gerçekleştirilen yeniden tasarılmadan daha fazla tasarım gayretinin harcanması anlamına gelebileceğini ifade etmektedirler. (Bu husus literatürde Büyük Tasarım Önermesi (Big Design Up Front – BDUF) bulunmaması olarak da kullanılmaktadır)
- Bir müşteri temsilcisinin proje ekibinde yer alması: Bu projenin başarısızlığında ana sebep olabilir ve bazı programcılarca bu durum stres kaynağı olarak görülebilir. Teknik bilgisi yeterli olmayan ancak teknik yazılım özellikleri ve mimarisinin kullanımına ilişkin yönlendirmelerde bulunmaya çalışan bir temsilci tarafından mikro seviyede yönetilme tehlikesi ortaya çıkabilir.

- **Diğer Metodolojilerle Birlikte Kullanımı**

Beck, XP tarafından reçete edilen uygulamaların denenmesini, çalışan uygulamaların elde tutulmaya devam edilmesini ve çalışmayan uygulamaların ise, yeniden düzenlenmesini ya da onlardan vazgeçilmesini tavsiye etmektedir. Ancak, XP'den sağlanacak en büyük faydanın bütün uygulamaların etkin kılınması sayesinde gerçekleştirilebileceği teorisini savunmaya devam etmekte ve bunu 20-80 kuralı (İşin % 80'ini bitirildiğinde elde edilecek faydanın ancak % 20'sinin görülebilmesi) olarak ifade etmektedir. En yüksek faydanın sağlanabilmesi için bütün uygulamaların kullanılmasına ihtiyaç olduğu savından hareketle XP'nin diğer metodolojilerle birlikte kullanımı sağlayacağı faydaların azalmasına ya da birlikte kullanımın XP'ye kendi uygulamalarından çıkarımda bulunulmadan yeni uygulamalar eklenmesiyle sınırlı olması gerektiği söylenebilir.

## **2.5. Scrum Metodolojisi**

Proje yönetiminde kullanılan hızlı yazılım geliştirme metodlarından biridir. 1990 yılından beri kullanılmakta olan bu metodoloji “hiper-verimli araç” olarak da isimlendirilmektedir ve daha önce çok az sayıda ya da hiç hızlı sonuç üretemeyen ağır yöntemlerle şaşkına dönmüş proje ekibinin verimliliğinin üst seviyede artırılması için dokümantasyonu yapılmıştır (Wikipedia, 2009).

1986'da, Hirotaka TAKEUCHI ve Ikujiro NONAKA ticari yeni ürün geliştirmede hız ve esnekliği artıran yeni bir yaklaşım tanımladılar (Takeuchi ve Nonaka 1986). Burada;

- Aşamaların üst üste bindiği ve tüm sürecin farklı aşamalar boyunca çapraz işlevli bir tim tarafından gerçekleştirildiği bu yaklaşımı, ileri ve geri paslaşmalarla belli bir mesafeye takım halinde gidilerek sayı yapılmasına dayanan Amerikan futboluyla (rugby) kıyaslamışlardır.

- Durum çalışmaları otomotiv, fotoğraf makinesi, bilgisayar ve yazıcı endüstrilerinden alınmıştır.

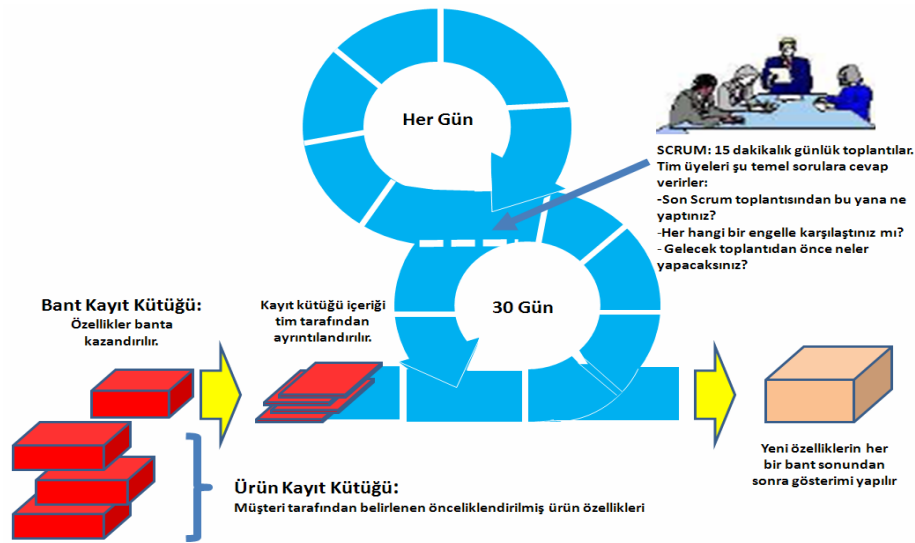
DeGRACE and STAHL, 1991'de yayınlanan kitaplarında (Degrace ve Stahl 1991) bu yaklaşımı Scrum olarak nitelendirmişlerdir.

1990'ların başında, Ken SCHWABER şirketinde "İleri Düzey Geliştirme Metotları" adında Scrum benzeri bir yaklaşım kullanmıştır.

Aynı tarihlerde, Jeff SUTHERLAND Easel Şirketinde benzer bir yaklaşım geliştirmiş ve bu yaklaşıma "scrum" adı vererek, bu terimi kullanan ilk kişi olmuştur.

1995'te SUTHERLAND ve Ken SCHWABER müştereken Scrum yaklaşımını tanımlayan ilk halk tanıtımı olarak kabul edilebilecek bir bildiri sunmuş, bugün Scrum olarak kabul ettiğimiz her türlü yazılı belge, tecrübe ve en iyi endüstri uygulamalarını bir araya getirmek için işbirliği içinde çalışmışlardır.

2001'de Ken SCHWABER, Mike BEEDLE ile Scrum ile Hızlı Yazılım Geliştirilmesi konusunda bir kitap yazmak üzere bir araya geldi.



Şekil-E5.8: Scrum İş Akışı (<http://www.controlchaos.com>)

Scrum, genellikle hızlı yazılım geliştirme (agile software development) metodolojisi ile birlikte kullanılan tekrarlı ve artırımlı bir yazılım geliştirme sürecidir. Yazılım geliştirme projeleri için geliştirilmiş olmasına rağmen, kullanımda olan yazılımların bakım timleri ya da program yönetim yaklaşımı olarak da kullanılabilir. Farklı aşamalar Şekil-E5-8'de sunulmuştur.

Scrum'da doğru çalışma yapılabilmesi için proje ekipleri küçük olmalıdır, ancak büyük ekipli projelerde bu durum proje ekibinin daha küçük ekiplere (alt ekipler) bölünmesiyle çözümlenebilir.

İdeal bir durumda tüm ekip üyelerinin aralarında en uygun iletişimin sağlanabilmesi bakımından aynı yerde olmaları istenir. Ancak bu mümkün olmadığında Scrum toplantıları telekonferans şeklinde icra edilir.

- **Scrum Süreci**

Scrum, ön tanımlı rol ve uygulamalar seti içeren bir süreç iskeletidir. Scrum yaklaşımında, süreçleri kontrol eden ve proje yöneticisi gibi çalışan Usta (ScrumMaster), Ürün Sahibi (Product Owner) ve programcılardan oluşan tim (Team) ana oyuncularlardır.

Scrum kapsamında uzunluğuna tim tarafından karar verilen, her biri 15 ile 30 günlük sürerlerle gerçekleştirilen ve ürün kayıt kütüğü dâhilinde tim tarafından yapılması öngörülen görev/işlerin gerçekleştirilme durumunun ele alındığı "sprint" adı verilen toplantılar yapılır. Bu toplantılarda belirlenen süreçte tim tarafından gerçekleştirilen potansiyel kullanılabilir yazılım konusundaki ilerlemeler değerlendirilir. Ürün kayıt kütüğü dokümanından (product backlog) alınan önceliklendirilmiş üst düzey ihtiyaçları gösterir özellik setleri her bir sprint içinde değerlendirilir. Sprint planlama toplantısıyla kayıtlı hangi hususların sprint dâhilinde ele alınacağı belirlenir ve bu hususlara ilişkin olarak tamamlanmasını istediği ürün kayıt kütüğü öğeleri konusunda tim üyelerini bilgilendirir. Tim müteakiben bir sonraki sprint toplantısına kadar istenilen hususlardan ne kadarını tamamlayabileceklerine ilişkin taahhütte bulunurlar (Schwaber 2004:164). İhtiyaçlar sprint esnasında sabitlenmiş olduğundan kimsenin sprint kayıt kütüğü dokümanında değişiklik yapma şansı yoktur

Sprint ve Scrum süreci yönetim sistemlerinin birçok başarılı uygulaması mevcuttur. Scrum tarafından sunulan en büyük avantaj hiç şüphesiz kolay öğrenilir ve küçük gayretlerle kullanılmaya başlanır olmasıdır.

- **Scrum Oyuncuları**

Scrum kapsamında tanımlanmış çok sayıda oyuncu, domuzlar (pigs) ve tavuklar (chickens) olmak üzere iki gruba bölünmüştür (Schwaber 2004:164). Domuzlar, yazılımın gerçekleştirilmesinden sorumlu iken, geri kalanlar tavuk olarak nitelendirilirler. Söz sahibi olmadıkları halde proje ile ilgili olan tavuklar, proje başarısız olsa bile domuz olmadıklarından bundan sorumlu olmazlar. Tavukların ihtiyaçları, tutkuları, fikirleri ve etkileri hâlihazırdaki Scrum projesini etkilemeyecek veya rahatsız etmeyecek şekilde hesaba katılmalıdır.

- **Domuz Görevler**

Domuzlar projenin gerçekleşmesinden sorumlu şahıslardır.

- **Ürün Sahibi (Product Owner)**

Müşteriyi sesini temsil eder. Scrum timinin iş bakış açısından doğru işlerle uğraştığından emin olunmasını sağlar. Kullanıcı hikayeleri yazar, bunları önceliklendirir ve bunları ürün kayıt kütüğüne koyarlar.

- **Usta (ScrumMaster, Facilitator)**

Usta, aslı görevi sprint hedeflerini gerçekleştirecek timin önündeki engellerin kaldırılması olan Scrum yöneticisidir. Kendi içinde teşkilatlanan timin lideri olmamasına karşın tim ile problemler arasında bir tampon görevi görür. Scrum sürecinin beklenen şekilde gerçekleştiğinden emin olunmasını sağlayan kişidir ve kural ve sprint uygulamalarını takip ve kontrol eder.

- **Tim**

Timin ürünün hazırlanmasında sorumluluğu vardır. Küçük bir tim mevcut işi gerçekleştirecek çapraz işlevli yeteneklere sahip (birden fazla görevi yürütebilecek kapasitede) tasarımcı, geliştirici vb. gibi 5-9 kişiden oluşur. .

- **Tavuk Görevler**

Scrum sürecinin bir parçası olmamalarına rağmen hesaba katılması gerekli görevlerdir. Hızlı geliştirme yaklaşımının önemli bir yönü de kullanıcı, iş ve müşteri katkısının sağlanmasıdır. Bu kişiler için önemli olan her bir sprint planlanması ve gözden geçirilmesi sonucu elde edilen çıktılara geri besleme sağlanmasıdır.

- **Kullanıcılar**

Yazılımı kullanacak hedef kitledir.



▫ **Müşteriler/Yatırımcılar**

Süreç içerisinde direkt olarak görev almadığı halde projenin gerçekleştirilmesini sağlayan kişilerdir.

▫ **Yöneticiler**

Yazılım geliştirme teşkilatı için uygun ortamın tesisi ve korunmasını sağlayan kişilerdir.

• **Scrum Toplantısı**

Sprint boyunca her gün, proje durum toplantısı gerçekleştirilir. Buna “scrum” ya da “günlük gösteri” adı verilir. Özellik arz eden Scrum kuralları:

▫ Toplantı tam zamanında başlar ve geç kalmalara verilecek cezalar timce kararlaştırılır.

▫ Herkes katılabilmesine karşın sadece domuzlar konuşabilir.

▫ Toplantı süresi timin büyüklüğüne bakılmaksızın 15 dakika ile sınırlandırılmıştır.

▫ Tüm katılımcılar ayakta durur, oturulmaz.

▫ Toplantı her gün aynı yer ve aynı zamanda başlar.

Toplantı sırasında, her bir tim üyesi üç soruya cevap vermek zorundadır (Schwaber 2004:164):

▫ Dünden beri ne yaptınız?

▫ Yarın ne yapmayı planlıyorsunuz?

▫ Hedefinizi gerçekleştirmekten sizi alıkoyan herhangi bir problem mevcut mudur? (Problemlerin hatırlanması Usta'nın görevidir)

Süreçte sürekli gelişimin sağlanması amacıyla, her bir sprint sonunda tüm tim üyelerinin bir önceki sprint kapsamındaki görüşlerinin alındığı geriye dönük süresi dört saat ile sınırlı bir sprint (sprint retrospective) gerçekleştirilir.

Scrum, tüm tim üyelerinin aynı mekânda çalışmasını sağlayarak kendi içinde organize olabilen timlerin oluşturulmasını ve tim içinde ve projede yer alan disiplinler arasında sözlü iletişim kurulabilmesini sağlar.

Scrum için kilit bir prensip, proje süresince müşterilerin fikirlerinin, isteklerinin ya da ihtiyaçlarının değişebileceğinin farkına varılması ve beklenmeyen problemlerin geleneksel tahminler ya da planlı tepkiler ile kolayca çözümlenemeyeceğinin anlaşılması gerekliliğidir. Scrum; problemin tamamı ile

anlaşılamayacağını/tanımlanamayacağını kabul eden, timin acil ihtiyaçları süratle belirleme ve cevap verme kabiliyetinin ençoklanmasına odaklanan deneysel bir yaklaşım getirmektedir.

- **Scrum Dokümanları**

- **Ürün Kayıt Kütüğü (Product Backlog)**

Ürün kayıt kütüğü, projenin bütünün dikkate alan, istek-listesi öğeleri gibi ihtiyaç duyulan tüm özelliklerin ayrıntılı tanımlarını içeren üst düzey bir dokümandır. İçerdiği gün bazında kaba tahminler sayesinde Ürün Sahibi'nin zaman dilimlerini ve bir noktaya kadar öncelikleri belirlemesine yardımcı olur.

- **Sprint Kayıt Kütüğü (Sprint Backlog)**

Sprint kayıt kütüğü, müteakip sprint kapsamında timin ihtiyaçların gereğini nasıl yapacağı konusunda bilgi içeren oldukça ayrıntılı bir dokümandır. Görevler saatlere bölünmüş olmakla birlikte hiçbir göreve 16 saatten fazla süre tahsis edilmez. Görevin daha uzun sürmesi durumunda alt görevlere bölünmesi gereklidir. Sprint kayıt kütüğündeki görevlere personel görevlendirmesi yapılmaz. Bunun yerine tim üyeleri istedikleri görevi kendiliklerinden üstlenirler.

- **İş Tamamlanma Durumu Çizelgesi (Burn-down Chart)**

Mevcut sprint için planlanan görevlerden ne kadarının tamamlandığını ya da ürün kayıt kütüğü içindeki konu sayısını gösteren herkese açık çizelgedir. Bu çizelge kazanılmış değer çizelgesi (earned value chart) ile karıştırılmamalıdır. Bu çizelge sprint için tahsisli sürenin büyük bölümünde boş kalabilmektedir.

- **Uygulama Alanları**

Scrum, teorik olarak bir grup insanın ortak bir hedefi arşivlemek için beraber çalışması gerektiğinde uygulanan bir metodolojidir. Sözde "Toplantıların Toplantısı (Scrum of Scrums)" olarak da ifade edilen bu yöntem bir proje yönetim yaklaşımı olarak da kullanılmaktadır.

Scrum, yazılım geliştirilmede kullanılmak üzere tasarlanmış olmasına karşın diğer endüstri dallarında da kullanılabilir. Günümüzde scrum, ürün geliştirme ya da iş yönetiminde tekrarlı ve artırımlı bir süreç olarak değerlendirilmektedir.

Scrum yaklaşımının ürün geliştirmede kullanımına ilişkin ilk referanslar "Yepyeni Ürün Geliştirme Oyunu" (Takeuchi ve Nonaka, 1986:137-146) ve "Bilgi Yaratım Şirketi" dir (Takeuchi ve Nonaka, 1995). Günümüzde kayıtlar

scrum yaklaşımının finans, internet ve tıbbî ürünlerin geliştirilmesinde kullanıldığını göstermektedir.

Pazarlama çoğunlukla proje esaslı bir yaklaşımla gerçekleştirilir, çoğu proje yönetim prensibi pazarlamaya uygulanır ve proje yönetim tekniklerine benzer yaklaşımlarla optimize edilir. Pazarlama alanında Scrum kullanımının pazarlama yöneticilerinin karşı karşıya kaldığı problemlerin çözümünde faydalı olduğu düşünülmektedir. Kısa ve aralıklı toplantılar, timlerde görevli tüm personelin diğerlerinin ne üzerinde çalıştığını ve tüm timin hangi doğrultuda işi devam ettirdiğini bilmek zorunluluğu nedeniyle küçük pazarlama timleri için son derece önemlidir. Pazarlamada scrum kullanımı:

- Muhtemel problemlerin projenin ilk aşamalarında görülebilmesini ve daha hızlı şekilde, daha az kayıplarla düzeltilebilmesini sağlar. Scrum yaklaşımının “hiçbir problemin hasıraltı edilmemesi” konusundaki anahtar prensibine göre her tim üyesi karşılaştığı bütün grubu etkilemesi muhtemel problemleri/zorlukları tanımlaması konusunda cesaretlendirilmektedir.

- Finansal riskleri azaltır. Her bir sprint periyodunun başlamasıyla işveren, tüketici sayısının artırılması için yatırımların sayısının artırılması, belirsizlikler kaybolana kadar yatırımların azaltılması ya da başka girişimlerin finansmanı gibi herhangi bir pazarlama proje parametresini hatasızca değiştirebilir.

- Pazar planının daha esnek olmasını sağlar. Sprint dâhilinde gerçekleştirilecek kısa vadeli pazarlama planları daha etkin olabilir. Pazarlama yöneticileri sprint periyodunda başarısız olan bir promosyon metodundan diğerine geçme fırsatına sahip yakalarlar. Her bir tim üyesince üstlenilmiş her bir küçük ancak önemli görevin tamamlanma tarihlerinin belirlenmesini kolaylaştırır.

- **Avantajları**

- Müşteriler geliştirme timinin bir parçası olabilmektedirler.
- Diğer hızlı yazılım geliştirme süreçleri gibi Scrum, çalışan durumda sık orta seviye dağıtım sağlayabilirler. Bu, müşterinin çalışan yazılımı erkenden elde etmesini ve projenin değişen ihtiyaçlara göre şekillenebilmesini sağlar.

- Risk ve riske karşı koyma planları timin kendisi tarafından geliştirilir. Bu sayede her aşamada risklere karşı önlem alınması, risk takibi ve yönetimi doğrulukla gerçekleştirilmiş olur.
- Planlama ve modüler geliştirmede şeffaflık sağlar.
- Süreçte alınan yolun müşteriler/sponsorlar tarafından izlenebilmesini sağlar.
- Hiçbir problem hasıraltı edilmez ve kimse öngörülemeyen bir problemi bulduğu ya da farkına vardığı için cezalandırılmaz.
- Çalışma alanları yeterli ve çalışma saatleri optimum seviyede tutulur.
- Bir Scrum fasılada dışarıdan herhangi rahatsız edici bir etki gelmediğinden proje ekibi görevlere odaklı ve yaratıcı olur. Bu da verimi artırır.
- Her bir fasılanın sonunda o ana kadar ne yapıldığı ve bundan sonraki fasılda neler yapılması gerektiği değerlendirilebilir. Bu da projenin esnek olmasını sağlar.

- **Dezavantajları**

Bir fasılda Scrum ekibi sadece kendisinden sorumlu olduğundan proje yöneticilerinin işin nasıl yapılacağına müdahale etmemeleri gerekir. Bu nedenle proje yöneticilerinin ekip üyelerinin doğru işi yaptıkları konusunda tam bir güvene sahip olmaları istenir ki bu da potansiyel olarak ileride problemleri bir durum yaratabilir.

- **Diğer Metodolojilerle Birlikte Kullanımı**

Scrum tek başına bir ürünün mühendisliğinin nasıl yapılacağını açıklayamaz, bu proje ekibine bağlıdır. Bu da Scrum'un diğer metodolojilerle birlikte kullanılabilmesine müsaade eder. XP gibi bir metodolojinin projenin mühendislik kısmında kullanılmasının projenin çok başarılı olmasını sağladığı rapor edilmiştir. Scrum metodolojisinin Crystal Clear metodolojisi ile kullanımının da başarılı sonuçlar verebileceği değerlendirilmektedir.

## **2.6. Crystal Clear Metodolojisi**

"Crystal Clear; tatminkâr bir ürün güvenliğini, etkin geliştirmeyi ve sürdürülebilir çalışma koşullarını önceliklendiren ve aynı yerde konuşlu küçük bir ekibin kullanılması için uygulanabilecek en uygun yoldur" (Cockburn 2004:307).

Bu metodoloji, her metodolojinin farklı bir renkle [Şeffaf (Clear), Sarı (Yellow), Portakal (Orange), Kırmızı (Red), Erguvan (Maroon), Mavi (Blue), Mor (Violet)] ifade edildiği Crystal firması metodolojilerinin bir bölümünü oluşturur. Renkler her bir metodoloji için ne kadar personel kullanılacağını gösterir. Crystal Clear en hafif renk kodu olup 2-8 kişiden oluşan en küçük proje grubunun projeye tahsis edildiğini ifade eder.

Crystal Clear metodolojisi, yedi özelliğinin her uygulandığı projede kullanılmasını gerektirir. Bunlardan ilk üçünün kullanımı zorunludur ve geri kalan dört özellik de uygulandığı takdirde projenin geleceği daha güvenli hale gelir. Bu özellikler:

- **Sık Dağıtım Yapılması:** Yapılan çalışmanın ve test edilmiş kodun gerçek kullanıcılara birkaç ayda bir ya da mümkün olduğu takdirde daha kısa zaman aralıklarıyla dağıtım yapıldığında, sponsorlar tarafından yapılan işin ve projedeki ilerlemelerin görülmesi, karşılanan ihtiyaçlar konusunda kullanıcılardan hızlı geri besleme alınması ve geliştiricilerin morallerinin yükseltilmesi sağlanmış olur.
- **Yansıtıcı Geliştirme:** Zaman ayrılarak proje ekibinin projede nelerin işler durumda olduğunu, nelerin işler durumda olmadığını gösterilmesinin ve işler durumda olmayanların iyileştirilmesinin sağlanması hedeflenmelidir.
- **Çok Yakın İletişim:** Tüm proje ekibinin mümkünse tek bir odada, mümkün olmadığı yan yana iki veya daha fazla odada birbirlerine çok yakın tutularak ekip üyelerinin soru yöneltme ve cevaplama ve fazla sorun yaşamadan anında yapabilmelerinin, beraber çalışabilmelerinin, birbirlerinin işlerini denetleyebilmelerinin ve geçerli bilgileri alabilmelerinin sağlanması gereklidir.
- **Kişisel Emniyet:** İnsanlar, korkusuzca konuşabildiklerinde başkalarının işleri konusunda yapıcı eleştiri yapabilirler ve hatalarını kabul ederler. Bu da proje içinde dürüst ve güvenilir bir ekibe sahip olunmasını sağlar.
- **Odaklanma:** İnsanlar toplantı veya diğer çalışmalar gibi düşünme melekelerini kullanmamalarına sebep olabilecek herhangi bir azaltma yapılmadan haftada iki tam gün ve günde iki saat kendi amaçlarına vakit ayırabildiğinde, işlerine daha kolay odaklanmakta ve işler daha çabuk bitmektedir.
- **Uzman Kullanıcılara Kolay Erişim:** Proje ekibi uzman kullanıcılara kolayca ulaşabildiğinde sorularına çabuk cevap alabilecek ve kalite ve tasarım kararlarına ilişkin geri besleme sağlayacaklardır.

- **Otomatik Testler, Konfigürasyon Yönetimi ve Anında Bütünleştirmeli Teknik Ortam Sağlanması:** Test ve konfigürasyon yönetimi/sürüm kontrol görevlerinin (yedekleme yapılması ve değişikliklerin birleştirilmesi) elle yapılmak zorunda olmaması geliştiriciler için hayatı daha kolay hale getirir. Bu da uygun bir teknik ortam sağlanması ile mümkündür.

Crystal Clear bu kritik özelliklerin kullanılmasına yardım edecek, seçimi isteğe bağlı birçok değişmez teknik sunmaktadır (Cockburn, 2004). Genel olarak, Crystal Clear özellikleri tekniklerden daha çok değer verdiği söylenebilir. Bu da Crystal Clear metodolojisini düşük eşikli bir metodoloji haline getirmektedir: Proje grupları; proje grubunun büyütülerek, ya da özel bir duruma uyum sağlaması amacıyla geliştirilerek kullanıldığı metot ve teknikleri, Crystal Clear'a uygulayabilirler. Bu nedenle de yeni teknik ve metotlar öğrenip kullanmak zorunda kalmazlar.

- **Uygulama Alanları**

Yukarıda da açıklandığı üzere Crystal Clear, aynı fizikî alanda çalışan 2-8 kişiden oluşan ve bir ya da iki uzman kullanıcıya ulaşabilen proje grupları tarafından kullanılabilir. Genelde, ilk üç özelliğin (idealde yedi) kullanılabildiği her türlü düzenleme için geçerli olabileceği söylenebilir. Ancak, buna sıkı sıkıya bağlı kalınması zorunluluğu yoktur. Crystal ailesindeki tüm metodolojiler, herhangi bir proje hedeflenen metodolojiye uymadığında, metodoloji tarafından gerçekleştirilen prensip ve pratiklerin özel bir duruma uydurulmak amacıyla çekildiği çek-uydur (stretch to fit) prensibini desteklemektedir. Bu sayede aslında sekiz kişiden oldukça fazla olan proje ekiplerinin Crystal Clear metodolojisini kendi ihtiyaçlarına göre başarıyla çekip kullandıkları rapor edilmiştir.

- **Avantajları**

- Yedi özelliği de kullanıldığı takdirde projenin başarıyla sonuçlandırılması olasılığı yüksek olmaktadır (Cockburn, 2004). Başarı için ilave bazı faktörlerin de bulunması gerektiği noktasından hareketle, bu özellikler her ne kadar bir başarı garantisi olmasa da bu özellikler başarılı ile başarısız arasında fark yaratabilecek bir kalite farkı yaratabilir.

- SSADM ya da PRINCE gibi geleneksel kalın (thick) metodolojilerden farklı olarak Crystal Clear proje ekiplerinden yapılması beklenenleri nasıl yapmaları gerektiği konusunda esnektir. Aslında, Crystal Clear öğrenilmesi gereken en az sayıda yeni teknikle mümkün olduğu kadar çok proje ekibi tarafından

kullanılabilecek şekilde tasarlanmıştır. Bu özelliği ile XP gibi hızlı geliştirme metodolojilerinden bile ayrılmaktadır.

- **Dezavantajları**

- Crystal Clear'in çok kuvvetli olduğu nokta kendisi için aynı zamanda ana dezavantajdır. Mümkün olduğunca fazla duruma uygulanabilir olmaya çalışmakta, bu da onu herhangi bir özel durumda kullanılabilecek en iyi metodoloji olmaktan alıkoymaktadır. Cockburn'ünde kabul ettiği gibi;

“Direkt olarak git ve XP'ye yapış”.....Crystal Clear çoğunluk içindir. Her kim XP'ye yönelirse bu onun faydasına olur”.

- Crystal Clear hâlâ oldukça yeni olduğundan hakkında yazılmış kitap sayısı sınırlıdır ve gerçek hayatta çok miktarda kullanılmamış olabilir. Öte yandan, metodolojinin arkasındaki prensipler gerçek projelerden elde edilen gerçek tecrübeler baz alınarak belirlenmiştir. Belki de bu nedenle daha geniş kullanımı aslında reklamı yapıldığı gibi olup olmadığını gösterecektir.

- **Diğer Metodolojilerle Birlikte Kullanımı**

Crystal Clear ince/hafif bir metodoloji olduğundan çok sayıda metodoloji ile birlikte her ikisinden faydalanılabilecek şekilde kullanılabilir. Bu, beraber kullanılacağı metodolojiden Crystal Clear içine bir ya da daha fazla tekniğin aktarılması veya her iki metodolojinin de aynı anda denenebilecek şekilde birleştirilmesiyle iki şekilde sağlanabilir. İlki şüphesiz daha kolaydır.

Diğer metodolojilerle kullanımına ilişkin aşağıdaki çıkarımlar yapılmıştır:

- XP uygulama tekniklerinden herhangi biri ya da birkaçının (rutin dağıtım ve hatırlatma/bilgilendirme dokümantasyonu gibi) Crystal Clear içine aktarılması, her ikisinin birleştirilerek kullanımını gerçekleyecektir (Cockburn 2004:251).

- Crystal Clear, Scrum metodolojisi ile birlikte kullanıldığı takdirde, herhangi bir noktadan başlanabilecek, yüksek iletişimli ve anlık geri beslemeli kısa döngülerle çalışılabilecek ve sonunda ihtiyacı karşılayacak şekilde sonuçlanacak, bir süreçsiz süreç (No-Process process) üretilmiş olacaktır (Cockburn, 2004:251).

- Crystal Clear, RUP gibi kalın bir metodoloji ile de kullanılabilir. Crystal Clear projelere uymak için çekilebilirken RUP tam tersine daraltıldığından

uygulanması güçtür. Bu mümkün olsa da sağlayacağı fayda tartışmaya açıktır (Cockburn, 2004:254).

### **3. PROJE YÖNETİM TEKNİKLERİ**

Bu bölümde bir dizi yazılım proje yönetim tekniği (PMBOK, COCOMO, MTA, EV and Critical Path) üzerinde durulacaktır. Bu teknikler, projenin farklı özelliklerinin tahmini, takibi ve değerlendirilmesi konusunda yardımcı bir araç olarak kullanılmaktadırlar.

#### **3.1. Proje Yönetimi Bilgi Ünitesi (Project Management Body of Knowledge - PMBOK)**

Proje Yönetimi Bilgi Ünitesi (PMBOK) terimi Proje Yönetim mesleği içinde bilginin nasıl toplanacağını açıklamaktadır. Hukuk, tıp ve muhasebe gibi diğer mesleklerde olduğu gibi, bilgi ünitesi de uygulayıcı ve akademisyenlerce uyarlanır ve iyileştirilir. PMBOK içinde kullanımı daha kısıtlı yenilikçi ve geliştirici uygulamalar yanında geniş uygulama alanına sahip geleneksel pratikler de bulunmaktadır. PMBOK hem yayınlanmış, hem de yayınlanmamış hususlar da içermektedir (PMI 2000).

PMBOK proje süreçlerini başlatma, planlama, icra, kontrol ve nihayetlendirme olmak üzere birbirinden farklı beş proje grubuna böler. Bu gruplar bir sıra dâhilinde uygulanmıyor olabilir. PMBOK aynı zamanda bütünleştirme (integration) yönetimi gibi birçok proje bilgi alanlarını da tanımlar. Bu süreç grupları ve bilgi alanlarından faydalanılarak hazırlanan proje süreç sınıflandırması Tablo E5.1'de sunulmuştur. PMBOK Kullanım Klavuzu bu proje süreçlerinin uygulanmasında kullanılan genel kabul görmüş tekniklerin ve metodolojilerin özetlerini sunmaktadır (PMI 2000).

- **Uygulama Alanları**

PMBOK, genel kabul görmüş uygulamaları, araçları, teknikleri ve geçerli hususları proje yönetimi sahasında bir araya getirerek bilgi ve uygulamalarda meydana gelen büyümeleri yansıtmaya çalışmaktadır. Genel kabul görmüş demek bilgi ve uygulamaların PMBOK çatısı altında tüm projelerde tamamen kullanılacağı anlamına gelmemektedir. Proje Yönetim Ekibi bu konuda proje için her zaman en uygunu belirlemekten sorumludur (PMI 2000).



**Tablo-E5.1:** Proje Yönetim Süreçlerinin Proje Grupları ve Bilgi Alanlarına göre Düzenlenmesi (PMI 2000)

| Bilgi Alanları                       | Proje Grupları               |  |                               |                             |                             |
|--------------------------------------|------------------------------|--|-------------------------------|-----------------------------|-----------------------------|
|                                      | Başlatma                     | Planlama   | İcra                          | Kontrol                     | Sonlandırma                 |
| Bütünleştirme (Entegrasyon) Yönetimi |                              | Proje Planı Tanımlaması  | Proje Planı İcrası            | Bütünleşik Değişim Kontrolü |                             |
| Kapsam Yönetimi                      | Başlatma, Kapsam Tanımlaması | Kapsam Planlaması  | Kapsam Değişim Kontrolü       | Kapsam Doğrulaması          |                             |
| Zaman Yönetimi                       |                              | Faaliyet Tanımlaması, Faaliyet Sıralaması, Faaliyet Süresinin Belirlenmesi, Program Geliştirme                     |                               |                             |                             |
| Maliyet Yönetimi                     |                              | Kaynak Planlaması, Maliyet Tahmini, Maliyet Bütçelemesi  |                               | Maliyet Kontrolü            |                             |
| Kalite Yönetimi                      |                              | Kalite Planlaması  | Kalite Garantisi              | Kalite Kontrolü             |                             |
| İnsan Kaynakları Yönetimi            |                              | Organizasyon Planlaması, Personel Alımı  | Proje Ekibinin Geliştirilmesi |                             |                             |
| İletişim Yönetimi                    |                              | İletişim Planlaması  | Bilgi Dağıtımı                | Performans Raporlaması      | İdari Kapanış               |
| Risk Yönetimi                        |                              | Risk Yönetim Planlaması, Risk Tanımlama, Kalitatif Risk Analizi, Kantitatif Risk Analizi, Risk Müdahale Planlaması |                               | Risk Takip ve Kontrolü      |                             |
| Satın Alma Yönetimi                  |                              | Satın Alma Planlaması, Solicitation Planlaması   | Solicitation Kaynak Seçimi    |                             | Sözleşmenin Sonlandırılması |

PMBOK çatısında yer alan ve iyi bilinen bazı teknikler: Kazanılmış Değer (EV) Yönetimi (4.4), Program Değerlendirme ve Gözden Geçirme Tekniği (PERT) (<http://www.netmba.com/operations/project/pert>) ve Kritik Yol Metodudur (CPM) (4.5).

- **Avantajları**

- PMBOK süreç grupları ve bilgi alanları şeklinde genel bir proje yönetim çatısı sağlar.

- PMBOK genel kabul görmüş proje yönetim prensiplerinin kısa özetini sunar ve onlara referans sağlar.

- PMBOK birleştirilmiş bir proje yönetim terminolojisi önerir.

- **Dezavantajları**

- PMBOK sadece bir taslak sağlar, projenin gerçek ihtiyaçlarına ilişkin sorgulama bilgisi bir yönetim ekibi tarafından belirlenmelidir.

- PMBOK, birçok proje yönetim metodoloji ve teknikleri konusunda çok az bilgi sağlar. Girdi ve çıktıları öğrenmek isteyenlerin kesinlikle bu konudaki özel dokümanlara bakması gerekir.

- PMBOK, sadece meslekten bağımsız proje yönetim süreçlerinin özelliklerini anlatır.

- **Diğer Metodolojilerle Kullanımı**

PMBOK gerçekten de genel kabul görmüş proje yönetim tekniklerinin bir araya gelmesiyle oluştuğundan, istenildiğinde diğer metodolojilerle kolayca bütünleştirilebilir.

### **3.2. Yapısal Maliyet Modeli (COCOMO)**

COCOMO, yazılım projesinin faaliyet, program ve maliyetlerinin tahmin edilebilmesi için kullanılan deneysel ve algoritmik bir modeldir. Bu model çok sayıda yazılım projesinden geçerli verilerin toplanarak analiz edilmesi ve gözlemlere en iyi uyacak şekilde formüle edilmesiyle elde edilir (Sommerville, 2001:522).

COCOMO modelinin ilk sürümü (COCOMO 81) seviyelerin maliyet tahmin analizinin detaylarını yansıttığı üç seviyeli bir modeldi. İlk seviyede kaba bir tahmin yapılmakta, ikinci seviyede bu tahmin proje tecrübeleri ve süreç çarpanları ile düzenlenmekte ve son seviyede de projenin farklı safhaları için en detaylı tahminlere

yer verilmekteydi. COCOMO 81, tahminler üretebilmek için yazılım geliştirme sürecine ilişkin birçok varsayımlar yapmaktaydı (Sommerville, 2001:523).

COCOMO 2 bileşen tabanlı geliştirme ve prototip hazırlama, dördüncü nesil programlama dilleri ve CASE destek araçları gibi bir çok geliştirme metodolojisini desteklemektedir. COCOMO 2 hâlihazırda birbirinden küçük farklılıklar gösteren üç seviyeden oluşur:

- **İlk Prototip Hazırlama Düzeyi:** İlk prototip büyüklük tahminleri hedef noktalarına göre yapılır. Bu hedef noktalar, karşılanması istenen ihtiyaçların karmaşıklığını ölçmenin basit bir yoludur. Yapılması gerekli işlerin yoğunluğu, programcının verimliliği ve hedef noktalar arasında basit bir ekstrapolasyon yapılarak hesaplanır. Hedef noktalar, üçüncü nesil programlama dillerindeki ekran görüntüsü, rapor ve modüllerin sayısına dayanıyor olabilir ve üzerinde çalışılan modül, rapor ya da ekran görüntüsünün karmaşıklığına göre ağırlıklandırılabilir.

- **İlk Tasarım Düzeyi:** Bu düzeyde sistem ihtiyaç tanımlarının tamamlanmasını müteakip bu ihtiyaçlara cevap verecek bir ilk tasarım gerçekleştirilmeye çalışılabilir. Tahminler, hedef noktaların ayrıntılı olarak açıklanmasından elde edilen işlev noktalarına dayandırılır. Toplam nokta sayısı ölçümle ya da program özelliklerinin, dış girdi ve çıktılarının, kullanıcı işlemlerinin, dış ara yüzlerin ve sistem tarafından kullanılan dosyaların tahmini ile hesaplanır. İşlev noktaları müteakiben COCOMO modelinin sunduğu tabloları kullanan kaynak kodu satır sayısına dönüştürülür.

- **Müteakip Mimari Düzeyi:** Sistem mimarisi tasarlandığında yazılımın büyüklüğüne ilişkin mantıklı ve doğru bir tahmin yapılabilir. Bu düzeyin kullandığı tahminleme sistemi, personel yeteneklerini ve ürün ve proje özelliklerini reddeden daha yoğun bir çarpan seti kullanır (Sommerville 2001:523-524), [Daha ayrıntılı bilgi için bkz. (Boehm ve Clark, 1995)]

- **Uygulama Alanları**  
COCOMO, en çok bilinen deneysel algoritmik maliyet tahminleme tekniğidir. Oldukça iyi seviyede yazılı hale getirilmiştir ve ticari araçlarla desteklenmektedir. Yaygın olarak kullanılmakta olup 1981'deki ilk piyasaya sürümünden bu yana önemli oranda yol kat etmiştir (Sommerville 2001:523-524).

Modelin ilk sürümü, geliştirme sürecindeki oldukça büyük değişkenlere bağlı olarak kısıtlıydı. Bu problem sürekli geliştirmeler ve model uzantıları ile giderildi ve bu çalışmalar COCOMO 2 ile sonuçlandı.

- **Avantajları**

- Herhangi bir proje maliyetinin tam olarak belirlenmesi zor olmasına rağmen, istenildiği takdirde iyimser ve kötümser tahminlerle kullanılabilir veriler elde edilebilir.

- Modelin uygulanması ve kullanımı kolay olup ihtiyacı karşılar niteliktedir. Bunun sonucu olarak da toplum ve ticari araçlarla desteklenmektedir.

- COCOMO iyi bilinen ve oldukça iyi seviyede yazılı hale getirilmiş bir tekniktir.

- **Dezavantajları**

- Proje büyüklüğüne ilişkin tatminkâr bir tahminin yapılması oldukça zordur.

- Kaynak kodu satır sayılarının karmaşıklık/zorluk derecesinin ölçülmesinde kullanımı neredeyse imkânsızdır. COCOMO, her ne kadar tüm ana programlama dilleri için farklı tablolar sunarak bunu hesaba katsa da kullanıcıların farklı bakış açıları, rutin dışı kullanımlar, kodların mükerrer kullanımı vb. gibi birçok uyumsuzluğu gideremez.

- Modeldeki çoğu girdi parametresi sayısal olarak belirlenemez ve tahmin edilmesi gerekir. Programcılarının tecrübe ve üretkenlikleri ile CASE araçlarının yetenek ve uygunlukları buna örnektir. Müteakip COCOMO tahminlerinin doğruluğu bir öncekilerin doğruluklarına son derece bağımlıdır. Öncekilerde problem varsa sonrakilerde de bu problem devam eder.

- COCOMO modeli sık sık güncellenen bir model olmadığından, model Yazılım Mühendisliği alanındaki yeni teori ve uygulamaları dikkate almaması ve yanlış tahminleme yapması muhtemeldir.

- **Diğer Metodolojilerle Birlikte Kullanımı**

Yukarıda açıklandığı gibi COCOMO modeli sadece üzerinde çalışılan projede belirli sayıdaki kriterin karşılandığı durumlarda kullanılabilir. İlave olarak, diğer tahminleme tekniklerinin [Uzman Karar Verme (Expert judgement), Tek Noktadan Tahmin (Estimation by analogy) ve diğer algoritmik maliyet tahminleme modelleri] kullanılması da elde edilen tahminlerin doğruluğunu perçinler. Mesleğin dinamik yapısına da bağlı olarak, yazılım mühendisliği proje maliyet tahmini oldukça

karmaşık bir görev olduğundan, bu tekniklerin avantajları ve dezavantajları olması yanında hiçbir bütün durumlar için uygun değildir.

- **Araçlar**

COCOMO modelinin uygulanması, üzerinde çalışılan proje özelliklerin mümkün olduğunca uyacak değişkenleri kullanan bazı basit matematik formüllerin değerlendirilmesi ile başlar. Diğer örnek uygulamaları; <http://www.schemequest.com>, <http://www.softstarsystems.com/overview.htm>., <http://www.jsc.nasa.gov/bu2/COCOMO.html> sitelerinden görülebilir.

### **3.3. Kilometre Taşı Trendi Analizi (Milestone Trend Analysis-MTA)**

MTA, bir projenin planlandığı şekilde ilerleyip ilerlemediğini değerlendiren bir yazılım mühendisliği tekniğidir. Oldukça basit olan bu teknik, değişikliğe neden olan kilometre taşı kabul edilecek olayların/faaliyetlerin (planlı bir faaliyetin ertelenmesi ya da geliştirilmesi gibi) tarihsel kaydının tutulmasından ibarettir. Bu sayede sütunlarının proje kilometre taşlarını, satırlarının ise değişiklik tarihlerini gösterdiği bir matris elde edilir. Bazı basit değerlendirme teknikleri kullanılarak bu verilerden önemli sayılabilecek çıkarımlar yapılabilir. Bu da tahminlenmiş kritik noktaların değerlendirildikleri günle karşılaştırılması ile gerçekleştirilebilir. Müteakiben öncekiler Y eksenine, sonrakiler X eksenine konur. Bu sayede projenin kilometre taşının grafiği bir eğri şeklinde gerçekleşir. Bu eğrinin aşağı doğru hareketi bahse konu kilometre taşının belirlenen süreden önce gerçekleşecek şekilde güncellendiğini, yukarı doğru hareket ise ileriki bir tarihe ötelendiğini ifade eder. Grafik üzerinde bir kilometre taşının tamamlanıp tamamlanmadığı ise eğrinin "X = Y" doğrusunu kesmesiyle anlaşılır. Grafik genellikle şekil olarak kaba bir üçgen halini de alabilir. Bu da kilometre taşı tamamlandığında eğri çiziminin durdurulmasının bir sonucu olarak ortaya çıkar.

- **Uygulama Alanları**

MTA, ilerlemenin ana göstergesi olarak kilometre taşlarını kullanan tüm projelerde uygulanabilir. Basit ve iyi sonuçlar veren bir teknik olması nedeniyle kolaylıkla proje ilerleme değerlendirmelerinde kullanılabilir. MTA, bir evrimsel bir teknik olduğundan projenin icrası esnasında kullanılması gereklidir. Programdan sapmaların önlenmesi, program kaymalarının düzeltilmesi ve program değerlendirmesinin yapılması askatlarıyla kullanılır.

- **Avantajları**

- MTA; basit, etkin ve etkili bir tekniktir.
- MTA yaygın olarak kullanılmakta ve desteklenmektedir.
- MTA oldukça geniş kullanım ve uygulama alanları vardır.

- **Dezavantajları**

- MTA, kendi içinde paketler arası bağımlılıkların kaydını tutmadığından, kesin bir kilometre taşı tamamlanma süresi ertelendiğinde/değiştiğinde, bahse konu bağımlılıkların da ertelenmesinin/değiştirilmesinin sağlanması gerekmektedir. MTA, bağımlılıkların kaydını tutan oldukça kapsamlı proje yönetim araçlarının bir parçası olarak kullanıldığından uygulamada bu eksikliğin giderilmesi de kolaydır.

- MTA tekniğinin girdilerinin kilometre taşlarının tamamlanmasına yönelik olarak yapılan tahminlerdir. Bu nedenle de bu tahminlerin bilgili ve tecrübeli mühendislerce yapılması önem arz etmektedir.

- **Diğer Metodolojilerle Kullanımı**

Yukarıda da açıklandığı gibi, dikkate alınan tek şey tahminlenen kilometre taşlarıdır. MTA, proje süreç modelinde başka bir kısıta ihtiyaç göstermez ve hemen her projede süreç ilerlemesine ilişkin değerlendirmelerin açıklığa kavuşturulmasında kullanılır.

### **3.4. Kazanılmış Değer (Earned Value - EV) Yönetimi**

Kazanılmış Değer Yönetiminde projenin ilerleme durumu, o ana kadar yapılanların projenin başında yapılan tahminlerle karşılaştırılması sonucunda belirlenir. Bu değerlerin ekstrapolasyonu sayesinde, proje yöneticisi projenin bitiminde kaynakların ne kadarının kullanılmış olacağı konusunda fikir yürütebilir. EV yönetiminde kullanılan bazı kısaltmalar aşağıda sıralanmıştır:

- BCWS (Budgeted Cost for Work Scheduled) – Programlanmış İş Bütçeleme Değeri
- BCWP (Budgeted Cost for Work Performed) – Gerçekleştirilen İş Bütçeleme Değeri
- ACWP (Actual Cost of Work Performed) – Gerçekleştirilen İş Gerçek Değeri

- BAC (Budget At Completion) –Tamamlama Bütçesi
- EAC (Estimate At Completion) – Tamamlama Tahmini

BCWP aynı zamanda kazanılmış değer olarak da ifade edilmektedir. Bu değer, zaman içinde belirli bir noktada projenin gerçekten ne kazandığını göstermektedir. Yapılan iş miktarının maliyeti saat ve para gibi farklı şekillerde ifade edilebilir. Bunun yanında, bir şeyin ne zaman kazanıldığını birilerinin belirlemesi gerekmektedir. İşin tamamen bitirilmesi bir kazanım olarak değerlendirilebileceği gibi, bir kısmının tamalanmış olması da kazanım olarak belirlenmiş olabilir. Kazanım işin sadece bir kısmını kapsadığı durumlarda problem bir görevin ne kadar sürede tamamlanacağını tahmin edilmesi oldukça güçtür. İşin bitirilmesi kazanım olarak kabul edildiği durumda ise problem bir görev kapsamında yapılacak iş tamamlanmadığı sürece kazanılmış değerın sıfır olmasıdır. İşin % 95'i tamamlanmış ve önemli miktarda harcanmış değer olmasına karşın % 100'e ulaşılmadığı sürece sonuç değişmemektedir.

- **EV Hesaplamaları**

- Maliyet Değişimi (Cost Variance:  $CV = BCWP - ACWP$ )

Bu eşitlik belli büyüklükteki bir iş için bütçelenmiş maliyet (BCWP) ile aynı işin gerçekleşen maliyetinin (ACWP) kıyaslanmasını sağlamaktadır. Eşitliğin negatif bir değer alması, maliyetin olması gerekenin altında tahminlendiğini, pozitif bir değer alması ise maliyetin gerekenin üstünde tahminlendiğini göstermektedir.

- Program Değişimi (Schedule Variance:  $SV = BCWP - BCWS$ )

Bu eşitlik belirli bir ana kadar bütçelenmiş maliyetin gerçekleşme durumu ile programlanmış hali arasındaki farklılıkları göstermektedir.

- Kalan Bütçe (Budget Remaining:  $BR = BAC - ACWP_{Cumulative}$ )  
Projenin tamamlanması için kalan bütçeyi göstermektedir.

- Kalan İş (Work Remaining:  $BCWR = BAC - BCWP_{Cumulative}$ )

Bu eşitlik projede kalan işlerin yapılabilmesi için halen daha kazanılması gerekli miktarı göstermektedir.

- Tamamlama Değişimi (Variance at Completion:  $V_{AC} = BAC - EAC$ )

Bu eşitlikle iş bitimindeki planana maliyet ile gerçekleşen maliyet arasındaki farklılıkları göstermektedir. Negatif bir değer alması durumunda

projenin planlanandan daha yüksek maliyetle tamamlandığını, pozitif değer alması durumunda ise proje sonucunda elde halen para kaldığını göstermektedir.

▪ Maliyet Performans Etkinlik İndeksi (Cost Performance Index (Efficiency):  $CPI_{Efficiency} = BCWP / ACWP$ )

Bu projenin maliyet bakımından ne kadar etkin gerçekleştirildiğini göstermektedir. Örneğin; 2 değerini alması projenin hâlihazırda planlanan miktarın yarısına mal olduğu ya da tahminlenenden iki kat daha etkin gerçekleştirildiği anlamına gelmektedir.

▪ Program Performans Etkinlik İndeksi (Schedule Performance Index (Efficiency):  $SPI_{Efficiency} = BCWP / BCWS$ )

Bu, projenin zaman bakımından ne kadar etkinlikle yürütüldüğünü göstermektedir. Eşitliğin 2 değerini alması projenin tahminlenenden iki kat daha hızlı yürütüldüğünü ya da iki kat daha etkin gerçekleştirildiğini göstermektedir.

▪ Tamamlama Tahmini (Estimate At Completion:  $EAC = BAC / CPI_{Efficiency}$ )

Bu eşitlikle proje sonunda maliyetin ne olacağını tahminlemesi yapılır. Bu tahmin değeri eşdeğer başka bir formülde ise “ $EAC = ACWP + (BA - BCWP) = CPI_{Efficiency}$ ” şeklinde gösterilmektedir.  $CPI_{Efficiency}$  değerinin projenin seyri esnasında değişebilen hareketli bir hedef olması nedeniyle, projenin tamamı için bu değer hesaplanması yerine son üç aylık vb. bir zaman aralığı için hesaplanması daha mantıklı olacaktır. Bu sayede projenin hâlihazırdaki performansının daha çok dikkate alınmasını sağlayacaktır.

- **Avantajları**

Proje esnasında proje yöneticisi tarafından projenin planlandığı/bütçelendiği ekseninde gerçekleşip gerçekleşmediği konusunda fikir yürütülebilir. Bu yapılamadığı durumda da en azından projenin planlanan bütçesini aşmasına ne kadar kaldığının görülmesi sağlanır.

- **Dezavantajları**

Belirli bir zaman dilimi için kazanılmış değer hesaplanması oldukça güçtür. Yanlış tahminlemeler projenin olduğundan daha iyi ya da daha kötü gittiğinin sanılmasına neden olabilir.



- **Diğer Metodolojilerle Birlikte Kullanımı**

EV yönetimi, önceden yapılan planlamalarda belirlenmiş hedeflere ne zaman ulaşılması gerektiğinin belirlenmesinde kullanılabilir. Bu nedenle de kalın metodolojilerle kullanımı daha uygun olur.

### **3.5. Kritik Yol Modeli (CPM, Critical Path Model)**

Kritik yol tekniği, projede tamamlanması gerekli bütün görevleri sıralayan ve döngüsel olmayan bir grafik üzerinde çalışır. Buna literatürde “proje ağı” adı verilir.

Proje ağına dâhil bir görev tipik bir İş Dağılım Ağacı (Work Breakdown Structure-WBC) tamamlanma elemanıdır. Grafik, tamamlanması gereken görevler ve bunlara arasındaki bağımlılıkları belirler. Her bir görev zaman içinde bağıl bir maliyet anlamına gelmektedir. Kritik yol projenin başlangıcından bitimine kadar olan en uzun yoldur ve projenin tamamlanabileceği en kısa süreyi göstermektedir. Görevlerdeki her hangi bir gecikme, kritik yolda bütün projenin ertelenmesi anlamına gelmektedir. Tüm projenin ertelenmesini engelleyici bir yöntem her bir proje görevi için bir “boş zaman (slack time)” tanımının yapılmasıdır.

- **Uygulama Alanları**

Kritik yol görev programlamasının yapılması maksadıyla tüm projelerde kullanılabilir. Görevler arasındaki bağımlılıkların kritik yol hesaplamasını geçerli kılacak şekilde yeter büyüklükte tanımlanmış olması gereklidir. Lineer şelale metodunda olduğu gibi tüm görevler için kritik yol tanımlaması yapılabileceği durumlarda mevcuttur.

- **Avantajları**

Kritik yol analizi, çok açık ve kafa karıştırmayan bir yöntemdir. Proje içinde en önemli noktaların belirlenmesi ve gerçekleştirilmeleri esnasında dikkatle takibi sağlanır. Bunun ötesinde, kritik yolda yer lamayan görevler için boş zamanlar belirlenerek dikkate alınabilir.

- **Dezavantajları**

Kritik yol kolayca ve doğru olarak tahmin edilebilen rutin faaliyetler için tasarlanmıştır. Bir görevin tamamlanma süresi konusundaki belirsizlik bu modelle gösterilemez ve bu nedenle de gerçek modelin sonuçlarından farklı olabilir.

- **Diğer Metodolojilerle Birlikte Kullanımı**

Kritik yol programlamasının etkili ve faydalı olabilmesi için görevlerin önceden bilinmesi ve aralarındaki bağımlılıkların tespit edilmiş olması gereklidir. Bu

da modeli XP gibi görevlerin; küçük olması, gerçekleştirmenin hemen öncesinde programlanması ve birbirleriyle oldukça az ya da hiç bağımlılıkları olmadığı durumları öngören metodolojilerle birlikte kullanılamamasına neden olmaktadır.

- **Araçlar**

Kritik Yol analizi temel bir proje yönetim tekniği olup birçok proje yönetim uygulaması tarafından da desteklenmektedir. Bunlardan bazıları;

Microsoft Project (Ticari),

PlanBee (Ticari),

OpenPlan (Ticari),

Manage-XPS (Ücretsiz)

Buraya kadar proje yönetimi konusunda çok bilinen ve sıkça kullanılan proje yönetim model, metodoloji ve teknikleri konusunda özet bilgi arz edilmiştir. Geçmişe bakıldığında yazılım proje yönetiminin hâlihazırda çok sayıda proje yönetim aracı ile desteklenmesine rağmen tahminlenmesi güç bir disiplin olduğu söylenebilir. Bu yazılım proje yönetiminin diğer mühendislik disiplinlerine göre oldukça yeni olmasından kaynaklanmaktadır. Yazılım mühendisliği konusunda hâlihazırda çok sayıda bilgi mevcut olmasına rağmen, halen daha yazılım proje yönetimi konusunda oldukça küçük adımlar atılabilmektedir. Oldukça detaylandırılmış RUP gibi yönetim metodolojileri büyük ve az değişken yazılım projelerinde kullanılabilmesine rağmen, yazılım mühendisliğinin geleceği XP gibi etkileşimli geliştirme metodlarının kullanımında yatmaktadır. Açık kaynaklı geliştirme metodlarının da karmaşıklığa karşı gelecekte savaşı kazanacağı aşikârdır.

## **KAYNAKLAR**

Beck, K. Extreme Programming Explained: Embrace Change, Addison Wesley, 1999

Boehm B, "A Spiral Model of Software Development and Enhancement", "Computer", "IEEE", 21(5), Mayıs 1988 ss.61-72,

Boehm B. ve Clark, B. Cost models for future life cycle processes: COCOMO Annals of Software engineering, 1995

Cockburn, A. <http://alistair.cockburn.us/crystal/books/alstairsbooks.html>., 2004.

Dyer M. ve Linger R. "Cleanroom Software Engineering". IEEE Software 4 (5), September 1987, ss. 19–25).

DeGrace, P. ve Stahl, L.H. Wicked problems, righteous solutions, 1990, ISBN 0-13-590126-X

Foreman, John (2005). "Cleanroom Software Engineering". Software Technology Roadmap. Software Engineering Institute (SEI).

Goodland, M. ve Riha, K. <http://www.dcs.bbk.ac.uk/~steve/1/> SSADM - An Introduction, (03.03.2009)

<http://www-306.ibm.com/software/awdtools/rup> (The IBM home page of the Rational Unified Process. 12.03.2009)

<http://www.comp.glam.ac.uk/pages/staff/tdhutchings/chapter4.html>., Introduction to Methodologies and SSADM, (03.04.2009)

[http://whatis.techtarget.com/definition/0,289893,sid9\\_gci213458,00.html](http://whatis.techtarget.com/definition/0,289893,sid9_gci213458,00.html), Structured Systems Analysis & Design Method- A whatis definition.

Kruchten, Philippe, The Rational Unified Process, Third Edition, 2004

Malcolm Eva, SSADM Version 4 - A Users Guide, McGraw Hill, 2006

Martin, J. Rapid Application Development, Macmillan Coll Div, 1991 ISBN 0-02-376775-8

PMI (Project Management Institute), A Guide to the Project Management Body of Knowledge Crystal Clear Methodology, 2000.

Rerych, M. Wasserfallmodell - Entstehungskontext, Markus Institut für Gestaltungs- und Wirkungsforschung, TU-Wien, 2009.

Schwaber, K. Agile Project Management with Scrum, Microsoft Press, January 2004, ss.163 ISBN 0-7356-1993-X

Spence, Fiona | What is PRINCE2? <http://www.crazycolour.com/p2/0009.shtml> (03.04.2009)

Somerville, I. Software Engineering, Addison Wesley, 2001

Takeuchi, H. ve Nonaka, I. Harvard Business Review, The New New Product Development Game (86116), 1986, ss. 137-146.

Takeuchi, H. ve Nonaka, I. The Knowledge Creating Company, Oxford University Press, 1995.

Wikipedia, Scrum (in management), [http://en.wikipedia.org/wiki/Scrum\\_\(in\\_management\)](http://en.wikipedia.org/wiki/Scrum_(in_management)) (22.03.2009)

## **EK-6:Taşıma Yazılımı İşlevsel Sistem İhtiyaç Özellikleri (SRS)**

### **TAŞIMA YAZILIMI İŞLEVSEL SİSTEM İHTİYAÇ ÖZELLİKLERİ (SRS)**

#### **1. Amaç**

Yazılımın amacı, Kullanıcının tüm kara yolu, deniz yolu (iç su yolu dâhil), hava yolu ve demir yolu ulaştırma ihtiyaçlarının en etkin, süratli ve ekonomik olarak karşılanmasına yönelik taşıma planlamalarının, envanterdeki ve/veya dış kaynaktan temin (Outsourcing, 3PL) edilecek ulaştırma vasıtalarının, araç boş kapasitelerinin ve faaliyetlerinin tüm ulaştırma planlamacıları tarafından karşılıklı olarak zamanında görülerek, paralel ve bütünleşik şekilde yapılabilmesi ve Kullanıcıya karar desteği sağlanması olacaktır.

#### **2. Kapsam**

Yazılım, sivil sektörden askeri ortama kadar geniş bir yelpazede, yapılacak küçük düzenlemelerle kullanım imkânı sunabilecek dinamik bir yapıya sahip olacaktır. Bir lojistik şirketinin ulaştırma faaliyetlerinin planlanmasında kullanılabileceği gibi bir askeri birliğin bir noktadan diğerine taşınmasının planlanmasında da eklenecek küçük ve basit modüllerle kullanılabilecek modülerlikte olacaktır. Başka bir ifade ile sistemin bütünlüğünü bozmadan sisteme yeni modüller eklenmesi veya geçerliliğini yitirmiş modüllerin sistemden çıkarılması mümkün olacaktır.

#### **3. Tanımlar**

Bu bölüm EK-1’de sunulmuştur.

#### **4. Kısaltmalar**

Bu bölüm EK-1’de sunulmuştur.

#### **5. Genel İşlevler**

Sistem kapsamındaki donanım ve yazılımlar; birbirleri ile uyumlu, modüler geliştirilebilir özelliğine sahip olacak ve bütün uygulamaların geliştirilmesinde “İnternet Teknolojileri” kullanılacaktır.

Geliştirilecek uygulama yazılımlarında, mevcut yasal mevzuat birebir geçerli olacak ve uygulanacaktır. Mevzuatta olası değişiklikler, sisteme kolaylıkla yansıtılabilecek ve kullanıma sunulabilecektir.

Kullanıcı için gizlilik arz eden, işlenmesi, saklanması, iletimi ve raporlanması aşamalarında yüksek güvenlik önlemlerinin alınmasını gerektiren veriler

kullanılabileceğinden, her seviyede veri güvenliğinin sağlanması için gerekli teknolojik ortam sağlanacak ve güvenlik standartları evrensel standartlara uygun olarak geliştirilecektir.

Kullanıcı yetkisi tanımlamaları, yazılım geliştirme aşamasında Kullanıcı türlerine göre tespit edilerek uygulamaya konacaktır.

Geliştirilecek sistemler, Kullanıcı yetkilerini kontrol ederek, kullanıcı personelin kimlik doğrulamasını [şifre (sesli ve/veya yazılı), parmak izi, el izi, iris tanımlaması, yüz tanıma] yapacaktır.

Geliştirilecek olan sistemlerin analizi, tasarımı ve gerçekleştirme aşamalarında, firma/kurum/kuruluş/şirketin mevcut olan yazılım ve iletişim altyapısı kullanılacaktır.

Sistem, Kullanıcı tesislerinde kurulu/kurulacak güvenli veri iletişim altyapısı ve GSM (GPRS) v.b. iletişim hatlarında sorunsuz çalışacaktır.

Sistem; Kullanıcı envanterinde bulunan donanımlar ve/veya ilave donanımlar üzerinde kusursuz olarak çalışacak, yazılım geliştirme esasları (yazılım mimarisi, kullanıcı ara yüzü, veri tabanı ve bileşen tasarımı) ve yazılım üretim tekniklerine uygun şekilde karar vericileri destekleyecek, gereksiz tekrarlar içermeyecektir.

Sistem;

- Taşımaya esas malzeme bilgileri ve ulaştırma alt yapısının planlanmasına ve coğrafi verilere yönelik bütünlük ortak bir veri tabanı (VTYM),
- Veri tabanını kullanarak maliyet etkin taşıma/kurye planlamasını, programlamasını, animasyonunu, simülasyonunu ve koordinasyonunu sağlayan, ulaştırma ihtiyaçlarının ve planlamalara ilişkin olası problem alanlarının belirlenmesini sağlayan planlama modülleri (TPM, KURPM),
- Gerçek/gerçeğe yakın zamanlı olarak araç/malzemelerin yer, konum ve durumlarına ilişkin iz bilgilerini sunan (aktif ve pasif iz takibi) etkin araç/malzeme iz takip modülü (ETARİTM),
- Güvenlik ve yetkilendirme modülü olmak üzere beş ayrı modülden oluşacaktır.

Teklif edilecek uygulama yazılımları dışındaki tüm sistem yazılımları, ofis yazılımları ve diğer hazır yazılımlar kendi türlerinin en son sürümleri (Version, Release, Level) olacaktır.

Teklif edilecek tüm yazılımlara ait dokümantasyon orijinal ve lisanslı olarak verilecektir.

İlişkisel Veri Tabanı Yönetim Sistemleri: İşletim ve ORACLE sistemlerinin yedekleme araçları kullanılarak hem işletim sisteminin hem de veri tabanının yedeği alınabilecek şekilde gerekli konfigürasyon yapılacaktır. Veri tabanının yedeklenmesi hem on-line hem de off-line yapılabilecektir.

Yazılım; kendi özgün raster harita sistemi yanında gerektiğinde güncellenebilen katmanlı "Map Info" benzeri bir paket program ve harita sistemini [harita veri tabanı üzerinde bütün dünya haritaları ve katmanları (layers) yüklü olarak] kullanacaktır.

### **5.1. Genel Yazılım Özellikleri**

Tüm yazılımlar grafik ara yüzü, dördüncü kuşak ve nesne tabanlı (object-oriented) özelliklere sahip yazılım geliştirme araçlarıyla geliştirilecektir.

Uygulama yazılımı tasarımında BDYM (CASE) aracı kullanılacaktır.

Yazılımlar modüler olacak, modüller arası ilişkilerde yapısallık göz önünde bulundurulacak ve programcı müdahalesi asgari seviyede olacaktır.

Sistemlere yeni modüllerin ilavesi, modüllerin değiştirilmesi ya da silinmesi mümkün olacak bu durumdan sistemin bütünü etkilenmeyecektir.

Yazılımın Kullanıcı, işlem ve bilgi düzeyinde, gizlilik ve güvenlik taşıması için programlarda yapılan tüm işlemleri takip eden bir modül devamlı çalışır vaziyette olacaktır. Bu modül veya yazılımın uygun nitelikteki farklı fonksiyonları aracılığı ile, herhangi bir işlemin kimin tarafından, hangi tarihte ve ne şekilde yapıldığı sorgulanabilecek ve cevaplanabilecektir. Yapılan değişiklikler eski ve yeni şekli ile sorgulanıp, cevaplanabilecek ve gerektiğinde yapılan değişiklikler geri alınabilecektir.

Uygulama yazılımında yer alan her bir program kendi içinde tanıtıcı nitelik taşıyacak ve sadece yetkili personel tarafından herhangi bir değişiklik yapıldığında, bu değişiklikleri yapan personel, yapılış tarihi ve gerekçesini belirten açıklama satırları ihtiyaç duyulduğunda izlenebilecektir. Bu yapı ve mekanizmaya ait yetki ve sorumluluklar Kullanıcı tarafından netleştirilerek uygulamaya konulacaktır. Yapılacak tüm değişiklikler sistemin kullanımda olduğu ömür devri boyunca saklanacaktır.

Yazılım geliřtirmede kullanılan CASE ürünlerinin bütün özellikleri bilgi ile doldurulacak, boş veri elemanı bulunmayacak, açıklamalar yazılacak, komut satırlarına açıklama yapılacak ve yazılımdaki prosedürlerin ne işe yaradığı belirtilecektir.

## **5.2. Yazılımlarda Uyulması İstenen Standartlar**

### **5.2.1. Kullanıcı Grafik Ara yüzü**

Geliřtirilecek yazılımlar, pencereler sistemi üzerine kurulacak, bütün kontroller mouse (fare) ve klavye yardımıyla gerçekleştirilebilecektir. Aksi belirtilmediği müddetçe yazılımlar web-tabanlı olarak, web-browser ile görütölenecek şekilde geliřtirilecektir.

Geliřtirilecek yazılımın bütün pencereleri istemci (client) bilgisayarının ekran çözünürlüğüne uyumlu olacaktır.

Ekranların, pencerelerin numaralandırılması için sistematik bir kod sistemi uygulanacaktır. Kullanılacak bu kodlar yapılacak bakım ve düzeltmeler için referans numarası olarak kullanılacaktır. Bu kod numaraları her pencerede yer alacaktır.

Yazılımdaki ekranların renkleri, karakter fontu ve büyüklükleri kendi içinde standart olacaktır.

### **5.2.2. Veri Giriři**

Aynı sistemde aynı fonksiyon tuřları tüm ekranlarda aynı amaç için kullanılacaktır. (Örn.: F1 tuřunun her zaman yardım, ESC tuřunun ekrandan çıkmak için kullanılması gibi.)

Standart ya da çok kullanılan deęerler için otomatik ilk deęer tayin edilecektir.

Güncelleřtirme ve özellikle sorgulama ekranlarında manuel bilgi giriři yapılmaktan ziyade sistemde mevcut bilgiler ve varsa kodlar seçilecek ve kullanılacaktır.

Aynı tür bilgi giriři ve sorgulamanın devam etmesi halinde aynı menü/ekran kullanılacak bir üst ekrana çıkmaya zorunluluęu olmayacaktır.

Veri giriş ekranlarında bilgi giriři yapılacak alana gelindiğinde, ilgili alan farklı renkte ve dikkat çekici bir tonda görülecektir.

Veri giriři için kullanılan ekranlarda gerektiğinde bilgi sorgulaması yapılabilecek ve istenen bilgilere kısa sürede ulaşım sağlanabilecektir.

Ekranlarda çeşitli maksatlarla tanımlanmış liste elemanlarına harflerle kısa sürede erişim mümkün olabilecektir.

### **5.2.3. Yardım Ekranları**

Kullanılan tüm ekranlara yönelik yardım alınabilecek açıklayıcı yardım ekranları bulunacaktır.

Yardım ekranları her türlü ekran, kayıt tipi ve hata için gerekli ve yeterli bilgiyi içerecektir. Daha ayrıntılı bilgi için Kullanıcı kılavuzunun ilgili bölümüne referans verilecektir.

Kodlanan tüm bilgilerde Kullanıcının kod bilme zorunluluğu olmaksızın açılacak pencereler yardımıyla kod ve anlamı birlikte tanımlanmış alanlar seçilebilecektir.

Sistemde kullanılan hata mesajlarının Kullanıcı Bilgi İşlem Merkezi (BİM) personeli tarafından değiştirilebilmesi mümkün olacaktır.

Ekranlar, raporlar, hata mesajları, ölçü birimleri v.b. Türkçe olacaktır (Arzu edildiğinde çoklu dil desteği sağlanabilecek şekilde düzenleme yapılabilecektir). Sorgulamalar küçük harf veya büyük harf ile yapılabilecektir.

### **5.2.4. Rapor (Çıktı) Standartları**

Bir veya daha fazla dosya ve tablolardan formatı belirlenmiş raporlar menüden seçilmek suretiyle ekrandan veya yazıcıdan alınabilecektir.

Kullanıcı hazırlanan raporları ekranda sayfa sayfa görebilecek, sayfalar arası ileri geri gidebilecektir.

Raporların numaralandırılması için sistematik bir kod sistemi uygulanacaktır. Kullanılacak bu kodlar daha ileride yapılacak bakım ve düzeltmeler için referans numarası olarak kullanılacaktır.

Bütün raporlar “ekranda ne görüyorsan çıktısını al” yöntemi ile hazırlanacaktır. Yazıcıdan alınacak raporların başlık ve son kısımlarına tarih, konu, sayfa numarası gibi hususlar otomatik olarak basılacaktır. Bazı raporların Kullanıcının kullandığı matbu formlara basılması mümkün olacaktır.

Tüm raporlar için farklı formatlarda (Word, Excel, pdf, text, XML, vb.) kaydedilebilme özelliği mevcut olacaktır.



Kullanılacak olan uygulama geliştirme araçları, nesneye yönelik bir uygulama geliştirme aracı olacak, nesneye yönelik veya nesne tabanlı uygulama geliştirme dillerinden birini veya birkaçını destekleyecektir.

### **5.2.5. Uygulama Yazılımları Genel Özellikleri**

Geliştirilecek uygulama yazılımları, mevcut Kullanıcı iletişim altyapısı üzerinde sorunsuz çalışacaktır.

Sürüm takibi, değişiklik önerilerinin kabul işlemleri vb. konfigürasyon yönetimi ile ilgili işlemler, uygulama yazılımları geliştirme sürecinde gerçekleştirilecektir.

Uygulama yazılımları geliştirme sürecinde;

- İş Akışları Tanımlama Dokümanı,
- Kullanıcı İhtiyaçları Dokümanı,
- Yazılım Geliştirme Planı,
- Uygulama Planı,
- Kullanıcı Ara Yüzleri Tasarım Dokümanı,
- Yazılım, Donanım, İletişim Altyapısı Mimarisi,
- Veri Tabanı Veri Modelleri (Kavramsal, Fiziksel, İlişkisel),
- Sistem Veri Sözlüğü,
- Sistem Bakım Kitabı,
- Kullanıcı El Kitabı,
- Sistem Kurtarma Planı,
- Uygulama Yazılımları Geliştirici Kılavuzu dokümanları hazırlanacaktır.

### **5.3. Veri Tabanı Yönetim Modülü (VTYM)**

Veri Tabanı Yönetim Modülü (VTYM), Kullanıcının hali hazırda kullanmakta olduğu sistemlerle karşılıklı veri alış verişi yapabilecek, veri tabanını yönetecek, işleyecek, değiştirebilecek, güncelleyebilecek işlevler içeren bir sistem olarak geliştirilecektir.

Veri tabanında asgari olarak yer alması istenen veri kategorileri ve bu verileri işlemek (veri girme, sorgulama, değiştirme v.b.) için geliştirilecek ara yüz ve alt modüller şu şekildedir:

### **5.3.1. Hiyerarşik Planlama Veri Giriş Alt Modülü**

Her seviyedeki taşıma planlarının mevcut yapısal hiyerarşi de dikkate alınarak hazırlanması ve bu sayede hiyerarşik bir plan ağacı yapısı oluşturulması bu modül kullanılarak sağlanacaktır.

Bu alt modüle ilişkin ara yüzde, asgari olarak yeni plan girişi, girilmiş plan özelliklerinde değişiklik, kaydetme, silme, yazdırma ve ara yüzde çıkış düğmeleri (button) olacak ve yine asgari olarak “Kod Numarası, ana plandan başlamak üzere hiyerarşik şekilde birbiriyle bağlantılı alt plan isimleri, Plana Referans Verilen (Planın Başlayacağı) Gün (G Günü gibi), Plana Referans Verilen (Planın Başlayacağı) Tarih ve Açıklamalar” gibi alanlar yer alacaktır.

### **5.3.2. Taşıma Görevleri Veri Giriş Alt Modülü**

Bir planda belirtilen bir taşıma için planlayıcının ihtiyaç duyacağı taşınacak birim verilerinin girişini sağlayacak bir alt modül geliştirilecektir.

Bu alt modüle ilişkin ara yüzde, asgari olarak yeni taşıma görevi girişi, girilmiş taşıma görevi özelliklerinde değişiklik, kaydetme, silme, yazdırma ve ara yüzde çıkış düğmeleri olacak ve yine asgari olarak, hangi plana ait Taşıma Görevi Listesi olduğu, Sıra Numarası, görevle taşınacak birim listesi, son varış noktası, Taşıma Hazırlık Süresi, Taşıma Önceliği, son noktada olması istenen en son zaman (CRD-Customer Required Date), görevle taşınacak tahmini yolcu sayısı (varsa), en erken Taşınmaya Başlama Tarihi (Early Departure), Açıklamalar gibi alanları yer alacaktır.

Üzerlerine çift tıklamak suretiyle bir (taşınacak) birimin belirlenen son noktaya gidebilmek için kullanacağı yükleme ve boşaltma liman, meydan ve istasyon verilerinin girilebileceği yeni bir ara yüz açılacak ve bu ara yüzdeki bir kayar menüden (combo box/list box) Coğrafi Bilgi Sisteminde daha önce oluşturulmuş ulaştırma altyapı verisinin ismi seçilecek ve Plana Referans Verilen Tarih ve En Erken Ayrılış Tarihi ile ilişkilendirilmiş ulaştırma altyapısından en erken ayrılış ve varış tarihlerine ilişkin veriler girilebilecektir.

### **5.3.3. Taşınacak Birim Profili Veri Giriş Alt Modülü**

Herhangi bir planda belirtilen bir görevle taşınacak birim yapısının aynıyla başka bir planda kullanılabilmesi için bu yapının isimlendirilmesi ve oluşturulan profil özellikleri verilerinin girişini sağlayacak bir alt modül geliştirilecektir.

Bu alt modüle ilişkin ara yüzde, asgari olarak yeni profil girişi, girilmiş profil özelliklerinde değişiklik, kaydetme, silme, yazdırma ve ara yüzde çıkış düğmeleri olacak ve Kod Numarası, Profil İsmi, Gizlilik Durumu (özel durumlarda), Açıklamalar gibi alanları içerecek şekilde geliştirilecektir.

#### **5.3.4. Veri Kategorileri Tercih Ara Yüzü**

Kullanıcıya (planlamacıya) veri tabanını her açtığına bir profilin ve plan kapsamında girilmiş verilerin müteakiben açılacak ara yüzlerde filtre (filter) edilmesini sağlayacak, raporlama ve görünüm tercihlerinin önceden ve tek seferde yapılmasına imkân sağlayacak bir ara yüz geliştirilecektir.

Bu ara yüz kullanıldıktan sonra diğer ara yüzler açıldığında Kullanıcıya Veri Kategorileri Tercih Ara Yüzüne dönmeden profili/planı değiştirebilme imkânı sağlanacaktır.

#### **5.3.5. Taşınacak Birim Verileri Giriş Alt Modülü**

Taşınacak birim özellik verilerinin girişini sağlayacak bir alt modül geliştirilecektir.

Bu alt modüle ilişkin ara yüzde, asgari olarak yeni birim özellikleri girişi, girilmiş birim özelliklerinde değişiklik, kaydetme, silme, yazdırma ve ara yüzde çıkış düğmeleri olacak ve yine asgari olarak, Taşınacak birim Profil İsmi (Bu alan Taşınacak birim/Taşınacak birim Profili tablosundan alınacak), Taşınacak Birim Kodu, Adı, Kısaltması, Taşıma Başlangıç Noktası, Açıklamalar gibi alanlar yer alacaktır.

#### **5.3.6. Rapor Edilebilir Malzeme Kod Görüntüleme Alt Modülü**

Taşımaya Esas Malzeme Listesi ara yüzünde girilecek malzeme verilerinin standart olarak kullanacağı taşımaya esas yolcu ve malzemelerin genel tiplerine ve kodlarına ilişkin (detaylı özellikleri dahil değil) paylaşılabılır verileri içeren bir alt modül geliştirilecektir.

Bu alt modüle ilişkin ara yüzde, asgari olarak, yazdırma ve ara yüzde çıkış düğmeleri olacak ve yine asgari olarak, taşımaya esas veriler kapsamında oluşturulmuş standart ve taşınacak birime özel paket bilgileri, Yolcu Bilgileri ve Bilinmeyen Malzeme ana başlıkları altında yer alan araç/malzemelerin kodları, adları, kısa adları, kısa tanımları ve ana kullanım amacı adı altında sadece

yetkilendirilmiş Kullanıcılar tarafından değiştirilebilir nitelikteki alanları içerecek şekilde geliştirilecektir.

### **5.3.7. Taşınacak Birim Envanter (Yolcu, Malzeme, Araç) Listesi Veri Giriş Alt Modülü**

Plan profili kapsamında tanımlanmış taşınacak birim envanterine ilişkin verilerin girişini sağlayacak bir alt modül geliştirilecektir.

Bu alt modüle ilişkin ara yüzde, asgari olarak yeni malzeme özellikleri girişi, girilmiş malzeme özelliklerinde değişiklik, kaydetme, silme, yazdırma ve ara yüzde çıkış düğmeleri olacak ve yine asgari olarak, Malzemenin İsmi, Rapor Edilebilir Malzeme Kodu (Rapor Edilebilir Malzeme Görüntüleme ara yüzünden sürükleme ve bırakma (Drag & Drop) yöntemi ile veri girişi yapılacak yazılım geliştirilecektir), Malzeme Tipi (Ekipman, paket/sandık, yolcu, tüketilebilir malzeme, diğer v.b.), Tehlikeli madde özelliği, Hareket Kabiliyeti, Taşınabilme kategorisi (mobility category), taşınabilme modu (kara yolu ile, hava yolu ile, hepsi/birkaç ulaştırma modu ile vs.), yükleme önceliği, Kullanım Boyutları (en, boy, yükseklik), Taşımaya Esas Boyutları (en, boy, yükseklik), Ağırlığı (Toplam/Brüt Ağırlığı, Darası), Malzemenin Resmi, Açıklamalar gibi alanlar bulunacaktır.

### **5.3.8. Taşınacak Birim Profiline Envanter Atama Alt Sistem Yazılımı**

Veri Tabanında Taşınacak Birim Listesi olarak tanımlanan tablodan sürükleme ve bırakma (Drag & Drop) yöntemi ile birimlerin seçilmesini ve bu birimler arasında yine sürükleme ve bırakma (Drag & Drop) yöntemi ile hiyerarşik bir kuruluş oluşturulmasını ve Envanter Listesinden seçilecek taşımaya esas malzemelerin sürükleme ve bırakma (Drag & Drop) yöntemi ile seçilmiş ve hiyerarşik yapı içerisine atılmış bir birimin üzerine bırakılması ve bu seçilen birime tahsis edilen malzemenin o birime ait miktara ilişkin verilerin girişini sağlayacak ve oluşturulan hiyerarşik taşınacak birim yapısına tahsis edilmiş tüm malzemeler ile miktarlarının genel bir dökümünü (sorgulama) yapacak bir alt modül geliştirilecektir.

Bu modüle ait ara yüzde, asgari olarak yeni envanter özellikleri girişi, girilmiş envanter özelliklerinde değişiklik, kaydetme, silme, yazdırma, sürükleme ve bırakma işlemi için Taşınacak Birim Listesi ve Envanter Listesini görüntüleme, hiyerarşik yapıya tahsis edilmiş toplam malzeme özetini görüntüleme ve ara yüzde çıkış düğmeleri olacak ve bu ara yüz yine asgari olarak, ara yüzün solunda ayrılmış bir bölümde yer alan hiyerarşik yapı içinden işaretlenen herhangi bir birime ait

malzemenin Stok Numarası, Rapor Edilebilir Malzeme Kodu, Malzemenin İsmi, Malzeme Miktarı, Açıklamalar gibi alanların ve toplam malzeme özetini görüntüleme düğmesine tıkladığında profil içinde yer alan tüm birimlerin malzemelerinin Rapor Edilebilir Malzeme Kodlarının, Malzeme İsimlerinin, Malzemenin Eldeki Miktarlarının görüntülenebileceği şekilde geliştirilecektir. İhtiyaç duyulabilecek ilave veri alanları veri tabanı tasarım aşamasında belirlenecektir.

### **5.3.9. Ulaştırma Araç Verileri Görüntüleme ve Veri Giriş Alt Modülü**

Her türlü taşıma moduna ilişkin taşıma araçlarının tipine, taşıma kapasitesi dışındaki özelliklerine (hız, azami yüklü ağırlık, hareket sırası, v.b.), kargo/yolcu bölmelerinin yükleme/bindirme kapasitelerine, kendi kendine yükleme yapabilecek yükleme ekipmanı kapasitelerine ilişkin verilerin girileceği bir alt modül geliştirilecektir.

Bu alt modüle ilişkin ara yüzde, asgari olarak yeni taşıma aracı özellikleri girişi, girilmiş taşıma aracı özelliklerinde değişiklik, kaydetme, silme, yazdırma ve ara yüzde çıkış düğmeleri olacak ve Kara yolu Ulaştırma Araç Verileri, Deniz yolu Ulaştırma Araç Verileri, Hava yolu Ulaştırma Araç Verileri ve Demir yolu Ulaştırma Araç Verileri ana kategori olarak gösterilecek şekilde geliştirilecektir. Ara yüzde seçilen taşıma araç kategorisine göre aşağıdaki maddelerde açıklanan ara yüzlere geçiş imkânı sağlanacaktır.

- Hava yolu Ulaştırma Aracı Veri Giriş Ara Yüzü: Bu ara yüzde, asgari olarak Araç Tip Kodu, Aracın Kısa Tanımı, Aracın Ulaştırma Modu, Modeli, Hızı (Knots, Km olarak), Bir Saatlik Uçuş Bedeli (TL, \$ ve € cinsinden), Yükleme ve Boşaltma Süreleri (Gün, Saat, Dakika olarak), Azami Yüklü Olduğunda İkmal Yapmadan Uçabileceği Azami Mesafe, Maksimum Yük Kapasitesi, Bir Günde Azami Görev Süresi, Uçak Kategorisi (Geniş Gövdeli Uçak, Orta Boy Ulaştırma Uçağı v.b.), Kargo Bölümünün Tipi (Yük, Yolcu, Yük/Yolcu v.b.), Kargo Bölümünün Yükleme Önceliği, Kargo Bölümünün Adı (Yolcu Bölümü, 1 nci Yük Bölümü v.b.), Kargo Bölümünün Depo Tipi (Klimalı, Kapalı Bagaj v.b.), Kargo Bölümünün ebatları (Uzunluğu-Genişliği-Yüksekliği), Uçağa Yüklenebilecek Maksimum Kargo ebatları (Uzunluğu-Genişliği-Yüksekliği ve Ağırlığı), Kargo Bölümü Tabanının Maksimum Yük Direnci (kg/m<sup>2</sup>), Maksimum İstif Ağırlığı (kg), Kendi Kendine Yükleme/Boşaltma Kapasitesi ve Ekipmanları, Açıklamalar gibi alanları içerecek, kargo bölümlerine “YENİ”, “EKLE” ve “SİL” düğmelerine basılmak suretiyle yeni kargo bölümleri ilave edilebilecek/silinebilecek ve bilgiler girildiğinde aracın toplam kapasitesi yolcu, ton,

LM (lanemetre), metrekare, metreküp ve 20 Feet Standart Konteyner (TEU, Twenty-Foot Equipment Unit) bazlarında ara yüz üzerinde görülebilecektir.

- Demir yolu Ulaştırma Aracı (Tren ve Vagon) Veri Giriş Ara Yüzü: Bu ara yüzde, asgari olarak Araç Tip Kodu, Aracın Kısa Tanımı, Aracın Ulaştırma Modu, Hızı (Mil, Km olarak v.b.), Bir Tam Vagon/Tren Kiralama Bedeli (TL, \$ ve € cinsinden), Vagon Ağırlığı (Dara), Vagon Tipi, Yükleme Platformunun Yüksekliği (Maksimum/Minimum), Yükleme ve Boşaltma Süreleri (Gün, Saat, Dakika olarak), Yükleme Yeri (Önden-Yandan-Her İkisi v.b.), Kargo Bölümünün Tipi (Yük, Yolcu, Yük/Yolcu v.b.), Kargo Bölümünün Yükleme Önceliği, Kargo Bölümünün Adı (Yolcu Bölümü, 1 nci Yük Bölümü v.b.), Kargo Bölümünün Depo Tipi (Klimalı, Kapalı Bagaj v.b.), Kargo Bölümünün ebatları (Uzunluğu-Genişliği-Yüksekliği), Yüklenebilecek Maksimum Kargo ebatları (Uzunluğu-Genişliği-Yüksekliği ve Ağırlığı), Kargo Bölümü Tabanının Maksimum Yük Direnci ( $\text{kg/m}^2$ ), Maksimum İstif Ağırlığı (kg), Kendi Kendine Yükleme/Boşaltma Kapasitesi ve Ekipmanları, Açıklamalar gibi alanları içerecek, kargo bölümlerine “YENİ”, “EKLE” ve “SİL” düğmelerine basılmak suretiyle yeni kargo bölümleri ilave edilebilecek/silinebilecek ve bilgiler girildiğinde aracın toplam kapasitesi yolcu, ton, LM, metrekare, metreküp ve TEU olarak ara yüz üzerinde görülebilecektir.

- Kara yolu Ulaştırma Aracı (Araç ve Grup araç) Veri Giriş Ara Yüzü: Bu ara yüzde, asgari olarak Araç Tip Kodu, Aracın Kısa Tanımı, Aracın Ulaştırma Modu, Hızı (Mil, Km olarak), Bir Tam Araç Günlük Kiralama Bedeli (TL, \$ ve € cinsinden), Hareket Sırası, Aracın Ağırlığı (Dara), Aracın Kullanım Tipi (Yolcu, Konteyner, Kamyon, Platform Römorklu v.b.), Aracın Fonksiyonu (Yolcu aracı, treyler, semi-treyler, yakıt tankeri), Yük Sınıfı, Yükleme Platformunun Yüksekliği (Maksimum/Minimum), Yükleme ve Boşaltma Süreleri (Gün, Saat, Dakika olarak), Kargo Bölümünün Tipi (Yük, Yolcu, Yük/Yolcu), Kargo Bölümünün Yükleme Önceliği, Kargo Bölümünün Adı (Yolcu Bölümü, 1 nci Yük Bölümü v.b.), Kargo Bölümünün Depo Tipi (Klimalı, Kapalı Bagaj v.b.), Kargo Bölümünün ebatları (Uzunluğu-Genişliği-Yüksekliği), Kara yolu aracına Yüklenebilecek Maksimum Kargo ebatları (Uzunluğu-Genişliği-Yüksekliği ve Ağırlığı), Kargo Bölümü Tabanının Maksimum Yük Direnci ( $\text{kg/m}^2$ ), Maksimum İstif Ağırlığı (kg), Maksimum İstif Yüksekliği (m), Kendi Kendine Yükleme/Boşaltma Kapasitesi ve Ekipmanları, Açıklamalar gibi alanları içerecek, kargo bölümlerine “YENİ”, “EKLE” ve “SİL” düğmelerine basılmak suretiyle yeni kargo bölümleri ilave edilebilecek/silinebilecek

ve bilgiler girildiğinde aracın toplam kapasitesi yolcu, ton, LM, metrekare, metreküp ve TEU olarak ara yüz üzerinde görülebilecektir.

- Deniz yolu (İç Su yolu dahil) Ulaştırma Aracı Veri Giriş Ara Yüzü: Bu ara yüzde, asgari olarak Araç Tip Kodu, Aracın Kısa Tanımı, Aracın Ulaştırma Modu, Hızı (Dz.mili, Km olarak), Bir Tam Araç Günlük Kiralama Bedeli (TL, \$ ve € cinsinden), Hareket Sırası, Tekne Kategorisi (Kuru Yük, Yolcu, Tanker v.b.), Aracın gerçek veya sanal (Örn. Orta Boy Ro-Ro Gemisi) olup-olmadığı, Yükleme ve Boşaltma Süreleri (Gün, Saat, Dakika olarak), Tekne Su Seviyesinden Yüksekliği (Height ABW), Tekne Uzunluğu, En Yüksek Seren Direği Yüksekliği, Draftı (Azami Yükle Su altında kalan bölüm yüksekliği), Boş Ağırlığı (Deadweight) (ton), Ambar Kapasitesi (ton), Tekne Tipi (Genel Kargo, Yolcu Ro-Ro, Konteyner, Yük/Eğitim Gemisi v.b.), Seyir Sınıflandırması (Açık Deniz, Sahil Gemisi, İç Su Yolu Gemisi v.b.), Kullandığı Yakıt Tipi, Yüklü/Yüksüz Bir Günde Kullandığı Yakıt Ortalaması (ton/gün), Yüklü/Yüksüz Sürati, buz kırma (Kutuapta Hareket) kabiliyeti, Rampa Tipi (Yan Rampa, Kıç Yükleme Demir yolu Rampası v.b.), Kuru Yük Bölümünün Ambar Sayısı, Kuru Yük Bölümünün Ambar Kapak Sayısı, En Büyük Ambar Kapak Uzunluk ve Genişliği, Çark Tipi, Çark Kapasitesi (ton), Mühimmat Dip Güvertesi Olup-Olmadığı, Yolcu/Yük Güvertesinin Sıra No, Yolcu/Yük Güvertesinin Yükleme Önceliği, Yolcu/Yük Güvertesinin Adı (Yolcu Güvertesi, 1 nci Yük Güvertesi v.b.), Yolcu/Yük Güvertesinin Depo Tipi (Klimalı, Kapalı Bagaj v.b.), Yük Güvertesinin yüklenebilir ebatları (Uzunluğu-Genişliği-Yüksekliği), Gemiye Yüklenebilecek Maksimum Kargo ebatları (Uzunluğu-Genişliği-Yüksekliği ve Ağırlığı), Kargo Bölümü Tabanının Maksimum Yük Direnci (kg/m<sup>2</sup>), Maksimum İstif Ağırlığı (kg), Kendi Kendine Yükleme/Boşaltma Kapasitesi ve Ekipmanları, Açıklamalar gibi alanları içerecek, güverte bölümlerine “YENİ” “EKLE” ve “SİL” düğmelerine basılmak suretiyle yeni kargo bölümleri ilave edilebilecek/silinebilecek ve bilgiler girildiğinde aracın toplam kapasitesi yolcu, ton, LM, metrekare, metreküp ve TEU olarak ara yüz üzerinde görülebilecektir.

### **5.3.10. Ulaştırma Araç Filosu Veri Giriş Alt Modülü**

Her çeşit taşıma moduna ait Ulaştırma Araç Verileri Tablosunda tip olarak tanımlanan taşıma araçlarından seçilecek araçları birden fazla sayıda çoğaltacak aracın lokasyonu (bağlama limanı, konuşlanma havaalanı, garaj, gar) ve haberleşme tipleri gibi aracı taşıma ve kurye planlaması için özellikli hale getirecek ilave verilerin girilmesini sağlamak suretiyle, belirli bir birimin filosunu oluşturacak ve

filo listesinin tamamının ve araç tipi-ulaştırma modu ve kaynak tipine göre (envanterdeki araçlardan oluşan filo ve envanter dışı araçlardan oluşan filo) bir kısmının rapor olarak alınmasını sağlayacak bir alt modül geliştirilecektir.

Bu alt modüle ilişkin ara yüzde, asgari olarak yeni ulaştırma aracı özellikleri girişi, girilmiş ulaştırma aracı özelliklerinde değişiklik, kaydetme, silme, yazdırma ve ara yüzden çıkış düğmeleri olacak ve yine asgari olarak, Kara yolu Ulaştırma Araç Verileri, Deniz yolu Ulaştırma Araç Verileri, Hava yolu Ulaştırma Araç Verileri ve Demir yolu Ulaştırma Araç Verileri ana kategori olarak gösterilecek ve ara yüzden seçilen ulaştırma araç tipinin Ulaştırma Araç Verileri Tablosundaki özelliklerinin yanı sıra Aracın İsmi, Aracın Kodu (Uçak Kuyruk Numarası, Kara yolu Aracı Plakası v.b.), Araç Tipi, Kaynak Ülke (varsa), Kaynak Tipi (Envanter, Kamu/Özel Sektöre ait, Yurt Dışı Kiralama, Diğer v.b.), Aracın Bulunduğu Ülke, Aracın Bulunduğu Yer (Bağlama Limanı, Üssü, Garaj, Gar v.b.) ve açıklamalar gibi alanlar bulunacak ve bu alanlar müteakip sürümlerde Kullanıcı ihtiyacına göre yeniden düzenlenebilecektir.

#### **5.4. Yükleme Öncelikleri Veri Giriş Ara Yüzü**

Yolcu ve malzeme yükleme önceliklerinin ulaştırma moduna ve aracına göre yetkili birim tarafından girişi yapılabilecektir.

Bu ara yüzde, asgari olarak yeni yükleme önceliği girişi, girilmiş yükleme önceliklerinde değişiklik, kaydetme, silme, yazdırma ve ara yüzden çıkış düğmeleri olacak ve yine asgari olarak, Ulaştırma Modu, Ulaştırma Araç Kodu, Yükleme Öncelik Sırası, Yolcu Kod No., Rapor Edilebilir Malzeme Kodu, Açıklamalar alanları yer alacaktır.

#### **5.5. Coğrafi Bilgi Sistemi Verileri Yaratma Alt Sistem Yazılımı ve Veri Giriş Ara Yüzleri**

Coğrafi Bilgi Sistemi üzerinde tüm dünya ve/veya fare ile seçilen bir bölgeyi içine alan jenerik ve gerçek ilgi sahası ve istenilen/seçilen katmanları (layer) gösterebilen haritalar yaratacak, "LOKASYON EKLE" komutu ile jenerik ve gerçek şehir, ulaştırma alt yapısı, ulaştırma ağları (network) v.b. yaratacak, harita üzerinde seçilen bir bölgedeki jenerik ve gerçek yerleşim yeri, ulaştırma alt yapısını (terminaller, limanlar, istasyonlar, römork değiştirme noktaları, v.b.) sorgulayacak, filtre edecek alt sistem yazılımı ile bunlara ilişkin özellik ve kapasite veri girişlerini sağlayacak ara yüzler geliştirilecektir. Bu kapsamda istenen özellikleri sağlayan bir paket programla (MapInfo gibi) ilişkilendirilecek bir modül geliştirilecektir.



### **5.5.1. İlgi Sahası (Harita Projesi) Yaratma/Seçme/Silme Ara Yüzü**

Coğrafi Bilgi Sisteminde açılacak bir menüden “İlgi Sahası” seçilerek açılacak ara yüz yeni jenerik/gerçek ilgi sahası oluşturulmasına, mevcut bir ilgi sahası seçilmesi ya da silinmesine imkân sağlayacak işlemlere sahip olacaktır. Bu işlemle Kullanıcının üzerinde çalıştığı birkaç sorumluluk sahasının birbirleriyle ilişkilendirilerek bir grup haline getirilmesi ve haritalar/bölgeler arasında geçişin kolaylıkla sağlanmış olacaktır. Kullanıcıyı ilgilendiren ancak sorumluluğunda olmayan bölgeler kullanılması gerektiğinde bu sayede daha etkin kullanım ve hazırlanacak taşıma planında kullanılacak bölgenin tanımlanması sağlanmış olacaktır.

### **5.5.2. Etki Sahası (Ayrıntılı Harita) Yaratma/Seçme/Silme Ara Yüzü**

Coğrafi Bilgi Sisteminde açılacak bir menüden “Yeni Etki Sahası Oluştur” ifadesi seçilecek, harita üzerinde fare ile dikdörtgen şeklinde taranarak bir bölge belirlenecek, fare serbest bırakıldığında katmanlar ekrana gelerek taranan alan detaylarının (dağlar, sınırlar, nehirler, şehirler, yollar v.b.) seçilmesine imkân tanıyacak, bu işlemi müteakip, “TAMAM” düğmesine tıklanarak açılan ara yüze yeni etki sahasının ismi yazılacak ve bu harita, seçili ilgi sahasının üzerinde farklı bir renk koduyla dikdörtgen şeklinde görülecektir. Yine menüden seçilecek “Etki Sahası Seç” ve “Etki Sahası Sil” düğmeleri ile aktif durumdaki etki sahası silinebilecek veya başka bir etki sahası ile değiştirilebilecektir. Bu işlemle aslında Kullanıcının üzerinde çokca çalışılan bir bölgeye ilişkin alt yapı ve lokasyon verilerinin her defasında yeniden filtrelenerek harita üzerinde görüntülenmesi yerine bir kez yapıldıktan sonra şablon (template) olarak kaydedilmesi ve kolayca erişilmesi sağlanmış olacaktır.

### **5.5.3. Coğrafi Lokasyon Veri Giriş Ara Yüzü**

Bu ara yüze, menüden seçilecek “Lokasyon Ekle” düğmesine tıklanıldığında, fare şekil değiştirecek ve harita üzerinde istenen bir noktaya tıklanmak suretiyle ulaşılabilecek ve bu ara yüz, asgari olarak coğrafi lokasyon kodu, sıra no, Lokasyon İsmi, Ait Olduğu Ülke, Coğrafi Lokasyon Tipi (şehir, uluslar arası havaalanı, liman, gar v.b.), seçilmiş olan noktanın koordinatları (kod şekli-584051N0241345W gibi olacak) (manuel olarak değiştirilebilir), Bölge İsmi, lokasyona ulaşabilen yolların ulaştırma modelleri (kara yolu, hava yolu v.b.) Açıklamalar gibi alanları içerecektir.

#### 5.5.4. Coğrafi Lokasyon Veri Giriş Alt Ara Yüzleri

Coğrafi Lokasyon Veri Giriş Ara Yüzü bir kez oluşturulup, onaylandıktan sonra haritada yeni oluşturulan coğrafi lokasyona tıklanarak açılacak ara yüz coğrafi lokasyonun ulaştırma altyapısı (liman, havaalanı, istasyon v.b.) olması durumunda sekmelerle ilave verilerin girilebileceği yeni alanlar bulunacaktır.

Coğrafi lokasyonun bir demir yolu işletme altyapısı (gar, istasyon v.b.) olması durumunda coğrafi lokasyon verilerinin yanında İstasyon Tipi, Kolaylıklar (Konteyner elleçleme, elektirifikasyon v.b.), Yan Rampa Sayısı, Baş Rampa Sayısı, Rampa Uzunluk, Genişlik ve Yüksekliği, İstasyon Tipi (Yolcu, Yük, Maden v.b.), Elleçleme Teçhizatları ve Kapasiteleri (Text Veri), Elleçleme operasyon tipi (liman boşaltma v.b.), Demir yolu noktasına ulaşan diğer ulaştırma modları, Kabul Edebileceği Tren Maksimum Tonajı, Kabul Edebileceği Tren Uzunluk, Genişlik ve Yüksekliği, Somaja Kalma Süresi, Yükleme/Boşaltma Personeli Mesai Başlangıç ve Bitiş Saatleri, Kullanım Maliyeti (saat bazında) gibi alanları içerecektir.

Coğrafi lokasyonun bir liman olması durumunda coğrafi lokasyon verilerinin yanında Liman Tipi, Kolaylıklar (Yakıt İkmali, Ağır Yük Yükleme Boşaltma, Mühimmat Yükleme Boşaltma v.b.), Maksimum Gemi Kiriş Yüksekliği, Maksimum Uzunluk, Maksimum Draft, Rıhtım Sayısı, Rıhtım No/Adı, Terminal İsmi, Rıhtım Tipi (RO-RO, Kuru Yük, Tren RO-RO v.b.) Maksimum Rıhtım Gemi Kiriş Yüksekliği, Maksimum Rıhtım Gemi Uzunluğu, Maksimum Rıhtım Gemi Draftı, Maksimum Rıhtım Deadweight, Tehlikeli Madde İzni (Evet/Hayır), Apron Genişliği ve Apron Yüksekliği, Demir yolu Bağlantısı Olup-Olmadığı, Elleçleme Teçhizatları ve Kapasiteleri (Text Veri), Elleçleme operasyon tipi (liman boşaltma v.b.), limana ulaşan diğer ulaştırma modelleri, Kabul Edebileceği Gemi Maksimum Tonajı, Kabul Edebileceği Gemi Uzunluğu, Genişliği ve Yüksekliği, Yükleme/Boşaltma Personeli Mesai Başlangıç ve Bitiş Saatleri, Kullanım Maliyeti (saat bazında) gibi alanları içerecektir.

Coğrafi lokasyonun bir havaalanı olması durumunda coğrafi lokasyon verilerinin yanında, Havaalanının deniz seviyesinden yüksekliği, Hangi Hava Organizasyonu (Sivil/Askeri) tarafından idare edildiği, Uçak Tiplerine Uygunluğu (Büyük Gövdeli Askeri Uçak-C141B, Jumbo Gövdeli Askeri Uçak-C5, Büyük Gövdeli Sivil Uçak, Dar Gövdeli Sivil Uçak v.b.), Gece Operasyon Sınırlılığı (evet/hayır), Uçak Park Kapasitesi (Maksimum Geniş Gövdeli Uçak sayısı, Maksimum Dar Gövdeli Uçak Sayısı v.b.), Elleçleme Kapasitesi (Maksimum Geniş Gövdeli Uçak

için, Maksimum Dar Gövdeli Uçak için v.b.) Pist Verileri (Pist (ID) No., Pist Uzunluğu ve Geniřlięi, Pistin Mevcut Durumu, Yüzey Cinsi, Pistin Aęırlık Tařıma Kapasitesi (WBC-Weight Bearing Capacity)), Uçak Park Alanı Verileri (Park Apron (ID) No., Apron Uzunluğu ve Geniřlięi, Apronun Mevcut Durumu, Apron Yüzey Cinsi, Apron Aęırlık Tařıma Kapasitesi (WBC)), Elleęleme Teęhizatları ve Kapasiteleri (Text Veri), Elleęleme operasyon tipi, Havaalanına ulařan dięer ulařtırma modelleri, Kabul Edebileceęi Uçak Maksimum Tonajı, Kabul Edebileceęi Uçak Uzunluğu, Geniřlięi ve Yükseklięi, Somaja Kalma Süresi, Yükleme/Bořaltma Personeli Mesai Bařlangıç ve Bitiř Saatleri, Kullanım Maliyeti (saat bazında) gibi alanları içerecektir. İhtiyaç duyulabilecek ilave veri alanları veri tabanı tasarım ařamasında Kullanıcı tarafından Geliřtiriciye verilecektir.

Yukarıda sayılan ulařtırma altyapıları dıřındaki tüm coęrafi lokasyonlar için Elleęleme Teęhizatları ve Kapasiteleri (Text Veri), Elleęleme faaliyet tipi, coęrafi lokasyona ulařan dięer ulařtırma modları, Kabul Edebileceęi Maksimum Tonaj, Kabul Edebileceęi Maksimum Uzunluk, Geniřlik ve Yükseklik, Somaja Kalma Süresi, Yükleme/Bořaltma Personeli Mesai Bařlangıç ve Bitiř Saatleri ve Yakıt İkmal İmkanı, Kullanım Maliyeti (saat bazında), Yakıt Tipi alanlarını içerecektir. İhtiyaç duyulabilecek ilave veri alanları veri tabanı tasarım ařamasında belirlenecektir.

#### **5.6. Ulařım Aęı Verileri Giriř Ara Yüzü**

Sistem, kullanacaęı paket CBS programı (MapInfo gibi) yardımıyla dijital haritalar üzerindeki katmanları kullanarak, ilgi sahası içindeki her türlü ulařtırma aęını görüntüleyebilecek, ilgi sahası içinde yer alan her hangi bir noktaya kendisi için özellik arz eden verileri girebilecek ve/veya daha önce girilmiř verileri silebilecektir. Bu deęiřiklikler sistemin varsayılan (default) deęerlerini etkilemeyecektir. Kullanıcı bu kapsamda belirlenen noktanın video görüntülerini, noktaya ait ipuçlarını, özellikleri gösterir ifadeleri ya da resimleri, uydu resimlerini, büyük ölçekli raster (resim) haritaları, uydu görüntülerini (Google Earth vb.) sisteme girebilecektir. Bunun yanında raster harita sistemi kullanılarak; her hangi bir ulařtırma moduna ait manuel ulařtırma aęı oluřturulması/çizilmesi, silinmesi, deęiřtirilmesi iřlevleri de sistem tarafından geręekleřtirilecektir. Bu çerçevede yurt içi ve yurt dıřında Karayolları, Demiryolları, Hava yolu FIR (Flight Information Route) Hatları, ve Deniz yolu aęlarının seęilen Tařıma Çıkıř Noktası (TÇN), Tařınacak Birim Tařıma Bitiř Noktası (TBN), kontrol noktaları, liman, havaalanı, istasyon v.b. arasında parça parça

manuel olarak çiziminin yapılabileceği ve bu çizimler üzerinde yer aldığı ulaşım hattında (LOC-Line of Communication) her bir parçaya tıklanarak açılan ara yüze o hatta bulunan kısıtlama (En Düşük kapasiteli Köprü Yük Sınıfı, En düşük gabarili Tünel Azami Yüksekliği, Köprü ve Tünelin faallik durumu v.b.) verilerinin girilebileceği bir ara yüz geliştirilecektir.

Veri Tabanına girişi yapılan ya da CBS paket programı katmanları içinde yer alan ulaşım ağları bilgilerinin planlamada kullanılacak gerçek/jenerik ilgi sahasına atanmasını sağlayacak bir ara yüz geliştirilecektir.

### **5.7. Taşıma Planlama ve Karar Destek Sistemi (TPM)**

Geliştirilecek bu karar destek sistemi/sistemleri; genel müdürlükler (yurt içi ve yurt dışı) seviyesinde uluslar arası stratejik ulaştırma araçları (örn.uçak, gemi), bölge müdürlükleri seviyesinde bölgeler arası ulaştırma araçları (örn. ağır araç taşıyıcı, konteyner taşıyıcı v.b.), il müdürlükleri seviyesinde mücavir iller arası ulaştırma araçları cins ve miktarları ile taşımaya esas yolcu malzeme cins ve miktarlarının takip edildiği bir planlama yapma imkânına sahip olacaktır.

Sistem/sistemler, taşınacak birimleri gerektiğinde taşıma bileşenlerine (component) ayırabilecek ve bu bileşenlere yolcu ve yük tahsis edebilecektir.

Sistem/sistemler, oluşturulan plana uygun Taşıma İhtiyaç Listesini esas alan, ulaştırma araçlarının TÇN'den (homebase), TBN'ye (final destination) gitmek için kullanabilecekleri ve boş olarak aynı ya da başka bir noktaya (homebase) geri dönen araçların takip edebilecekleri tüm ulaştırma modellerini içeren ana ulaştırma ağını (kullanılabilir olan bütün yollar, yükleme/boşaltma terminal noktaları, aktarma noktaları, kontrol noktaları) ve ana ağ üzerindeki alternatif güzergâhları [alt ağların, yükleme/boşaltma terminal noktaları (Port of Embarkation/Debarcation), kontrol noktaları] Coğrafi Bilgi Sistemi altlıklı haritalar üzerinde oluşturabilecektir.

Araç Havuzuna konmak üzere envanterde bulunan ya da envanter dışından temin edilebilecek ulaştırma araçları ve bu araçların hazır bekleme noktaları (bağlama limanı, terminali v.s.) seçilebilecektir (ulaştırma araçlarının ilk tahsis yerleri tespit edilecektir).

Oluşturulan ulaştırma ağının özelliklerine bağlı olarak (köprü yük sınıfı-tünel/üst geçit yüksekliği genişliği, havaalanı LCN değeri, limanın draftı v.b.) ulaştırma araçları tarafından ağın kullanılabilirliğini tespit edecek, ulaştırma altyapısı

analizi yapılacak ve ihtiyaç duyulduğunda Kullanıcı tarafından ulaştırma ağıının kullanılabilirliğine ilişkin ilave kısıtlama getirilebilecektir.

Sistem/sistemler, taşınacak birimlerin, oluşturulan ulaştırma ağı veya seçilecek alternatif güzergâhlar üzerinde, envanterde bulunan ve envanter dışından temin edilecek ulaştırma araçları kullanılarak, TÇN'den TBN'ye en maliyet etkin şekilde varmasını sağlayacak, asgari aşağıdaki işlevleri otomatik olarak uygulama yeteneğine sahip olacaktır :

- Seçilecek plana ait ihtiyaç duyulan ulaştırma araçlarının tip ve sayıları işlenecek, analiz edilecek ve raporlanabilecektir.
- Ulaştırma araçlarının takip edecekleri güzergâhlar ve kullanım zamanları (sefer/sorti sayıları) analiz edilecek ve raporlanabilecektir.
- Ulaştırma araçlarının taşınacak birimler tarafından kullanım durumu analiz edilecek ve raporlanabilecektir.
- Ulaştırma araçları tarafından taşınacak birim envanter cins ve miktarları analiz edilecek ve raporlanabilecektir.
- Taşınacak birimlerin takip edecekleri güzergâhlar analiz edilecek ve raporlanabilecektir.
- Ulaştırma altyapısının taşınacak birimlere tahsisi analiz edilecek ve raporlanabilecektir.
- Taşınacak birimlerin taşınma başlama (TBZ) ve taşınma tamamlama zamanları (TTZ) analiz edilecek ve raporlanabilecektir.
- Ulaştırma altyapılarının taşımaların belli bir anında ya da tüm taşıma faaliyetleri boyunca kullanım durumu analiz edilecek ve raporlanabilecektir.
- Taşınamayacak (oversize veya diğer özellikleri nedeniyle) malzeme analiz edilecek ve raporlanabilecektir.

Geliştirilecek olan Karar Destek Sistem(ler)i/Modelleri asgari olarak aşağıdaki verileri kullanacaktır. Bu verilere ait kriterler sistem tasarım aşamasında Kullanıcı tarafından Geliştiriciye verilecektir.

- TÇN verileri kullanılacaktır.
- TBN verileri kullanılacaktır.

- Taşınacak birimlerin hazırlık süresi (en erken taşınmaya başlanabilecekleri zaman) verileri kullanılacaktır
- Taşınacak birimlerin taşımaya esas yolcu sayıları ve yük ağırlık, hacim ve miktar verileri kullanılacaktır.
- Taşınacak birimlerin son noktada bulunmaları gereken en erken ve en geç zaman (CRD-Customer Required Date) verileri kullanılacaktır
- Araçların yükleme ve boşaltma zamanları verileri kullanılacaktır.
- Araçların yollar üzerinde taşıma süresi verileri kullanılacaktır.
- Bekleme yerlerinde bulunan ulaştırma araçları cins ve sayıları (Filo) verileri kullanılacaktır.
- Envanterdeki ve envanter dışından uzun süreli temin edilebilecek araçların kiralama, işletme maliyetleri (gün, saat ya da km bazında) verileri kullanılacaktır
- Ulaştırma araçlarının taşıma kapasiteleri (ağırlık, hacim ve yolcu bazında) hızı, hareket sırası gibi, aracın yüklenmesi ve ulaştırma ağını kullanabilirliğine etki eden özellikleri kullanılacaktır.
- Ulaştırma altyapılarının kapasite kısıtları (örn. havaalanı LCN değeri, liman günlük gemi kabul kapasitesi, köprü yük sınıfı, yolların trafik yoğunluğu ve günlük araç kapasitesi v.b.) verileri kullanılacaktır.

Geliştirilecek olan Karar Destek Sistem/Sistemlerine ilişkin modellerde asgari olarak aşağıdaki kısıtlar uygulanacaktır.

- Envanterde bulunan ve dış kaynaktan temin edilecek (Outsourcing, 3PL) araçlardan oluşan ulaştırma filosundaki araç sayıları aşılmayacaktır. (Örneğin Kullanıcının elinde 5 (beş) adet 5 (beş) tonluk kamyon var ise, geliştirilecek sistem/sistemler çalıştırıldığında 6 (altı) ve daha fazla kamyon kullanımına izin vermeyecektir.)
- Geliştirilecek sistem/sistemler, taşınacak birimin G günü + taşıma hazırlık zamanından önce taşınmaya başlayamayacak şekilde planlama yapma yeteneğine sahip olacaktır.
- Geliştirilecek sistem/sistemler, taşınacak birimlerin son noktaya tespit edilen en erken varış ve en geç varış (CRD-Customer Required Date) zaman dilimi

içerisinde ulaşılmamasını sağlayacak şekilde planlama yapma yeteneğine sahip olacaktır.

- Geliştirilecek sistem/sistemler, ulaştırma araçlarının yolcu ve yük kapasiteleri aşılmayacak şekilde planlama yapma yeteneğine sahip olacaktır.

- Geliştirilecek sistem/sistemler, taşımaya esas yolcu ve yükün, taşımaya uygun ulaştırma araçları ile taşınması sağlanacak şekilde planlama yapma yeteneğine sahip olacaktır.

- Geliştirilecek sistem/sistemler, ulaştırma araçlarının yüklenme özelliklerine göre ihtiyaç duyulan yük/yakıt, yük/yolcu dengesi kurulacak şekilde planlama yapma yeteneğine sahip olacaktır.

- Geliştirilecek sistem/sistemler, ulaştırma araçlarının, ulaştırma ağı üzerinde boş ve dolu olarak gidiş gelişleri sağlanacak şekilde planlama yapma yeteneğine sahip olacaktır.

- Geliştirilecek sistem/sistemler, ulaştırma modelleri arasında yolcu malzeme transferi yapılan noktalarda eş zamanlama (senkronizasyon) yapılacak şekilde planlama yapma yeteneğine sahip olacaktır.

- Geliştirilecek sistem/sistemler, ulaştırma altyapısının ulaştırma araçları tarafından kullanılabilirliğine ilişkin kısıtlamalara uyulacak şekilde planlama yapma yeteneğine sahip olacaktır.

Karar destek sistemi ya da sistemleri kullanılarak hazırlanan taşıma planlarında bulunan hangi araca hangi malzemenin yüklendiği bilgisini esas alarak; spesifik araç bazında, Kullanıcı tarafından belirlenmiş kriterler doğrultusunda Otomatik/Manuel Yükleme Planlama Modülü geliştirilecektir.

Alt birimler tarafından gönderilen taşıma planlarının birleştirilmesi sonucunda ortaya çıkabilecek çatışmaların giderilmesi ya da aynı/yakın zamana ve/veya aynı güzergâha planlanmış yolcu/yükün, büyük kapasiteli ulaştırma araçlarında birleştirilmesini sağlayacak bir modül geliştirilecektir.

Karar destek sistemi ve yükleme planlama modülü kullanılarak hazırlanan taşıma planlarının uygulanabilirliğini görmek amacıyla taşıma planlarını baz alan coğrafi bilgi sistemi altlıklı bir animasyon programı geliştirilecek, bu animasyon programı asgari aşağıdaki fonksiyonları ihtiyaç durumunda uygulayabilecektir:

- Taşıma Planının dayandırıldığı bir zamana (G Günü gibi) ve/veya planlandığı gerçek zamana göre uygulaması, taşınacak birimlerin ve ulaştırma araçlarının (ikonlarla hangi tip ulaştırma aracı olduğu görülecek) hareketleri, planın uygulandığı an ve o andaki taşınmış ve taşınmakta olan yolcu yük toplamları, hareket halindeki bir ulaştırma aracı, bu aracın üzerindeki taşıma bileşeni (orijinal veya alt bileşen) ve manifesto istendiğinde hareket durdurulmak suretiyle görülecektir.

- Planın uygulandığı herhangi bir anda, belirlenecek bir zaman aralığında ya da planın icrası süresince; bir ulaştırma altyapısını (liman, havaalanı, kara yolu, istasyon) ne kadar ulaştırma aracının kullandığı (örn. bir havaalanında kaç uçak yükleme/boşaltma faaliyeti yürütüyor.), altyapıya ulaşan ya da altyapıdan ayrılan yolcu/yük miktarları ile yük cinslerinin toplu bir listesi veya grafiği gibi bilgiler istendiğinde hareket durdurulmak suretiyle görülecektir.

Taşıma Planlama ve Karar Destek Sistemi ile en az aşağıdaki çıktılar üretilebilecektir:

- Taşıma Görevleri Listesi: Belli bir taşıma planıyla taşınacak birimlerin ve bu birimlerin bölündüğü bileşenlerin (movement component) hiyerarşik olarak asgari TÇN, (taşımanın başlayacağı terminal/liman/havaalanı/istasyon isimleri), TBN (taşımanın bittiği terminal/liman/havaalanı/ istasyon isimleri), taşıma öncelikleri ve son noktada bulunması istenen en geç zaman (CRD-Customer Required Date) gibi bilgileri içeren rapor sistem tarafından üretilecektir.

- Taşınacak Birim Taşıma Çizelgesi/Grafiği: Taşınacak birimlerin ve bu birimlere ait bileşenlerin; taşıma planı kapsamında, taşıma çıkış ve bitiş noktaları arasındaki ardışık (successive) tüm taşımalarının ve/veya belli bir ulaştırma moduna (hava yolu, deniz yolu vb) ilişkin taşımalarının ve bu taşıma bacaklarına ulaştırma aracı planlanmış olup olmadığının listelendiği sorgunun haritalar üzerinde çıktısı alınabilecektir.

- Taşınacak Birim Taşıma İhtiyaç Çizelgesi: Taşınacak birimlerin ve bu birimlere ait bileşenlerin; envanterinde bulunan taşımaya esas yolcu ve yüklerin taşınmasında Taşınacak Birim Taşıma Çizelgesinde belirtilmiş her bir bacak için planlanmış ulaştırma araçlarına miktar ve zamanca olan ihtiyaçlarının her bir bacak için ayrı ayrı ve tüm taşımalar için toplam olarak belirlendiği çizelge çıktısı alınabilecektir.



- Taşınacak Birimlerin Taşınmasında Kullanılacak Envanter Dışı Ulaştırma Araç İhtiyaç Çizelgeleri: Taşınacak Birim Taşıma İhtiyaç Çizelgesinden, Kullanıcı envanterinde bulunan araçlar dışındaki araçların taşıma planına tahsis edildiği satırların sorgulanması ile taşınacak birim taşınmasında kullanılacak envanter dışı taşıma araç ihtiyaç çizelgesi elde edilecektir. Bu çizelgede, her bir bacak için planlanmış taşıma araçlarına miktar ve zamanca olan ihtiyaçların her bir bacak için ayrı ayrı ve tüm taşımalar için toplam olarak belirlendiği bir çizelge olacaktır.

- Ulaştırma Altyapısı Kullanım Durumu Çizelge/Grafiği: Herhangi bir seviyede (genel müdürlük, bölge müdürlüğü, il müdürlüğü, şube) hazırlanmış olan taşıma planında yer alan çıkış ve varış liman, hava alanı, TIR Konaklama noktası, gar/istasyon gibi günlük kullanım kapasiteleri belirli coğrafi lokasyonların belirli bir anda ve/veya zaman diliminde ve/veya bir plan kapsamındaki tüm taşımaların başlangıcından, sonuna kadar geçecek süre içerisindeki (örn. G+15 gününde İzmir Limanının, G+3 ila G+28 günleri arasında Diyarbakır Havaalanının) kullanım yoğunluklarının (Kullanıcı ve gerektiğinde DHMİ web alanından alınacak verilerle kendi veri tabanını güncelleyebilecektir), taşıma aracı ve yolcu sayıları, yük tonajı bazında görüldüğü ve kapasite aşımalarının ya da boş kapasitelerin gösterildiği çizelge ve grafik. Taşıma planının Coğrafi Bilgi Sistemi altlıklı haritalarda simülasyonu esnasında da, simülasyon durdurulmak ve ilgili altyapı bilgileri açılmak suretiyle bu grafik rapor görülebilecektir.

- Ulaştırma Altyapısı Çatışmalarının Giderilmesi (Deconfliction) Çizelge/Grafiği: Ulaştırma altyapısı kullanım durumu çizelgesinde belirlenen altyapı kapasite aşımalarının, boş kapasiteli günlere, taşıma planında son noktaya varılması istenen en son zaman (CRD-Customer Required Date) konusunda bir sorun yaşanmaksızın otomatik olarak aktarıldığı çizelge/grafiktir. Taşıma planının Coğrafi Bilgi Sistemi altlıklı haritalarda simülasyonu esnasında da, simülasyon durdurulmak ve ilgili altyapı bilgileri açılmak suretiyle bu grafik rapor görülebilecektir.

## **5.8. Kurye/Ring Yolcu/Yük Planlama ve Karar Destek Sistemi (KURPM)**

Maliyet etkin bir kurye/ring yapısı oluşturmak amacıyla yeni kurye/ring seferleri ihdas edilmesinde ve mevcut seferlerinin birleştirilmesi veya uygulamadan kaldırılmasında karar desteği sağlayacak ve ihdas edilmiş kurye/ring seferlerine yönelik taleplerin maliyet-etkin olarak karşılanmasının planlanmasını sağlayacak iki ana fonksiyon içeren bir sistem geliştirilecektir.

### **5.8.1. Kurye / Ring Planlama Modülü Veri Giriş Ara Yüzü**

Kurye/ring seferi planlamasında kullanılmak üzere, her yöndeki güzergâhlarda ve/veya birbiri üzerine bindirmeli güzergâhlarda yapılan münferit (yolcu/yük, konteynerli) ve mevcut kurye/ring seferlerine ilişkin yolcu/malzeme akışlarının miktar ve tiplerinin, bu taşımalarda kullanılan ulaştırma araçlarına ilişkin verilerin girişini sağlayacak bir ara yüz geliştirilecektir.

Bu ara yüzde, aracı çıkaran/kiralayan birim, nereden nereye taşıma yapıldığı, takip edeceği güzergâh, taşınan yükün cins ve miktarı, aracın cins ve miktarı, taşımanın yapıldığı zaman, aracın görevlendirme/kiralama sebebi, açıklamalar gibi alanları içerecektir. Ayrıca, Pasif İz Takip yeteneğine sahip terminallerden gelen verilerle de veri tabanı (Yolcu/Malzeme Akış Verileri) otomatik olarak güncellenecektir.

### **5.8.2. Kurye / Ring Görüntüleme ve Kurye/Ring Talebi Alt Sistemi**

Kurye/Ring Seferi Verileri Tablosunda tanımlanan kurye/ring seferlerinin dökümünün ulaştırma modu ana kategorileri altında görüntülenebileceği ve bunlardan birisi seçildiğinde seçilen kuryenin özellikleri, bacaklarının durumu ve o ana kadar yapılmış olan (yetkili kurye planlamacısı tarafından onaylanmış ya da onaylanmamış) talepler sonucu bacak bacak onaylı/onaysız doluluk durumu, taşıma talebi yapılacak yolcu ve/veya malzeme özellikleri ve taşıma yapılması istenen bacak/tarih veri girişlerinin yapılacağı ve talebin yetkili üst birimler tarafından onaylanarak (birleştirilerek) bir üst birime gönderilmesini sağlayacak onay sürecinin yer alacağı ve talep yapan birimin talebinin mevcut statüsünü (onaylanmış talep, yedek talep, onaylanmaması durumunda muhtemel aktarılacağı kurye/ring seferi v.b.) görebilmesini ve talebini iptal etmesini ya da alternatif başka bir kurye/ring seferine kaydırmasını sağlayacak şekilde bir alt modül geliştirilecektir.

Bu alt sistem yazılımı ana ara yüzü, asgari olarak yeni kurye/ring yolcu talebi girişi, yeni kurye/ring malzeme talebi girişi, yeni kurye/ring malzeme-refakatçi talebi girişi, girilmiş kurye/ring talebinde değişiklik, kaydetme, silme, yazdırma, alt birimden gönderilen talebin yetkili üst taşıma birimi tarafından onaylanarak (diğer ast taşıma birim talepleri ile birleştirilerek) bir üst taşıma birimine gönderilmesi onay düğmesi ve ara yüzden çıkış düğmeleri olacak ve yine asgari olarak, ara yüzde Kurye/Ring Seferleri Verileri ana kategori olacak ve altlarında Kurye/Ring Seferi Verileri Tablosunda tanımlanan seferlerle ilişkilendirilmiş planlı Kurye/Ring Seferi verileri gösterilecek ve ara yüzden seçilen Kurye/Ring Seferi özellikleri ile taşınacak birime,

o ana kadar yapılmış olan (yetkili kurye planlamacısı tarafından onaylanmış ya da onaylanmamış) talepler, bu taleplerin yükleme öncelik sırası (Yükleme Öncelik Sırası Tablosu ile ilişkilendirilmiş), bu talepler sonucu bacak bacak onaylı/onaysız doluluk durumu, taşınma talebi yapılacak yolcu ve/veya malzeme/malzeme-refakatçi düğmesi ve taşıma yapılması istenen bacak/tarih veri giriş kayar menüleri (combo box) ve [talep yapan birim talebinin mevcut statüsünü (onaylanmış talep, yedek talep, onaylanmaması durumunda muhtemel aktarılacağı kurye/ring seferi v.b.) görebilmesi ve talebi iptal etmesini ya da alternatif başka bir kurye/ring seferine kaydırması sağlanacaktır] ve açıklamalar alanlarını içerecektir.

Bu ana ara yüz üzerinde yer alan yolcu/malzeme taşıma talebi düğmesine tıklanıldığında açılan ara yüzde yük tipi bölümündeki kayar menüden yolcu, malzeme ve/veya malzeme-refakatçi seçildiğinde giriş ekranı değişecek; yolcu seçiminde, Talebi Yapan Taşıma Birimi Mesaj Adresi, Adı Soyadı, Görevi, Kurye/Ring Seferi Kod No., Yolcu Kod No., Yolcu Öncelik Kategorisi (VIP, Business Class, Birinci Sınıf, İkinci Sınıf) Taşıma Yapılacak Bacak/Bacaklar, Talep Yapılan Kurye Tarihi, Ağırlık (110 kg malzeme eşdeğeri olarak sabit giriş yapılmış olacaktır.), Hacim, Tercih Edilen Alternatif Kurye/Ring Seferi ve Talebin Kıymetten Düşme Zamanı gibi alanları içerecek; malzeme seçiminde, Talebi Yapan Birim Mesaj Adresi, Malzemenin Cinsi, ebatları (En, Boy, Yükseklik, Ağırlık) (Spesifik malzemeler Envanter Tablosu ile ilişkili kayar menüden otomatik olarak seçilecek, diğer paket/çuval/ sandık tipi malzemelerin hacim, ağırlık bilgileri manuel girilerek), Malzeme Cinsine göre öncelik sırası (Yükleme Öncelikleri Tablosundan Otomatik olarak girecek), Kurye/Ring Seferi Kod No., Taşıma Yapılacak Bacak/Bacaklar, Talep Yapılan Kurye Tarihi, Tercih Edilen Alternatif Kurye/Ring Seferi, Talebin Kıymetten Düşme Zamanı ve varsa refakatçi bilgileri (yolcu bilgisi girer gibi) gibi alanları içerecektir.

Ana ara yüz üzerinde yer alan ast taşıma birimlerinden gönderilen talebin yetkili üst planlamacılar tarafından onaylanarak (diğer ast taşıma birim talepleri ile birleştirilerek) bir üst taşıma birimine gönderilmesi, onay düğmesine tıklanıldığında ast taşıma birimlerinden yapılan talepler yolcu ve malzeme bazında listelenecek ve her bir münferit talebin önünde onay kutucukları bulunacak; onay kutucukları işaretlenip, gönder düğmesine basıldığında talebin bir üst taşıma birimine iletilmesini sağlayacak bir ara yüz geliştirilecektir.

### **5.8.3. Yeni Kurye/Ring Seferleri İhdası, Mevcut Kurye/Ring Seferlerinin Analizi, Birleştirilmesi ve Uygulamadan Kaldırılması**

Kurye/ring seferleri işletme sisteminin maliyet-etkin olarak çalıştırılabilmesi için, Kurye/Ring Seferlerinin ihdas edilmesinde karar desteği sağlayacak bir karar destek sistemi geliştirilecektir.

Bu sistem, aynı ve farklı ulaştırma modellerine ait ulaştırma araçlarının, var olmayan (yeni) bir kurye/ring seferinin hangi sıklıkla, hangi güzergâh ve bacaklarda uygulanacağı (kurye/ring seferi ihdas etme), var olan kurye/ring seferlerinden hangilerinin birleştirilebileceği veya uygulamadan kaldırılacağı ya da güzergâhlarında ne gibi iyileştirmeler yapılacağı konularında maliyet-etkinliği esas alan karar desteği sağlayacak şekilde geliştirilecektir.

Bu sistem, belirli bir güzergâhta ve/veya birbiri üzerine bindirmeli güzergâhlarda yapılan münferit (yolcu/yük merkezli, konteynerize edilmiş) taşımaların belirli bir periyot içindeki (yolcu/yük) miktar ve tipleri ile yürütülmekte olan kurye/ring seferlerinin doluluk durumları ve taşınan yolcu/yük miktar ve tipleri gibi istatistiksel verileri kullanarak, kurye/ring planlaması yapılmasını sağlayacaktır.

Geliştirilecek karar destek sistemi ya da sistemlerinin, kurye/ring seferi planlama sürecinde kullanılacak ara yüz ve sistem özelliklerine ilişkin ayrıntılar aşağıdaki maddelerde açıklanmıştır :

- Yurt dışı ve yurt sathında ve/veya ihtiyaç duyulan daha dar bir bölgede her yöndeki yolcu ve malzeme akışlarının gerçekleştiği depo, kontrol noktaları ve terminaller ile bunları birbirine bağlayan ve planlaması yapılacak olan ulaştırma model/modellerine ilişkin güzergâhları içeren kurye/ring ulaştırma ağı haritalar üzerinde oluşturulabilecektir.

- Geliştirilecek sistem ihtiyaç duyulduğunda, oluşturulan kurye/ring ana ulaştırma ağı üzerinde belli noktalar (ülkeler, depolar, terminaller v.b.) arasında kurye/ring seferi ihdas edilmesine, ihdas edilecek kurye/ring seferleri için güzergâhlar tespit edilmesine imkân sağlayacaktır.

- Kullanıcı envanterinde bulunan ya da envanter dışından temin edilebilecek ulaştırma araçları ve bu araçların hazır bekleme noktaları (bağlama limanı, terminali v.s.) seçilebilecektir (Kurye/ring seferini icra edebilecek olan en uygun ulaştırma araçları tespit edilecektir).

- Oluşturulan ulaştırma ağı özelliklerine bağlı olarak (köprü yük sınıfı-tünel/üst geçit yüksekliği genişliği, havaalanı LCN değeri, limanın draftı v.b.) ulaştırma araçları tarafından ağın kullanılabilirliği tespit edilebilecek, ulaştırma altyapısı analizi yapılabilecek ve ihtiyaç duyulduğunda Kullanıcı tarafından ağın kullanılabilirliğine ilişkin ilave kısıtlama (güvenlik, gizlilik v.b.) girilebilecektir.

- Kurye/ring seferi planlamasına etki edeceği değerlendirilen, planlamanın ayrıntısına ve yapıldığı seviyeye bağlı olarak taşınacak malzeme tiplerinin, malzeme tiplerine ilişkin model/modeller tarafından kullanılacak verilerin özelliklerinin (Örn. Kurye seferinin planlanmasında iş makinası cinsi malzemenin yıllık toplam taşınan miktarı, günlük taşınan miktarları, belli zaman aralıkları için ortalama taşınan miktarları) seçilerek, veri tabanından (Kurye/ring Seferi Planlamasına Esas Yolcu/Yük Akış Verileri Tablosundan) çekilmesini sağlayacak bir alt modül geliştirilecektir.

- Geliştirilecek model/modellerin çalıştırılması sonucunda elde edilen sonuçlar ve/veya Kullanıcı'nın manuel olarak yapmaya ihtiyaç duyduğu değişikliklerin onaylanmasını müteakip, Uygulanmakta Olan Kurye/Ring Seferleri Tablosu otomatik olarak güncellenecektir.

#### **5.8.4. Kurye/Ring Seferlerine Yönelik Taleplerin Karşılanmasının Planlanması**

- Kurye / Ring Seferi Yolcu/Yük Planlama ve Yönetimi: Bu modül, yetkili talep birimleri tarafından veri tabanı yönetim modülü (VTYM) kullanılarak hazırlanmış ve yetkili ara birimlerce onaylanmış olan kurye/ring taleplerinin yolcu/yük önceliklerini dikkate alarak araca otomatik olarak yüklenmesini; bu taleplerin mevcut statüsünün (onaylanmış, yedek v.b.) talep makamları tarafından görülmesini ve planlamacının, talepleri kalem kalem ve/veya tümünü onaylayarak ve kapasiteyi aşan talepleri Kurye / Ring Seferleri Arası Yolcu/Yük Aktarma ve Yönetim Modülüne gönderilmesini sağlayacaktır. Bu modül ile, hazırlanan Kurye/Ring planları WEB ortamında görülebilecektir. Kurye/ring planlaması yapacak terminal birimleri, kurye/ring seferinin tamamlanmasından sonra planlama ve uygulama farklılıklarını (gelmeyen yolcu/teslim edilmeyen yük gibi) sisteme bu modülü kullanarak girecek ve planlama birimine gönderecektir.

- Kurye / Ring Seferleri Arası Yolcu/Yük Aktarma ve Yönetimi: Yetkili talep birimleri tarafından VTYM kullanılarak hazırlanmış ve yetkili ara birimlerce onaylanmış olan ve yolcu/yük öncelikleri nedeniyle ilgili Kurye/ring seferinin kapasitesini aşan kurye/ring taleplerinin, talep yapılan birim tarafından belirtilen

alternatif kurye/ring seferine veya böyle bir sefer belirtilmemişse talebin planlama dışı bırakıldığı Kurye/ring Seferinin tarih ve güzergâhı ile talebin kıymetten düşme zamanına en uygun Kurye/ring Seferine talep yapılmış gibi yetkili planlamacının onayı ile aktarılmasını sağlayacak bir modül geliştirilecektir.

- Kurye / Ring Seferleri Etkinlik Analiz Modülü: Belli bir tarihteki belli bir kurye/ring seferinin kurye/ring taleplerinin sisteme girilmesi ile tüm sefer boyunca doluluk durumunun bacak-bacak ve/veya kümülatif hacim, ağırlık ve yüzde olarak görülmesi, kurye/ring seferinden faydalanan yolcu/yükün birim taşıma maliyetinin hesaplanması ve etkin olmayan taşıma konusunda planlamacının uyarılması ile belirli bir dönem (ay, mevsim, yıl, yıllar) boyunca bir kurye/ring seferinin ortalama doluluk durumu, kurye/ring seferi ortalama toplam maliyetinin hesaplanması ve aynı güzergâhta çalışan birden çok kurye/ring seferlerinin birbirlerine göre etkinlik analizlerinin yapılmasını sağlayacak bir modül geliştirilecektir. Hazırlanmış olan Kurye/Ring Planlarını alan ilgili Kurye/Ring aracının yüklenmesinden sorumlu terminal görevlilerinin, Kurye/Ring aracının ayrıntılı yükleme planlamasını otomatik olarak yapmaları amacıyla Otomatik ve Manuel Yükleme Planlama Modülü'nü Kurye/Ring planlamaları kapsamında da kullanmalarını sağlayacak bir yapı geliştirilecektir.

#### **5.9. Etkin Araç/Malzeme İz Takip Modülü (ETARİTM)**

Kullanıcı taşıma koordinasyon merkezinde bulunan bir sunumcuya otomatik olarak gönderilen enlem, boylam, yükseklik, UTC zamanı, aracın hızı ve yönü gibi konum verileri ETARİTM sistemi özellikleri kullanılarak Kullanıcı kontrolünde veri tabanına otomatik olarak aktarılacak ve güncellenmesi sağlanacaktır.

İnternet üzerinden anılan aracın, araç tipi ve özellikleri; sürücü/mürettebat kimlik bilgileri; aracın takip edeceği güzergâha ilişkin çıkış, varış, kritik nokta bilgileri; araç harekete başlamadan önce barkod okuyucular ve/veya Taşıma ve Kurye/Ring Planlama sisteminden almak suretiyle ve/veya (barkod okuyucuların bulunmadığı depo/terminallerde) manuel olarak yüklenebileceği bir ara yüz ve/veya yazılım geliştirilecektir.

Depo ve ilgili Kullanıcı birimlerinden ulaştırma terminallerine gönderilen malzemelerin, terminal deposuna alındığına ilişkin veriler ile terminal deposundan malzemelerin yükleneceği, taşıma görevi alan ulaştırma aracı ve araçta yüklü yolcu/yük verileri (Malzeme stok no., cinsi, miktarı, çıkış terminali, varış terminali, yükü teslim eden, yükü teslim alacak olan Kullanıcı birim bilgileri) terminal/depo giriş

ve çıkışlarındaki barkod okuyucular vasıtasıyla ve/veya kurye/ring planlama mesajları, ayrıntılı taşınacak birim taşıma planları vasıtasıyla VTYM'ye otomatik olarak girilecek ve bu bilgilerin seyahat süresince ve boşaltılıncaya kadar taşındığı araç bilgileri ile ilişkilendirilecek bir alt sistem geliştirilecektir.

Bu alt sistem sayesinde, takip edilmekte olan araca ilişkin veriler görülmek istendiğinde, iletişim altyapısından alınan araç iz bilgileri ile veri tabanından alınan yolcu/yük bilgileri otomatik olarak birleştirilerek, izleme yetkisine sahip planlamacı tarafından görülebilecektir.

Ulaştırma terminal depolarından teslim alınan ve/veya bu depolardan teslim edilen malzeme bilgileri ile ulaştırma aracına yüklenen (taşınmakta olan) ve/veya taşıma faaliyeti tamamlanıp terminal deposuna teslim edilen malzeme verileri, terminal deposunda bekleyen malzemeler, hareket halindeki malzemeler, teslim edilen malzemeler tablosuna barkod okuyucularla okumak suretiyle otomatik olarak aktarılacak ve veri tabanını güncelleyecek bir alt sistem geliştirilecektir.

Ulaştırma aracının taşıdığı yük cinsine göre (mühimmat, iş makinası v.b.) araçta bir renk kodlamasının otomatik olarak yapılabileceği bir alt sistem geliştirilecektir.

## **5.10. Test İşlemleri**

Uygulama yazılımlarının; tasarımıyla uyumluluğu, sistem kalitesinin kullanım için uygunluğu ve ortaya çıkabilecek problemler açısından test edilecektir. Ortaya çıkacak problemler ve ihtiyaçlara göre gerekli düzeltmeler yapılacaktır. Kullanıcı tarafından onaylanan test sonuç raporları muayene heyetine ibraz edilecektir. Geliştirici, sözleşmenin imzalanmasını müteakip 3 (üç) ay içerisinde yapacağı tüm testlerde kullanacağı aracı ve testlerin kriterlerini ayrıntılı olarak belirleyerek Kullanıcıya onaylatacaktır. Kullanıcının istemediği test aracı ve kriterler Kullanıcının istediği şekle getirilecektir.

### **5.10.1. Modül ve Entegrasyon Testleri**

Modüller arasında bilgi paylaşımı ile ara yüzlerin doğrulanması için Modül-Modül, Program-Program, Alt sistem-Alt sistem arasında gerekli testler Geliştirici tarafından gerçekleştirilecektir. Kullanıcı ihtiyaçlarına cevap vermeyen ya da hatalı bir durum olduğunda gerekli değişiklikleri yapılacaktır.

### **5.10.2. ALFA (Sistem) Testi**

Sistemin bütünü göz önünde bulundurularak aşağıda açıklanan Fonksiyon Kontrol ve Performans Değerlendirme testleri Geliştirici tarafından yapılacak, analiz ve tasarımda ortaya konulan kriterlere göre sonuçlar analiz edilip değerlendirilecek ve gerekli değişiklikler yapılacaktır.

Gerek modüllerin, gerekse programların amaçlanan fonksiyonları tam ve etkin olarak yerine getirdiği denenerek doğrulanacak, fonksiyon testleri aşağıdaki soru ve kıyaslamalara cevap verecektir.

- Veri tabanının büyüklüğü ve getirilen/sorgulanan kayıt sayısı arttıkça sistemin cevap süresi nasıl değişmektedir?
- Veri tabanına bir veya birden çok Kullanıcı aynı anda eriştiğinde cevaplama süresi nasıl değişmektedir?

Herhangi bir kaynağın aşırı yüklenmesi sonucu sistem performansında meydana gelebilecek belirgin düşmelerin olacağı noktalar tespit edilecektir. Bu seviyeler tespit edilirken Kullanıcı sayısı, donanım yapısı ve yapılan iş çeşidi göz önünde bulundurulacaktır.

### **5.10.3. BETA (Kurulum) Testi**

Modül Entegrasyon ve Sistem Testlerinden sonra ortaya çıkan değerlere uygun olarak gerçek ve örnek bilgi ve verilerle, gerçek Kullanıcı, donanım ve işletim ortamında tüm ihtiyaçların tam olarak ne oranda karşılandığının ortaya konması amacıyla icra edilecektir.

Bu deneme esnasında ortaya çıkan aksaklık veya eksiklikler, uygun olmayan performans ve dokümanlarla teyit edilenlerin dışındaki hususlar düzeltilecektir.

Bu pilot uygulama dönemi için gerekli eğitim ve dokümantasyon Geliştirici tarafından sağlanacaktır.

## **6. Ulaştırma Araçlarına İlişkin İşlevler**

### **6.1. Ulaştırma Aracı Oluşturma**

#### **6.1.1. Genel Araç Tipi Oluşturma**

Yazılım, hali hazırda Kullanıcı envanterinde veya mevcut piyasada kullanılan bütün taşıma araç model ve tiplerine ilişkin verilere sahip olacaktır. Gerektiğinde



Kullanıcı tarafından bu verilere yenileri eklenebilecek, veri giriş alanları artırılıp azaltılabilecektir.

### **6.1.2. Ulaştırma Aracı Oluşturma**

Yazılım Kullanıcının sisteme belirli bir araç modu/tipi üzerinden yeni kodlu taşıma aracı tanımlamasına imkân tanıyacaktır.

Oluşturulacak tüm araçlar için bir model/tip belirleme zorunluluğu kapsamında yazılım, veri tabanında mevcut araç modu/tiplerinden Kullanıcının yeni araçlar üretmesine imkân tanıyacaktır. Yeni oluşturulan araç için her hangi bir araç tipi seçilmemiş ise, yazılım belirlenen taşıma çeşidine (moduna) en uygun araç tipini mevcut listeden bularak otomatik olarak oluşturulan yeni araç ile ilişkilendirecektir.

Yazılım/Kullanıcı, oluşturulan taşıma aracı taşıma planı dahilinde otomatik/manuel tahsis edebilecek/yükleyebilecektir.

Kullanıcı yüklü bir taşıma aracı boşaltabilecektir.

Kullanıcı yüklü bir taşıma aracı tek adımda boşaltıp silebilecektir.

Yazılım, Kullanıcıya taşıma aracı ilişkin “Araç Kodu, Araç Tipi, Araç Tanımı, Ayrılış Tarih ve Zamanı, Varış Tarih ve Zamanı ve Mevcut Taşıma Programları (liste halinde)” veri alanlarını gösteren bilgileri sunacaktır.

Mevcut taşıma programları listesindeki veri alanları “Konuş Yeri, Hazırlık Durumu, Ayrılış, Varış, Kargo Yüzdesi, Yolcu Yüzdesi, Mesafe” bilgilerini içerecektir.

Kullanıcı tavsiye edilen ayrılış zamanlamasını düzenleyebilecektir.

Yazılım taşıma aracı yüklemesini optimize edecek ve en uygun taşıma planını verecektir.

Yazılım, tüm manifestoyu elleçleyebilecek bir işlevselliğe sahip olacaktır.

Yazılım, tahsis kurallarını uygulayabilecek bir algoritma kullanacaktır.

Yazılım, çok araçlı taşıma planlama faktörlerini kullanacak bir algoritmayı kullanacaktır.

Yazılım mevcut yolcu ve malzeme miktarlarını dikkate alarak, Kullanıcının tercih edeceği bir veya daha fazla taşıma aracı için (SPESİFİK ARAÇ TERCİHLERİ) bahse konu envanterin taşınmasına ilişkin verileri Kullanıcıya sunacaktır. (TAŞIMA PLANI TAŞINABİLİRLİK ANALİZİ YAPMA)

### 6.1.3. Çoklu Ulaştırma Aracı Oluşturma

Kullanıcı tek adımda birden çok araç oluşturup, bu araçlara envanter tahsisi yaparak yükleyebilecektir.

### 6.1.4. Ulaştırma Aracı Silme

Kullanıcı taşıma planı araç havuzundan bir veya daha fazla aracı seçerek bir defada silebilecektir.

Uygulama, taşıma planı araç havuzundan araç silinebilmesi için gerçekten silmek istiyor musunuz? gibi bir doğrulama yapacaktır. Şayet araç plan içerisinde kullanılıyor ve kargo tahsisi yapılmış ise, uygulama Kullanıcıyı aracın bazı taşımalara tahsis edildiği ve manifestosu bulunduğu konusunda uyaracaktır.

## 6.2. Araç Havuzu Oluşturma (Plana Araç Tahsis Etme) ve Araç Atama

Yazılım, taşıma planında kullanılmak üzere araç havuzuna araç eklemeye müsaade edecektir.

Yazılım; plana dâhil edilmiş taşıma araçlarına ilişkin aşağıdaki verileri istenildiğinde üzerine tıklanarak görüntülenebilecek bir ara yüz sunacaktır:

- Ulaştırma Modu (Transport Mod): Kara yolu (KK), Demir yolu (DM), Deniz yolu (DZ), Hava yolu (HV), İç Su yolu (IS). Bu hane doldurulduğunda "Araç Tip Kodu", "kategori" haneleri kayar menülerinde (combo box) o modele uygun araç çeşitleri otomatik olarak listelenecektir.

- Kategorisi (Category): Gemi (Kuru yük gemisi (KYG), Ro-Ro Gemisi (RRG), Tanker Gemi (TNG), Konteyner Gemisi (KNG), Ro-Pax Gemi (RPG), uçak, Panelvan (P)

- Araç Tip Kodu (Asset Type Code): Bu kod seçildiğinde "Markası ve Marka Uzantıları" hanesi otomatik olarak dolacak ve Kullanıcı bu alanlara her hangi bir giriş yapmayacaktır. Bu hane boş bırakıldığı takdirde "Üretici ve Marka Uzantıları" hanesine girilen verilere göre sistem kendisi bir kod oluşturacaktır. Bu kod mevcut kodlar içerisinde girilen veriye en yakın kodun yeni bir versiyonu olacaktır.

- Üretici ve Marka uzantıları (Maker and Mark Extensions)

- Araç Kodu (Asset Code): Ait olduğu birim ve seri numarasını da gösterecek şekilde yukarıda belirtilen veriler sistem tarafından otomatik olarak

tanımlanır. Örneğin Kullanıcının sahip olduğu tüm araç çeşitlerinin toplamı 25 ve bu araçlardan 7'nci çeşit aracın Ford Transit Panelvan olduğunu kabul edelim. İzmir bölgesi 3 No.lu Şube Müdürlüğünün elinde bu araçtan 105 adet olsun. Bizim kullandığımız araçta 101'inci araç olsun Bu bilgiler ve yukarıdaki veriler ışığında sistemin yaratacağı kod, 35KKP0307101 olacaktır. Bölge ve şube kodları sisteme giren şube temsilcisinin verileri kullanılarak otomatik olarak derlenecektir.

- Bağlı Olduğu Terminal: Araç şu an nerede bulunuyor. Ait olduğu yerde veya görevli olarak gittiği başka bir yerde olabilir. Bu bilgiler, sisteme araç tanıtıldıktan ve verileri girildikten sonra icra edilecek her taşımayı müteakip her bir araç için veri tabanından otomatik olarak güncellenecektir.

- Tahsis Eden Birim: Aracı, araç havuzuna dâhil etme yetkisi bulunan birimin verileri burada görüntülenecektir. Daha ast seviyede ve tahsis yetkisi olmayan birimin bu tahsisi yapması mümkün olmayacak ancak aracın tahsisi için yetkili birimden istek yapılabilecektir.

- Hazır Bulunma Durumu (Availability): Aracın görevde olması (G), Arızalı olması (A), Hazır olması (H). Görevde olması durumunda görevin bitiş ve aracın hazır duruma geliş zamanı otomatik olarak Kullanıcıya verilecektir. Şayet araç başka bir görev için önceden tahsis edilmiş ancak hali hazırda hazır bekler durumda ise tahsis edildiği görevin başlangıç zamanı da Kullanıcıya uyarı mahiyetinde verilecek ve Kullanıcının tahsis edeceği görev bu süreden önce bitmiyorsa söz konusu araç için tahsis yapılamayacak ve Kullanıcı uyarılacaktır.

Yazılım, Kullanıcı araç havuzuna araç eklemek istediğinde karşısına mevcut havuzdaki araçlar listelenecek ve ekle tuşuna tıklanınca açılacak listeden seçilecek her bir araç için taşıma programları görüntülenecektir. Bu programlarda “taşıma bacağı, araç cinsi, aracın hazır bulunduğu terminal, varış limanı, yük durumu, ayrılış zamanı, varış zamanı ve kat edeceği mesafe” görülebilecektir.

Yazılım, ulaştırma araç listesinden çoklu araç seçimini ve seçilen araçların programlarına ilişkin düzenlemelerin Kullanıcı tarafından yapılmasına imkân tanıyacaktır.

### **6.2.1. Taşıma Planına Ulaştırma Aracı Tahsisinde Filtreleme Yapabilme**

Yazılım liste içerisinde aşağıda belirtildiği şekilde araç filtrelemesi yapılmasına imkân tanıyacaktır.

Yazılım, Kullanıcının Taşıma Planı içerisindeki araçları “Ulaştırma Modu (Transport Mod), Kategorisi (Category), Araç Tip Kodu (Asset Type Code), Üretici ve Marka uzantıları (Maker and Mark Extensions), Araç Kodu (Asset Code), Bağlı Olduğu Terminal, Tahsis Eden Makam, Hazır bulunma durumu (Availability)” başlıkları altında filtrelemesine imkân tanıyacaktır.

Kullanıcı tarafından bu filtreleme özelliklerinin VE, VEYA mantık kapıları kullanılarak doğrulanmasına yönelik düzenleme yapılabilecektir.

Kullanıcı filtrelemeyi temizleyebilecektir.

Kullanıcı Taşıma Planı içinde yaptığı filtrelemeyi iptal edebilecektir.

### **6.2.2. Taşıma Planına Tahsis Edilecek Ulaştırma Aracının Seçimi**

Yazılım içerisinde aşağıda belirtildiği şekilde listeden araç seçilmesine imkân tanıyacaktır.

- Kullanıcı Taşıma Planında kullanılan araçları listeden seçebilecek, seçilen araç farklı renk ve yazı tipiyle yazılım tarafından belirginleştirilecektir.

- Yazılım, Kullanıcının Taşıma Planı içerisindeki araçları “Ulaştırma Modu(Transport Mod), Kategorisi (Category), Araç Tip Kodu (Asset Type Code), Üretici ve Marka uzantıları (Maker and Mark Extensions), Araç Kodu (Asset Code), Bağlı Olduğu Terminal, Tahsis Eden Makam, Hazır bulunma durumu (Availability)” başlıkları altında seçmesine imkân tanıyacaktır.

- Kullanıcı tarafından bu seçim özelliklerinin VE, VEYA mantık kapıları kullanılarak doğrulanmasına yönelik düzenleme yapılabilecektir.

- Kullanıcı bütün araçların gösterildiği seçim listesini temizleyebilecektir.

- Kullanıcı yaptığı seçimi iptal veya değiştirebilecektir.

Yazılım seçilen araç/araçların Kullanıcı tarafından plana eklenmesine imkân tanıyacaktır.

### **6.2.3. Taşıma Planı İçindeki Her Bir Taşıma/Taşıma Bacağı İçin Ulaştırma Aracı Tahsisi**

Kullanıcı, taşıma planı içerisinde yer alan bir taşımaya veya taşıma bacağına araç tahsisi yapabilecektir.

Kullanıcı, bahse konu taşımaya veya taşıma bacağına tahsis edilmek üzere seçilen araca ilişkin olarak;

- Araç Sırası Kontrolü,
- Geri Dönüş Sıra Kontrolü,
- Duraklama Noktası Kontrolü,
- Araç Tam Sefer (Gidiş-Dönüş) Kontrolü,
- Bütün Uygunluk Kontrollerinin Devre Dışı Bırakılması,
- Manifesto Uygunluk Kontrolü seçeneklerini işaretleyerek/işaretleri kaldırarak söz konusu ara kontrol ve işlemlerin yapılmasını/iptal edilmesini sağlayabilecektir.

Kullanıcı değerleri tekrar belirleyebilecek ve tekrar hesaplayabilecektir. Bu işlem kapsamında önceki tahsis gerçekleştirilmeyecek, tahsis sonuçları yazılım algoritması tarafından belirlenen duraklama noktalarını da kapsayacak şekilde temizlenecek ve yeni bir tahsis gerçekleştirilerek sonuçları görülebilecektir. Bu işlem tahsisin iptali ve taşı-bırak işleminin tekrarlanması şeklinde sistem tarafından uygulanacaktır.

Kullanıcı envanteri, güzergâhı ve araç programlarını tekrardan düzenleyebilecektir. Bu işlem duraklama noktalarının silinmesi haricinde “tekrar düzenle ve hesaplama” işlemiyle aynı görevi icra edecektir. Bu işlem Kullanıcının sistem tarafından belirlenen ulaştırma aracı ara duraklama noktalarını dikkate almayarak kendi belirlediğinde daha kullanışlı olacak ve sistem Kullanıcı tarafından belirlenen bu noktaları dikkate alarak tekrar tahsis yapacaktır.

#### **6.2.4. Plan İçindeki Her Bir Taşımaya/Taşıma Bacağına Yapılan Tahsis Detay Raporları**

Yazılım şu bilgileri içerecek şekilde yapılan tahsisin detaylarını gösterecektir:

- Ulaştırma Modu (Transport Mode),
- Kategorisi (Category),
- Araç Tip Kodu (Asset Type Code),
- Araç Kodu (Asset Code),
- Tahsis Eden Birim,

- Hazır bulunma durumu (Availability)
- Konuş yeri (Bağlı Olduğu Terminal/Birim),
- Gideceği yer,
- Tahsis edildiği birim,
- Tahsis edildiği bileşen ismi,
- Tahsis edildiği bileşen kodu.

Yazılım aşağıdaki raporları Kullanıcıya sunacaktır:

- **TAHSİS RAPORU** : Bu bölümde; "Tahsisi gerçekleştirmek için "ARAÇ ATAMA/EKLEME" düğmesine basınız", "Tahsis edilen araç müsaittir", "Tahsis başarıyla gerçekleştirildi. Yeni araç programları oluşturuldu, manifesto belirlendi ve zamanlama düzgün şekilde ayarlandı" veya "Tahsis edilen araç müsait değildir", "Yüklenmeye çalışılan kargo, aracın yükleme bölmesi ebatlarını aşmaktadır. Yüklenemeyen malzeme tahsis edilen araç için uygun değildir." ifadelerinden uygun olanı Kullanıcıya bildirilecektir.

- **YÜKLEME RAPORU**: Bu bölümde; "Yüklenmeye çalışılan kargo, aracın yükleme bölmesi ebatlarını aşmaktadır.", "Bir kısım kargo araca yüklenebilmektedir, ancak belirlenen zaman aralığında taşınması mümkün değildir" ifadelerinden uygun olanı Kullanıcıya bildirilecektir.

- **ZAMANLAMA RAPORU**: Bu bölüme ilişkin olarak yazılım ayrılış ve varış zamanlarını belirlenen zaman aralığında olduğu takdirde verecek, aksi takdirde her hangi bir bilgi vermeyecektir.

- **DURAKLAMA RAPORU**: Gerektiğinde yazılım "Duraklama Noktalarını" verecek veya bekleme/ikmal için uygun duraklama noktası bulunamadığına ilişkin bilgiyi Kullanıcıya sunacak ve Duraklama noktalarının Kullanıcı tarafından belirlenmesini sağlayabilecektir.

- **ARAÇ TAŞIMA YOĞUNLUK RAPORU**: Yazılım tahsis edilen aracın gerçekleştireceği taşımalara ilişkin bilgileri sunacaktır.

### **6.2.5. Taşınacak Birim Zamanlamasının Tahsisli Araç Programı ile Eş Zamanlanması**

Yazılım tahsis edilen araca uygun olarak, planlı ayrılış(en erken ve en geç) ve varış(en erken ve en geç) zamanlarını gösterecek şekilde hâlihazırda Planlanmış Bileşen Taşıma Zamanlamalarını yapacaktır.

Yazılım, her hangi bir bileşen taşımaya zaten tahsis edilmiş bir araç için planlı araç ayrılış(en erken ve en geç) ve araç varış(en erken ve en geç) zamanlarını gösterecektir.

Yazılım, "Tavsiye Edilen Araç Program Zamanlamaları"nı, planlı ayrılış(en erken ve en geç) ve varış(en erken ve en geç) zamanlarını gösterecektir.

Yazılım, bu tahsis için getirilen kısıtlar arasından aşağıdaki zaman bilgilerini Kullanıcıya sunacaktır.

- Kargo Mevcudiyet Ayrılış Zamanları.
- Yüklenmiş Araç En Erken Ayrılış Zamanı.
- Yüklenmemiş Taşınabilir Kargo En Erken Ayrılış Zamanı
- Ayrılış Terminali Bekleme Zamanı.
- Varış Terminalinde Bekleme Zamanı.
- Araç Varış Terminali En geç Varış Zamanı.

Kullanıcı, tavsiye edilen ayrılış zamanını zaman düzenleme fonksiyonuyla değiştirebilecektir.

Kullanıcı, son planlı varışı kullanarak bekleme zamanının senkronizasyonunu sağlayabilecektir.

Kullanıcı, ayrılışı bir sonraki yüke göre ayarlayabilecek ve tavsiye edilen ayrılış zamanını zaman düzenleme işleviyle değiştirebilecektir.

### **6.2.6. Ulaştırma Aracı Tahsisi Sonrası Yükleme Sonucunun Raporlanması**

Taşıma planına tahsis edilmiş ulaştırma aracına manifesto atanabilecektir. Bu kapsamda;

- Yazılım, envanteri Kullanıcıya "Miktar, REK, Tanım, Kargo Bölümü, Taşınacak birim Kodu, Bileşen Kodu, Kategori, Ağırlık, Uzunluk, Yükseklik, Genişlik" verilerini gösterecek şekilde listeleyecektir.

- Uygun durumdaki malzemelerin miktarına göre araçlara envanter ataması yapılacaktır.

- Yazılım, araçlar için bu maksatla Kargo Bölümleri için de “Kargo Bölümü Kodu, Adı, Yolcu, Ağırlık (Ton), LaneMetre, Metreküp, Metrekare, TEU(Twenty-foot Equipment Unit” veri alanlarını gösterecektir.

- Yazılım, kargoya ilişkin hem **Kullanım Ebatlarını** hem de **Taşıma Ebatlarını** gösterecek şekilde kargo ebatlarını Kullanıcıya sunacaktır.

- Yazılım, kargoya ilişkin hem **Dara Ağırlığını** hem de **Brüt Ağırlığı** gösterecek şekilde kargo ağırlığını Kullanıcıya sunacaktır.

- Kullanıcı uygun durumdaki malzemelerin miktarlarını değiştirebilecektir.

- Yazılım, seçilen bir malzemenin özelliklerini Kullanıcıya sunacaktır.

Manifesto ataması sonrasında yüklenemeyen envanter raporlanabilecek ve bu bağlamda yazılım, yapılan manifesto tahsisi sonrası yapılan yükleme neticesinde yüklenen ve yüklenemeyen envantere ilişkin bilgileri liste halinde Kullanıcıya sunacaktır.

- Yazılım, yüklenemeyen envanter için “Kod/REK, Müsait Olma Durumu (Evet/Hayır), Uyumluluk (Evet/Hayır), Tanım, Miktar, Ağırlık, Uzunluk, Yükseklik, Genişlik” veri alanlarını Kullanıcıya sunacaktır.

- Yazılım, yüklenen envanter için “Kod/REK, Tanım, Miktar, Ağırlık, Uzunluk, Yükseklik, Genişlik, Taşınabilme Kategorisi, Yüklendiği Kargo Bölmesi” veri alanlarını Kullanıcıya sunacaktır.

Manifesto ataması sonrası beklenen işlevler;

- Yazılım, toplam Yolcu, Ağırlık (Ton), Lane Metre, Metrekare ve Metreküpü Kullanıcıya sunacaktır.

- Kullanıcı, ara yüzde yer alacak tahsis edilmeyen malzeme bölümünden bir veya daha fazla malzemeyi alarak tahsis yapılmış envanter bölümüne taşıyabilecektir.

- Yazılım tahsis edilmiş kapasiteyi, Yolcu ve Kargo Kapasitelerinin toplam yüzdeleri şeklinde grafik olarak gösterecektir.



- Yazılım, Yolcu ve Kargo Kapasitelerinin yüzdelerini Araçtaki Mevcut Yük, Yeni Yük ve Toplam olarak grafik şeklinde gösterecektir.

- Yazılım, toplam yük hacmini gösterirken **Kullanım Ebatları** veya **Taşımaya Esas Ebatlar** olmak üzere iki farklı şekilde seçim yapılmasına imkân tanıyacaktır.

Yazılım, toplam yük ağırlığını gösterirken **Dara Ağırlığı** ve **Brüt Ağırlık** olmak üzere iki farklı şekilde seçim yapılmasına imkân tanıyacaktır.

#### **6.2.7. Araç Tahsisi Sonrası Taşıma Esnasında Üzerinden Geçilecek Lokasyonların Belirlenmesi**

“Taşıma Esnasında Üzerinden Geçilecek Lokasyonlar” ara yüzü, tahsis edilecek aracın değişik hareketlerine ilişkin duraklama noktaları konusundaki bilgileri de Kullanıcıya sunacaktır. Aynı zamanda, Kullanıcının istediği takdirde yeni duraklama noktaları belirlemesine imkân tanıyacaktır.

Mevcut iki araç programı arasına yeni bir araç programı sisteme girildiğinde taşıma için üç şekilde Üzerinden Geçilen Lokasyon belirlenebilecektir. Şöyleki;

Bekleme maksatlı kullanılacak Geçiş Lokasyonları; örneğin, kargo yüklenmesine İhtiyaç duyulan yükleme terminaleri.

Adrese teslim taşıma maksatlı kullanılacak Geçiş Lokasyonları; örneğin, Yükleme terminalinden indirme terminaline hareket

Dönüş maksatlı kullanılacak Geçiş Lokasyonları; örneğin, bir indirme terminalinden önceden belirlenmiş müteakip yükleme terminaline hareket.

Ara yüz bu üç taşıma şekli için duraklama noktalarının eklenmesine, yeniden düzenlenmesine veya silinmesine imkân tanıyacaktır.

#### **6.2.8. Taşıma Planına Ulaştırma Aracı Tahsisinde Rotalama**

Rota ve Araç Programları ara yüzü Kullanıcı “tahsis et” düğmesine basarak yapılan tahsisi onayladığında ulaştırma aracına ait bütün taşıma programları görüntülenecektir.

Belirginleştirilmiş satırlar yapılan son tahsisten etkilenen taşıma programlarını gösterecek ve yeni taşıma programları bir ikon ile gösterilecektir.

Yazılım, rota ve araç taşıma programları için “Konuş Yeri, Hazırlık Durumu, Ayrılış, Varış, Kargo Yüzdesi, Yolcu Yüzdesi ve Mesafe(Km)” veri alanlarını gösterecektir.

### **6.3. Ulaştırma Aracı Varış Doğrulaması**

Kullanıcı, seçilen araç taşıma programlarının “Planlandı, Kaydedildi ve Doğrulandı” şeklinde doğrulamasını yapabilecektir.

### **6.4. Ulaştırma Aracı Hazır Bulunma Durumu**

Yazılım, araç listesi üzerinde her hangi bir araç kodu üzerine çift tıklanarak araç özelliklerinin görüntülenmesini sağlayacaktır.

Yazılım, seçilen araç için ilk konuş lokasyonunun veri tabanında veya taşıma planı içinde bulunan mevcut lokasyon listesinden seçilmesine imkân tanıyacaktır.

Yazılım, seçilen aracın geri döneceği lokasyonun veri tabanında veya taşıma planı içinde bulunan mevcut lokasyon listesinden seçilmesine imkân tanıyacaktır.

Yazılım, seçilen aracın ilk olarak ne zaman tahsise uygun olacağını gün saat ve dakika olarak Kullanıcıya sunacaktır.

Kullanıcı, seçilen aracın ilk olarak ne zaman tahsise uygun olacağını gösteren veri alanı ile tahsisi yapan taşınacak birim hanesinde değişiklik yapabilecektir

### **6.5. Ulaştırma Aracı Program ve Manifestolarının Birleştirilmesi**

Yazılım, farklı iki veya daha fazla araçla veya aynı araçla iki veya daha fazla sortide/seferde taşınması planlanmış envanterin maliyet etkinliği göz önünde tutularak uygun durumdaki daha büyük bir araçla taşınabilmesi için karar desteği sağlayacaktır.

Yazılım, daha büyük bir araçla taşıma planlarken önceden planlı araçların sortilerine/seferlerine ait taşıma programlarını dikkate alacak ve zamansal açıdan en uygun planlamayı Kullanıcıya sunacaktır.

Yazılım, birleştirme faaliyeti için araç tahsisine ilişkin bütün esasların uygulanmasını sağlayacaktır.

### **6.6. Spesifik Araç Tercihleri**

Kullanıcı, bir veya daha fazla araç tipini ekleyebilecek/düzenleyebilecektir. Bu durumda sadece aynı taşıma moduna ait araçlar seçilebilecektir.

Kullanıcı, her bir araç tipi için öncelik derecesini ve maksimum sorti/sefer miktarını sisteme tanımlayacaktır. Tanımlanmadığında maksimum sorti miktarı sistem tarafından "99" girilmiş olarak kabul edilecektir.

Yazılım, Kullanıcıya oluşturulan araç tipleri listesinde "Öncelik Derecesi, Araç Tip Kodu, Tanım, Tahsisli Sorti/Sefer Miktarı, Kullanılan Sorti/Sefer Miktarı" veri alanlarını görebilecektir.

- Taşımaya Uygun Araç Programları: Yazılım, taşıma programında taşımaların başlangıç ve bitiş yerlerini ve ulaştırma modunu gösterecektir. Aynı zamanda taşımaya uygun durumdaki araç programları listesinde "Kodu, Adı, Araç Tip Kodu, Rotası, Ayrılış Yeri, Varış Yeri, Kargo Yüzdesi, Yolcu Yüzdesi" veri alanlarını gösterecektir.

- Taşıma Envanteri: Kullanıcı taşıma envanteri listesinde "Kod/REK, Tanım, Miktar, Araç Kodu, Ayrılış Yeri, Varış yeri, Kategorisi, Ağırlığı, Uzunluğu, Yüksekliği, Genişliği" alanlarını görebilecektir. Kullanıcı, seçilen envanter malzemesi ve Kullanım/Taşıma Ebatlarından ve/veya Dara Ağırlığı/Brüt Ağırlık seçeneklerinden birinin seçimine bağlı olarak toplam envanteri "Yolcu, Malzeme, Ton, LaneMetre, Metrekare, Metreküp, TEU" alanlarını görüntüleyebilecektir.

- Taşıma Bilgisi: Kullanıcı taşıma bilgileri dahilinde, tahsis edilen taşınacak birim ve bileşen adı ve kodu, ulaştırma modu, TÇN, TBN, planlı ayrılış tarih ve saati, araç tahsis edilmiş taşıma çıkış tarih ve zamanları, araç tahsis edilmiş taşıma varış tarih ve zamanları ve mesafe bilgi alanlarını görebilecektir. Kullanıcı, araç tercihlerini;

- Bünyesindekileri/Kontrolüne Verilenleri Kullan,
- Tercih Edilenleri Kullan,
- Araç Hizmet Alımı (Kiralama)
- Hepsini kullan seçeneklerinden her hangi birini veya bir kaçını seçerek belirleyebilecektir.

Kullanıcı, içinden transit olarak geçilen lokasyonları bir liste halinde görebilecek, ekleme, silme ve değiştirme yapabilecektir.

## **6.7. Ulaştırma Aracı Ayrılış Doğrulaması**

Kullanıcı, seçilen araç taşıma programları ayrılış durumları için "Planlandı, Kaydedildi ve Doğrulandı" şeklinde doğrulama yapabilecektir.

## 6.8. Ulaştırma Aracı Taşıma Mesafelerinin Düzenlenmesi

Kullanıcı mevcut ulaştırma ağı üzerinde mesafeleri otomatik olarak güncelleyebilecektir. Mesafelerin hesaplanmasında hangi birimin (km, deniz mili ve kara mili) kullanılacağına Kullanıcı tarafından karar verilebilecektir.

## 6.9. Ulaştırma Aracı Manifestosu

Kullanıcı araç manifestosuna yönelik olarak;

- Taşıma Planı araç havuzu içerisinde araç tipi ve ulaştırma modu şeklinde listelenmiş ulaştırma araçları içerisinde her hangi bir ulaştırma aracını seçebilecektir.

- Uygulama, aracın cinsini gösterecektir.
- Uygulama, aracın icra edeceği planlı taşımaları gösterecektir.
- Uygulama, her bir taşıma için taşıma çıkış yeri, varış yeri, hazır olma durumu, çıkış ve varış zamanlarını gösterecektir. Kullanıcı, araç taşıma bacaklarını "Araç Taşıma Bacağı Düzenleme" fonksiyonunu kullanarak düzenleyebilecektir.

- Kullanıcı her hangi bir taşımayı seçebilecektir.

- Uygulama, yüklenmiş bileşenler için; taşınacak birim ve bileşen kodu, çıkış yeri, varış yeri bilgilerini sunacaktır.

- Kullanıcı, yüklenmiş bileşenlerin ait olduğu taşınacak birime ilişkin verilere ulaşabilecektir.

- Kullanıcı, seçilen bileşeni taşıyan aracın tahsisini kaldırabilmeli veya söz konusu aracı boşaltabilecektir.

- Kullanıcı, yüklü bileşenlere ilişkin "Bileşen Manifesto Atama" verilerine ulaşabilecektir.

- Uygulama, her hangi bir araç manifestosu içerisinde yer almayan envanteri gösterecektir.

- Uygulama, her hangi bir araç manifestosu içerisinde yer almayan envanter malzemelerine ilişkin olarak, miktar, kategori, ağırlık, uzunluk, genişlik ve yükseklik bilgilerini görüntüleyecektir.

- Uygulama, her hangi bir araç manifestosu içerisinde yer almayan envanter malzemelerine ilişkin olarak toplam yolcu, kargo (ton), LaneMetre, metrekare ve metreküp miktarlarını gösterecektir.
- Uygulama, seçilen her hangi bir araca tahsis edilmiş envanter bilgilerini gösterecektir.
- Uygulama, tahsis edilmiş envanter için Kod/REK, tanım, kargo bölümü kodu, miktar, kategori, ağırlık, uzunluk, genişlik ve yükseklik bilgilerini görüntüleyecektir.
- Uygulama, manifesto içerisinde yer alan toplam yolcu, kargo (ton), LaneMetre, metrekare, metreküp ve TEU miktarlarını gösterecektir.
- Kullanıcı, seçilen miktardaki envanteri, “Hiçbir Manifestoda Olmayan Envanter” ve “Atanmış Envanter” grupları içerisinde ileri ve geri şeklinde hareket ettirebilecektir. Her hangi bir malzemeyi/envanteri tahsis ederken/atamasını yaparken Kullanıcı hedef bir kargo bölümü seçebilecektir.
- Kullanıcı, her hangi bir manifestodaki ağırlık ve hacminin belirlenebilmesi için; kargonun kullanım boyutları ve taşımaya esas boyutlar ile boş ve brüt ağırlık kullanılmasına ilişkin seçeneklerden istediğini seçebilecektir.
- Her hangi bir malzeme listesinden seçilecek bir malzeme için Kullanıcı malzeme özelliklerini görüntüleyebilecektir.
- Kullanıcı istediğinde seçilen her hangi bir aracın bilgilerine ulaşabilecektir.
- Yazılım manifestonun bütününün elleçlenmesine yönelik bir algoritma kullanacaktır.
- Ulaştırma aracı manifestosu özelliklerinin görüntülenmesi konusunda aşağıdaki hususlar dikkate alınacaktır:
  - Uygulama, seçilen her hangi bir ulaştırma aracının toplam manifestosunu gösterecektir.
  - Uygulama manifestodaki her bir malzeme için Kod/REK, tanım, kargo bölümü kodu, miktar, kategori, ağırlık, uzunluk, genişlik ve yükseklik bilgilerini görüntüleyecektir.

- Uygulama, manifesto içerisinde yer alan yolcu, kargo miktarı, ağırlık (ton), LaneMetre, metrekare ve metreküp miktarlarını seçilen malzemeler için gösterecektir.
- Uygulama, her bir kargo bölümü için manifesto bilgisini gösterecektir.
- Uygulama, toplam kapasite ve her bir kargo bölümü kapasitesine ilişkin yolcu, ağırlık (ton), LaneMetre, metrekare, metreküp ve TEU” miktar bilgilerini gösterecektir.
- Uygulama, her bir kargo bölümü için tahsis edilen kargo kapasitesini toplam kapasitenin yüzdesi şeklinde yolcu ve kargo olmak üzere ayrı ayrı grafik barlar şeklinde gösterecektir.
- Uygulama, kargo ağırlık ve hacminin belirlenebilmesi için; kargonun kullanım boyutları ve taşımaya esas boyutları ile boş ve brüt ağırlık kullanılmasına ilişkin seçeneklerden istediğini seçebilecektir.

#### **6.10. Çoklu Taşıma İçin Ulaştırma Aracı Tercih Yapma**

Kullanıcı, birden fazla taşıma için araç seçebilecek ve araç tercihlerini düzenleyebilecektir.(KURAL-9)

Kullanıcı, taşıma planlarına araç atayabilecek/silebilecektir.

Kullanıcı, öncelik, araç tipine bağlı araç kodu ve maksimum sorti/sefer özelliklerini belirleyebilecektir.

Uygulama, aracın tanımını ve özelliklerini gösterecektir.

#### **6.11. Ulaştırma Aracı Güzergâh Planı**

Uygulama, araç havuzuna dâhil edilmiş bir aracın araç programını taşıma çıkış noktası (TÇN), taşıma bitiş noktası (TBN), hazır olma durumu, çıkış ve varış zamanları, yolcu yüzdesi, kargo yüzdesi, mesafe, araç kodu ve adı veri alanlarını gösterecektir.

Uygulama, araç programlarını seçebilecek veya seçimi iptal edebilecektir.

Uygulama, araç taşıma bacaklarını “Araç Taşıma Bacağı Düzenleme” fonksiyonunu kullanarak düzenleyebilecektir.

Uygulama, Araç Manifesto düzenleme fonksiyonunu kullanabilecektir.

Uygulama, “Araç Programlarından Malzeme ve Kargoyu Boşalt” fonksiyonunu kullanabilecektir.

Uygulama, “ Seçilen Araçların Taşıma Güzergâhlarını Yeniden Düzenle” fonksiyonunu kullanabilecektir.

Uygulama, kaydet fonksiyonunu kullanabilecektir.

Uygulama, “Araç Zamanlamasını Girme” fonksiyonunu kullanabilecektir.

Uygulama, “Araç Zamanlamasını Öteleme” fonksiyonunu kullanabilecektir.

Uygulama, “Araç Taşımalarını Otomatik Planlama” fonksiyonunu kullanabilecektir.

Uygulama “Öncesine Lokasyon Ekle/Gir” fonksiyonunu kullanabilecektir.

Uygulama, “Sonrasına Lokasyon Ekle/Gir” fonksiyonunu kullanabilecektir.

Uygulama, “Araç Taşıma(lar)ını Sil” fonksiyonunu kullanabilecektir.

Uygulama, “Araç Taşımalarını Tekrarla” fonksiyonlarını kullanabilecektir.

Uygulama, “Görüntüle” fonksiyonunu kullanabilecektir.

#### **6.12. Araç Zamanlamasını Girme**

Kullanıcı, araçların sefer bacakları bazında zamanlamasını gün saat ve dakika olarak yapabilecektir. Bunu yaparken de hazırlık zamanı, en erken ayrılış/en geç ayrılış, en erken varış/en geç varış zamanlarını ayarlayabilecektir.

#### **6.13. Araç Zamanlamasını Öteleme**

Kullanıcı hazırlık, ayrılış ve varış zamanlarını ayrı ayrı veya toptan belli bir tarihe saate öteleyebilecek/geri alabilecektir. Bunu önce/sonra şeklinde gün saat ve dakika olarak yapabilecektir.

#### **6.14. Araç Taşımalarının Otomatik Planlanması**

Kullanıcı, aracın hareketinin hazırlık, ayrılış veya varış zamanlarından birinden itibaren başlatılmasını sağlayabilecektir.

Kullanıcı, zamanı hazırlık zamanını müteakiben ileri doğru veya son noktaya ulaşması gerekli en son zamandan (CRD) geriye doğru planlayabilecektir.

Kullanıcı, mümkün olan boş zamanı kullanacak şekilde sistemin otomatik planlama yapmasını sağlayabilecektir.

### **6.15. Öncesine Lokasyon Ekleme (Yeni Bir Taşıma Bacağı Oluşturma)**

Kullanıcı, seçilen bir araç taşıma bacağının öncesine yeni bir lokasyon ekleyebilecektir.

Kullanıcı, taşıma planı içerisinde kullanılan lokasyonlar veya veri tabanı içerisinde yer alan tüm lokasyonlardan tarayarak lokasyon seçebilecektir.

Kullanıcı, lokasyonları ülke, tüm altyapı çeşitleri, hava alanları, deniz limanları, tren istasyon/garları, depolama bölgeleri, sınır geçiş bölgeleri ve diğer seçeneklerinden birini seçerek filtreleyebilecektir.

### **6.16. Sonrasına Lokasyon Ekleme (Yeni Taşıma Bacağı Oluşturma)**

Kullanıcı, seçilen bir araç taşıma bacağının sonrasına yeni bir lokasyon ekleyebilecektir.

Kullanıcı, taşıma planı içerisinde kullanılan lokasyonlar veya veri tabanı içerisinde yer alan tüm lokasyonlardan tarayarak lokasyon seçebilecektir.

Kullanıcı, lokasyonları ülke, tüm çeşitler, hava alanları, deniz limanları, tren istasyon/garları, depolama bölgeleri, sınır geçiş bölgeleri ve diğer seçeneklerinden birini seçerek filtreleyebilecektir.

### **6.17. Araç Taşımalarını Silme**

Kullanıcı, araç taşımalarını silebilecektir.

### **6.18. Araç Programlarının Tekrarlanması**

Kullanıcı, zamanı ayarla fonksiyonunu kullanarak tekrarlanan programlar arasında gün, saat ve dakika olarak zaman ayarlaması yapabilecektir.

Varsayılan değer 1 olmak üzere, Kullanıcı seçilen programların kaç defa tekrarlanacağını belirleyebilecektir.

Kullanıcı, tekrarlanacak araç programlarını, programın en sonundaki taşıma bacağını müteakip veya seçili olan taşıma bacağından sonra tekrarlanacak şekilde düzenleyebilecektir.

### **6.19. Araç Güzergâh Raporu Görüntüleme**

Uygulama, Araç Güzergâh Planlama Raporunu görüntüleyebilecektir.

Uygulama, Kullanıcının söz konusu rapor için gizlilik derecesini girmesine imkân tanıyacaktır.



Uygulama, söz konusu raporda Kullanıcının gerektiğinde kendi yorum ve değerlendirmelerini yazmasını sağlamak üzere bir açıklama/değerlendirme bölümü sağlayacaktır.

#### **6.20. Ulaştırma Aracı Tercihleri**

Uygulama, taşınacak birim adı, bileşen adı, taşıma modu, taşıma adı, TÇN, TBN bilgilerini ve araç tipleri listesini tercihlerin yapılması için gösterecektir.

Araç Tipleri Listesinde listelenen her bir araç tipi için minimum olarak; öncelik, araç tip kodu, tanımı, maksimum sorti/sefer sayısı ve kullanılan sorti/sefer sayısı alanları gösterilecektir.

Kullanıcı söz konusu listeye araç ekleyip çıkarabilecektir.

#### **6.21. Ulaştırma Aracı Özellikleri**

Uygulama, araç çeşidi, ismi, seri numarası, araç kodu, araç tip kodu, bağlı olduğu terminal, terminalin bulunduğu yer alanlarını içeren araç bilgilerini Kullanıcıya sunacaktır.

Araç özelliklerinde, genel bilgiler sekmesi bulunacaktır.

Araç özelliklerinde, araç çeşidi (uçak, gemi, kara yolu aracı, grup araç, demir yolu aracı ve tren) bilgileri yer alacaktır.

Uygulama, yolcu ve kargo bölümlerini gösterecektir.

Uygulama, elleçleme kapasitesini gösterecektir.

Uygulama, Taşıma Planı spesifik bilgilerini sunacaktır.

Uygulama, yük cinsi bölümünü Kullanıcıya sunacaktır.

Uygulama, şu genel bilgileri içerecektir:

- Araç Tip Kodu, araç çeşidi, hızı (Knot, km/sa, mi/sa), tanımı, birim maliyeti, yükleme ve boşaltma zamanları (gün, saat, dakika), kargo bölümü adedi, elleçleme kapasitesi, güncelleme bilgileri.

- Uygulama, kargo bölümü listesi sunacaktır. Her bir kargo bölümü için şu veriler sunulacaktır: Kargo bölümü kodu, kapasitesi (ton), yolcu kapasitesi, LaneMetre Kapasitesi, metrekare kapasitesi, metreküp kapasitesi.

- Uygulama, elleçleme kapasitesi listesi sunacaktır. Her bir elleçleme tipi şu bilgileri sunacaktır: Araç tipi, elleçleme işlem kodu, elleçleme hızı, maksimum

kargo uzunluk/genişlik/yükseklik/ağırlığı, bekleme süresi, açıklamalar, başlama ve durma zamanları.

- Uygulama, elleçleme aracı listesini sunacaktır. Her bir elleçleme araç listesinde şu bilgiler yer alacaktır: Araç tipi, elleçleme ekipmanı tip kodu ve miktarı.

#### **6.21.1. Ulaştırma Araç Çeşidi Özellikleri**

Uygulama, her bir araç çeşidine spesifik özellikleri sunacaktır.

Araç çeşitleri olarak uçak, gemi, kara yolu aracı ve çoklu kara yolu araç grupları (convoy), demir yolu aracı ve katar bilgileri yer alacaktır.

Uygulama, gemi tipi için; araç tip kodu, çeşidi, tanımı, gemi kategorisi (kuru yük, yolcu, tanker), gerçek mi sanal mı, yükseklik ABW, uzunluk, bum (dip) uzunluğu, draft, yakıt tipi, tüketim miktarı (Ton/Gün), soğuk iklim kullanım durumu (evet/hayır/bilinmiyor), boş ağırlık, gemi tipi (genel kargo, hastane gemisi, likit doğalgaz gemisi, likit petrol gemisi, Ro-Ro gemisi, konteyner gemisi, feribot, maden filizi gibi), sınıfı (denizde giden, sadece sahil boyunca kullanılan ve iç su yolu) veri alanlarını sunacaktır.

Uygulama, uçak tipi için; araç tip kodu, çeşidi, tanımı, açıklamalar, maksimum yükle hızı (km, kara mili, deniz mili), günlük kullanım süresi (saat ve dakika olarak), maksimum yük kapasitesi (ton), kategorisi, havadan havaya yakıt ikmal yeteneği (var/yok/bilinmiyor) bilgilerini sunacaktır

Uygulama demir yolu aracı ve katar için; araç tip kodu, çeşidi, tanımı, açıklamalar, raylar arası açıklık (Gauge)(m), darası (ton), vagon kategorisi, yükleme tabanı yüksekliği (HLBT)(m) bilgilerini sunacaktır

Uygulama kara yolu aracı ve grup tipi için; araç tip kodu, çeşidi, tanımı, açıklamalar, hız (km), boş ağırlığı, yükleme tabanı yüksekliği (m), taşıma tipi, taşıma fonksiyon kodu ve yük sınıfı bilgilerini sunacaktır

Uygulama, her bir kargo bölümü için; kodu, önceliği, adı, depolama tipi, açıklamalar, kargo bölümü ebatları, maksimum kargo ebatları, maksimum taban dayanımı (kg/m<sup>2</sup>), maksimum istif ağırlığı (kg), yolcu, ton, lane metre, metrekare, metreküp ve 20 Ayaklık Malzeme Birimi (TEU) bilgilerini sunacaktır.

Uygulama, yükleme ve boşaltma için Araç tip kodu, elleçleme türü, açıklamalar, hız, konaklama süresi, başlama zamanı (sa, dk.), bitiş zamanı (sa, dk.), elleçlenecek maksimum kargo ebatları gibi elleçleme kapasite verilerini sunacaktır.

### **6.21.2. Ulaştırma Aracı Taşıma Planına Özel Bilgiler**

Uygulama, ulaştırma araçları için şu taşıma planı spesifik bilgilerini sunacaktır:

- Çıkış Lokasyonu,
- Dönüş Lokasyonu,
- İlk Hazır Olma Zamanı
- Son Hazır Olma Zamanı

Araç taşıma listesi Kullanıcıya sunulmalı ve TÇN, TBN, hazırlık, ayrılış ve varış zamanları bilgilerini gösterecektir.

### **6.21.3. Ulaştırma Aracı Kuru Yük Özellikleri**

Uygulama, araç tip kodu, çeşidi, tanımı, açıklamalar, kapalı bölme sayısı, hangar kapak sayısı, rampa tipi, maksimum hangar kapağı uzunluğu/genişliği, tehlikeli madde güvertesi bilgilerini sunacaktır.

### **6.21.4. Ulaştırma Araç Tipi Özellikleri**

Uygulama araç tiplerine ilişkin olarak genel özellikleri Kullanıcıya sunacaktır.

Uygulama, araç çeşidi özelliklerini (Uçak, Gemi, Kara yolu Aracı ve Grup araçlar, demir yolu aracı ve katarlar) Kullanıcıya sunacaktır.

Uygulama, yolcu ve kargo bölümlerine ilişkin özellikleri Kullanıcıya sunacaktır.

Yazılım elleçleme kapasite bilgilerini sunacaktır

### **6.21.5. Ulaştırma Aracı Taşıma Güzergâhının Yeniden Belirlenmesi**

Kullanıcı araçların taşıma güzergâhlarını yeniden belirleyebilmeli, değiştirebilmeli ve istediği düzenlemeleri yapabilecektir.

### **6.21.6. Ulaştırma Aracı Kargo Boşaltma**

Kullanıcı araçlardan kendilerine tahsis edilen kargoyu tek işlemle boşaltabilecektir.

### **6.21.7. Ulaştırma Aracından Bileşen ve Kargoyu Boşaltma**

Kullanıcı, araçlardan taşınacak birimleri ve bileşenlerini boşaltabilecek ve araç tahsislerini kaldırabilecektir.

## **7. Taşınacak Birim Bileşenlerine İlişkin Özellikler**

### **7.1. Bileşenlere Ayırma**

Kullanıcı seçilen birliği bileşenlerine ayırabilecektir.

Uygulama, REK, tanım, miktar, kategori, ağırlık, uzunluk, genişlik ve yükseklik veri alanlarını gösterecek şekilde ana bileşendeki (taşınacak birimin kendisi) bütün malzemeleri listeleyebilecektir. Uygulama, toplam yolcu, malzeme, ağırlık, LaneMetre, metrekare, metreküp veri alanlarını bu malzemeler için gösterecektir.

Uygulama, bileşenlerine ayrılmamış taşınacak birimleri "Ana Bileşen" olarak gösterecektir.

Kullanıcı istediği zaman ilave bileşen ekleyebilecektir.

Her bir bileşen için uygulama, REK, tanım, miktar, kategori, ağırlık, uzunluk, genişlik ve yükseklik bilgilerini listeleyebilecektir. Uygulama, toplam yolcu, malzeme, ağırlık, LaneMetre, metrekare, metreküp veri alanlarını bu malzemeler için gösterecektir.

Kullanıcı, bileşenlere malzeme atayabilecektir. Kullanıcı, seçilen malzemeleri/tüm malzemeleri istendiğinde miktarlarını belirleyerek ileri ve geri olacak şekilde ana bileşen ile alt bileşen arasında bir seferde hareket ettirebilecektir (bileşen malzeme atayıp, bu atamayı iptal edebilecektir).

Kullanıcı önceden oluşturulmuş bir bileşeni silebilecektir.

Uygulama, kullanım/taşımaya esas boyutlar ile darasız/brüt ağırlık verilerini sunacaktır.

Kullanıcı, oluşturulacak her bir yeni bileşen için tercih edilecek ulaştırma aracını araç çeşidi ve tipini seçerek belirleyebilecektir.

Kullanıcı, sadece tercih edilen araçla taşınabilecek malzemeleri atayabilecektir.

Envanteri ana bileşenden alıp alt bileşene tahsis ederken Kullanıcı, envanter miktarlarını yolcu, ton, m, M2, M3 şeklinde belirleyebilecektir.

Uygulama, araç tip bilgilerini bu aşamada Kullanıcıya sunacaktır.)

## **7.2. Bileşenlere Otomatik Ayırma**

Kullanıcı, taşınacak birimleri otomatik olarak üç taşıma bileşenine (Yolcu, Malzeme, Araç) ayırabilecektir.

## **7.3. Bileşen Envanteri**

Uygulama, bir bileşene ait tüm malzemeleri gösterecektir.

Uygulama, envanterdeki malzemeler için REK, tanım, miktar, bileşen kodu, kategori, ağırlık, uzunluk, genişlik ve yükseklik bilgilerini görüntüleyecektir.

Kullanıcı bir veya daha fazla malzemeyi aynı anda seçebilecektir.

Uygulama, seçilmiş ve tüm malzemelerin toplamlarını yolcu, malzeme, ton, LaneMetre, metrekare, metreküp şeklinde verecektir.

Uygulama, taşınabilme kategorisine göre gruplandırılmış yolcu, tekerlekli araç, konteyner, diğer, ton, LaneMetre, metrekare, metreküp bilgilerini ekranda raporlayabilecektir.

Uygulama, aynı zamanda bu miktarların toplamlarını da verecektir.

Kullanıcı, kendi kendine taşıma eden araçları belirleyebilecektir.

Uygulama, malzemelerin ebatları için kullanım ve taşımaya esas değerlerini sunacaktır.

Uygulama, malzemeler için darasız ağırlık ve brüt ağırlık değerlerini verecektir.

Uygulama, Milli kod, adı, birliği, miktarı, brüt ağırlığı, darasız ağırlığı, taşımaya esas boyutları (cm), kullanım boyutları (cm), genel toplam ve taşınabilme kategorilerine göre toplamları gösteren bilgi alanlarından oluşacak "Taşınacak birim Envanter Raporunu" gösterecektir.

Rapordaki tüm bilgiler taşınabilme kategorilerine dayandırılacaktır.

## **7.4. Bileşen Manifestosu**

Kullanıcı, bir taşınacak birim manifestosunu görüntüleyebilecek ve değişiklik yapabilecektir.

Kullanıcı, bir taşınacak birimi ve envanter bilgileri için taşınacak birimin bileşenini Taşınacak Birim Kodunu kullanarak taşıma planı içerisinden seçebilecektir.

Uygulama, taşınacak birim bileşeni için de taşıma listesi sunacaktır. Uygulama, TÇN, TBN, ulaştırma modu, ilk ayrılış ve en son varış zamanları bilgilerini her bir taşıma için sunacaktır.

Kullanıcı, araç taşıma bacaklarını “Araç Taşıma Bacak Düzenleme” fonksiyonunu kullanarak düzenleyebilecektir.

Kullanıcı her hangi bir taşımayı seçebilecektir.

Uygulama, seçilen bir taşımaya tahsis edilen araçlar için; araç, tip, rota, ayrılış, varış, kargo yüzdesi ve yolcu yüzdesi veri alanlarını sunacaktır.

Kullanıcı, tahsis edilen araçlar için araç bilgileri ara yüzüne ulaşabilecektir.

Kullanıcı, seçilen araçlardan yüklenmiş bileşenlerin boşaltılmasını ve araç tahsisinin kaldırılmasını sağlayabilecektir.

Kullanıcı, tahsis edilen araçlar için araç manifesto verileri ara yüzüne ulaşabilecektir.

Uygulama, her hangi bir manifestoda yer almayan envanter malzeme listesini sunacaktır. Uygulama, her hangi bir manifestoda yer almayan envanter için REK, tanım, miktar, kategori, ağırlık, uzunluk, genişlik ve yükseklik bilgilerini görüntüleyecektir.

Uygulama, her hangi bir manifestoda yer almayan envanter için yolcu, kargo (ton), LaneMetre, metrekare, metreküp toplamlarını sunacaktır.

Uygulama, tahsis edilmiş araçlar arasından seçilmiş araçlara tahsis edilmiş envanteri Kullanıcıya sunacaktır. Uygulama, tahsis edilmiş araçlar arasından seçilmiş araçlara tahsis edilmiş envanter için REK, tanım, kargo bölümü kodu, miktar, kategori, ağırlık, uzunluk, genişlik ve yükseklik bilgilerini görüntüleyecektir.

Uygulama, seçilen araçlara tahsis edilmiş envanter için yolcu, kargo (ton), LaneMetre, metrekare, metreküp toplamlarını sunacaktır.

Kullanıcı, Kullanıcı, seçilen miktardaki envanteri, “Hiçbir Manifestoda Olmayan Envanter” ve “Atanmış Envanter” grupları içerisinde ileri ve geri şeklinde hareket ettirebilecektir. Malzemeyi tahsis ederken Kullanıcı hedef kargo bölümünü de seçebilecektir.

Kullanıcı, kullanım/taşımaya esas boyutlar ve darasız/brüt ağırlık değerlerinden istediğini kargonun hacim ve ağırlığını belirlemek maksadıyla seçebilecektir.

Her hangi bir Envanter Listesi için, seçilen bir malzemenin malzeme özellikleri görüntülenmek/değiştirilmek üzere seçilebilecektir.

Kullanıcı, taşınacak birim bilgilerine görüntülemek/değiştirmek üzere ulaşabilecektir.

Kullanıcı, REK Kodu, tanım, araç kodu, miktar, kategori, ayrılış, varış, ağırlık, uzunluk, genişlik ve yükseklik bilgilerini görüntüleyebilecektir.

### **7.5. Kendi Kendini Taşıyan Bileşen**

Kullanıcı, kendi kendini taşıyabilen bileşenleri (araç tahsisi yapmadan kendi kendine gidebilecek nitelikteki tekerlekli araç vb. tipi malzemelerden oluşan bileşen) belirleyebilecektir.

### **8. İkon Tanım ve Özellikleri**

Uygulama, yazılım içerisinde kullanılan ikonların (kısayol göstege tuşları) tanımlarını sunacaktır.

### **9. Taşınacak Birimlere İlişkin Özellikler**

#### **9.1. Taşınacak Birim İhtiyaç Listesinden/Taşınacak Birim Konuşlanma Listesinden Taşınacak birim Ekleme**

Kullanıcı, taşınacak birim ihtiyaç listesinden veya taşınacak birim konuşlanma listesinden taşıma planına taşınacak birim ekleyebilecektir.

Uygulama, TBYDL'deki tüm taşınacak birimleri taşıma planındaki taşınacak birim listesine eklemek üzere görüntüleyebilecektir.

Kullanıcı sunulan listedeki taşınacak birimlerden taşıma planına taşınacak birim seçerek ekleyebilmeli ve taşıma planından taşınacak birimleri silebilecektir.

Uygulama, taşıma planı ismini seçilmek üzere Kullanıcıya sunmalı ve seçilecek plana ilişkin listeleri ekrana getirecektir.

Uygulama, Plan Adı, Plan Tipi, Hazırlanma Tarihi, Gizlilik Derecesi, Açıklamalar gibi plan detaylarını ekrana getirecektir. Kullanıcı, açıklamalar alanını güncelleyebilecektir.

## 9.2. Taşıma Profilinden Taşınacak Birim Ekleme

Kullanıcı TP'na taşınacak birim profilinden seçilecek taşınacak birimleri ekleyebilecektir. Kullanıcı önce bir taşınacak birim profili seçebilmeli ve profile yer alan taşınacak birimlerin listesini görüntüleyebilecektir. Kullanıcı aynı zamanda seçilmiş taşıma planındaki tüm taşınacak birimlerin listesini ve seçilmiş taşıma planını da görebilecektir.

Uygulama, Plan Adı, Plan Tipi, Hazırlanma Tarihi, Gizlilik Derecesi, Açıklamalar gibi plan detaylarını ekrana getirecektir. Kullanıcı, açıklamalar alanını güncelleyebilecektir.

Kullanıcı, taşınacak birimlerden taşıma planına taşınacak birim seçerek ekleyebilmeli ve taşıma planından taşınacak birimleri silebilecektir.

Kullanıcı, TP'ye bir taşınacak birim eklediğinde taşınacak birim kodu, taşınacak birim adı, barış konuş yeri bilgilerini görmeli ve Taşıma İhtiyaç Listesi verilerini TBN, CRD, en erken ayrılış ve öncelik verilerini belirleyebilecektir.

Kullanıcı, belirlediği Taşıma İhtiyaç Listesi verilerini söz konusu taşınacak birimin tüm alt birimlerine de uygulayabilme seçeneğini seçebilecektir.

Kullanıcı taşınacak birim eklediğinde TP'nin açıklamalar bölümünü güncelleyebilecektir.

## 9.3. Taşınacak Birim Tahsis Bilgileri Güncelleme

Kullanıcı, bir bileşene ait bir veya daha fazla taşımayı seçebilmeli ve bu taşımaların tahsis durumlarını güncelleyebilecektir.

## 9.4. Taşınacak Birim Varış Doğrulaması

Kullanıcı, seçili taşınacak birimlerin varış durumları için "Planlandı, Kaydedildi ve Doğrulandı" şeklinde doğrulama yapabilecektir.

## 9.5. Taşınacak Birimleri Başka Planlara Kopyalama

Kullanıcı taşınacak birimleri başka TP'ye/plana kopyalayabilecektir.

Uygulama, mevcut planları ve TP'leri Kullanıcıya sunacaktır.

Kullanıcı, TPM'de açık olan bir TP'nin taşınacak birim listesinden seçilmiş taşınacak birim/taşınacak birimleri, yeni bir hedef plan ve TP'nı seçerek kopyalayabilecektir.



Uygulama, Plan Adı, Plan Tipi, Hazırlanma Tarihi, Gizlilik Derecesi, Açıklamalar gibi plan detaylarını ekrana getirecektir. Kullanıcı, açıklamalar alanını güncelleyebilecektir.

#### **9.6. Taşınacak Birim Silme**

Kullanıcı taşıma planından taşınacak birimleri çıkarabilecektir.

#### **9.7. Taşınacak Birim Ayrılışları Doğrulaması**

Kullanıcı, seçili taşınacak birimlerin ayrılış durumları için "Planlandı, Kaydedildi ve Doğrulandı" şeklinde doğrulama yapabilecektir.

#### **9.8. Taşınacak Birim Taşınma Mesafelerinin Düzenlenmesi**

Kullanıcı, kullanılacak güzergâhta değişiklik yaptığında uygulama otomatik olarak mesafeleri güncelleyecektir.

#### **9.9. Taşınacak Birim Envanteri**

Uygulama, seçilen taşınacak birimin envanteri raporlayacak bir fonksiyon sunacaktır.

Uygulama, bir birliğe ait tüm malzemeleri görüntüleyecektir.

Uygulama, listedeki her bir malzeme için REK, tanım, miktar, taşınacak birim kodu, bileşen kodu, kategori, ağırlık, uzunluk, genişlik ve yükseklik veri alanlarını sunacaktır.

Kullanıcı bir veya daha fazla malzemeyi seçebilecektir.

Uygulama, yolcu sayısı, malzeme sayısı, tonaj, LaneMetre, metrekare, metreküp birimlerinde seçilen veya tüm malzemelerin toplam bilgilerini sunacaktır.

Uygulama, ekranda taşınabilme kategorisine göre gruplandırılmış olarak yolcu, tekerlekli malzeme, konteyner, diğer, ton, LaneMetre, metrekare, metreküp verilerini bir rapor halinde sunacaktır. Bunu yaparken de toplamalarını da verecektir.

Kullanıcı malzemeleri kullanacak taşınacak birimleri de belirleyebilecektir.

Kullanıcı kendi kendine taşıma edecek araçları belirleyebilecektir.

Uygulama, kullanım/taşımaya esas boyutlar ile net ağırlık/brüt ağırlık değerlerini sunmalı, bunu yaparken de ekranda etkilenen tüm değerleri güncelleyebilecektir.

Uygulama, envanterin çıktısının alınmasına imkân tanıyacak bir rapor ön görüntüleme fonksiyonu sunacaktır.

Kullanıcı raporun gizlilik derecesini uygun şekilde belirleyebilecektir.

Kullanıcı rapora bir başlık ve yorum ekleyebilecektir. Uygulama raporda, Milli kod, adı, birliği, miktarı, brüt ve net ağırlığı, taşımaya esas boyutları (cm), kullanım boyutları (cm) veri alanlarını Kullanıcıya sunacaktır. Malzemelerin taşınabilme kategorilerine göre toplamları ve genel toplamları Kullanıcıya sunulacaktır. Rapor içeriği, taşınabilme kategorilerine göre belirlenecektir.

#### **9.10. Taşınacak Birim Taşımalarının Belirli Bir Tarihe Ötelenmesi**

Kullanıcı birliğe ilişkin sisteme girilmiş olan zamanları taşımanın başlaması emredilen tarih ve saate (G-Günü) göre öteleyebilecektir.

TPM, yapılan planlamalardaki zamanların belirli bir tarihe göre ayarlanmasına imkân tanıyacaktır.

#### **9.11. Taşınacak Birim Özellikleri**

Uygulama, birliğe ilişkin taşınacak birim kodu (UIC), profil kodu, bağlı olduğu müdürlük, önceliği, seri numarası, adı, son noktada bulunması istenen zaman (CRD), bulunduğu bölge, bilgileri Kullanıcıya sunulacaktır.

Uygulama bir birliğe ait tüm malzemelerin listesini REK, tanım, miktar, kategori, ağırlık, uzunluk, genişlik ve yükseklik bilgileri ayrıntısında sunacaktır.

Her bir malzeme için, Kullanıcı malzeme bilgilerine ulaşabilecek ve gerektiğinde değiştirebilecektir.

Uygulama tüm malzemelerin ve seçilen malzemelerin toplamlarını yolcu sayısı, malzeme sayısı, tonaj, LaneMetre, metrekare, metreküp ayrıntısında sunacaktır.

Kullanıcı malzeme listesi ve toplamlarda kullanım/taşımaya esas boyutlar ile net/brüt ağırlıklardan istediğine göre seçerek görüntülenmesini sağlayabilecektir.

Uygulama, seçilen taşınacak birimin her bir bileşenine ait verileri bileşen adı, taşınacak birim kontrol numarası (UCN), seçilen bileşenin taşıma listesi, seçilen bileşenin malzeme listesi bilgilerini içerecek şekilde sunacaktır.

Kullanıcı, her bir malzeme için malzeme bilgileri ara yüzüne ulaşabilecektir.

TÇN, TBN, ulařtırma modu, hazır olma en erken zamanı, hazır olma en ge zamanı, en erken ayrılıř zamanı, en ge ayrılıř zamanı, en erken varıř, en ge varıř zamanı verileri tařıma ayrıntıları olarak Kullanıcıya sunulmalı ve Kullanıcı, ara tařıma bacaklarını “Ara Tařıma Bacak Dzenleme” fonksiyonunu kullanarak dzenleyebilecektir.

Her bir bileřene ait malzemeler iin REK, tanım, miktar, kategori, ağırlık, uzunluk, geniřlik ve ykseklik bilgileri sunulacaktır. Seilen bileřene ait tm malzemelerin yolcu sayısı, malzeme sayısı, ton, LaneMetre, metrekare, metrekp řeklinde toplamları gsterilecektir.

Kullanıcı malzeme listesi ve toplamlarda kullanım/tařımaya esas boyutlar ile net/brt ağırlıklardan istediđine gre seerek grntlenmesini sađlayabilecektir

#### **9.12. Tařınacak Birim Envanter Dzenlemesi**

Kullanıcı birliđi bileřenlerine ayırabilecektir. Yeni bileřenin oluřturulmasını mteakip, Kullanıcı ana bileřenin tařıma gzerghını bu bileřene kopyalayabilmeli veya ana bileřenin sadece TBN bilgilerini kullanmasını sađlayabilecektir.

Kullanıcı bunu yaparken ana bileřenin zelliklerini grntleyebilecektir.

Uygulama, ana bileřene ait malzemeler iin REK, tanım, miktar, kategori, ağırlık, uzunluk, geniřlik ve ykseklik bilgilerini kullanım/tařımaya esas boyutlar ile net/brt ağırlıklardan istenilene gre seilerek grntlenmesini sađlayacaktır.

Uygulama aynı malzemeler iin, kullanım/tařımaya esas boyutlar ile net/brt ağırlıklardan istenilene gre seilecek toplam yolcu, malzeme, tonaj, LaneMetre, metrekare, metrekp bilgilerinin grntlenmesini yapılmasını sađlayacaktır.

Kullanıcı yeni bir bileřen ekleyebilecektir. Ana bileřenin malzeme listesinden malzeme seebilmeli ve bu malzemeleri alt bileřene atayabilecektir. Bunu da bir seferde yapabilecektir.

Uygulama, her bir bileřen iin malzeme listesini REK, tanım, miktar, kategori, ağırlık, uzunluk, geniřlik ve ykseklik ayrıntılarında kullanım/tařımaya esas boyutlar ile net/brt ağırlıklardan istenilene gre seilerek grntlenmesini sađlayacaktır.

Uygulama aynı malzemeler iin, kullanım/tařımaya esas boyutlar ile net/brt ağırlıklardan istenilene gre seilecek toplam yolcu, malzeme, tonaj, LaneMetre, metrekare, metrekp bilgilerinin grntlenmesini yapılmasını sađlayacaktır.

Kullanıcı alt bileşen için tercih edilen aracı belirleyebilmeli, çeşidini ve kodunu seçebilmeli ve tercih edilen araç tipinin özelliklerine ulaşabilecektir

Kullanıcı, sadece tercih edilen araçlarla taşınabilecek malzemeleri atayabilecektir (KURAL-10)

Kullanıcı bir bileşeni silebilecektir.

Listedeki her bir malzeme için, Kullanıcı malzeme özellikleri ara yüzüne ulaşabilecektir.

### **9.13. Taşınacak Birim Taşıma Güzergâhının Tersine Çevrilmesi**

Kullanıcı, seçilen bir taşınacak birim ya da bileşenin taşıma güzergâhını ters yönde olacak şekilde değiştirebilecektir.

### **9.14. Taşınacak Birim Taşımalarının Ulaştırma Araç Programı İle Eş Zamanlaması**

Kullanıcı, taşınacak birim taşıma zamanlarını birliğe tahsis edilen araçların programlarına göre bir seferde senkronize edebilecektir.

### **9.15. Taşınacak Birim Zamanlamalarının Otomatik Yapılması**

Uygulama, taşınacak birimlerin/bileşenlerin zaman planlamasının otomatik yapılmasını sağlayacak bir fonksiyon sunacaktır.

Kullanıcı, zamanlamayı ileri veya geri doğru yapabilecektir. İleri seçeneği seçildiğinde, Kullanıcı hazır olma zamanını en erken ayrılış zamanı veya belirlenecek tarih ve saate göre ayarlayabilecektir. Tarih ve zamanı belirlemek için Kullanıcı tarih ve saat veri alanlarını düzenleyebilecektir.

Geri seçeneği seçildiğinde, Kullanıcı tarih ve saati taşıma alanında bulunulması gerekli en son tarih ve saate veya belirlenecek tarih ve saate göre ayarlayabilecektir. Tarih ve zamanı belirlemek için Kullanıcı tarih ve saat veri alanlarını düzenleyebilecektir.

Hazır olma zamanı tarih ve saat girilerek düzenlendiğinde, bileşenin planlanmış hazır olma zamanı, ayrılış ve varış zamanları belirtilen zaman kadar ötelenecektir.

### **9.16. Taşınacak Birim Bileşen Zamanlamalarının Düzenlenmesi**

Kullanıcı girilecek zamanı gün, saat ve dakika olarak düzenleyebilecektir. Kullanıcı ileri ve geri öteleme/alma seçeneklerinden birini seçebilecektir.

Kullanıcı, zamanı en erken hazır olma zamanı, en geç hazır olma zamanı, en erken taşınmaya başlama zamanı, en geç taşınmaya başlama zamanı, en erken varış zamanı, en geç varış zamanı şeklinde düzenleyebilecektir. Öteleme/Geri alma zamanı girildikten sonra bileşenlerin/taşınacak birimlerin tüm taşıma zamanlarının uygun şekilde ötelenmesi/geri alınması sağlanacaktır.

#### **9.17. Taşınacak Birim Taşıma Zamanı Ötelemesi**

Kullanıcı, gün ve saat ve dakika verilerini belirleyerek zamanı daha erken veya daha geç bir tarihe öteleyebilecek/geri alabilecektir.

Kullanıcı en erken hazır olma zamanı, en geç hazır olma zamanı, en erken taşınma başlama zamanı, en geç taşınmaya başlama zamanı, en erken varış zamanı, en geç varış zamanlarından birini veya bir kaçını güncelleyebilecektir.

#### **9.18. Taşınacak Birim Taşıma Başlangıç ve Bitiş Alt ve Üst Sınırının Düzenlenmesi**

Uygulama, zamanlamada bir alt sınır ve üst sınırın gün, saat ve dakika olarak belirlenmesini sağlayacaktır.

Kullanıcı yeni zamanları gün, saat ve dakika olarak belirleyebilecektir.

Uygulama girilecek zamanlar arasındaki zaman farkını verecektir.

Kullanıcı otomatik zamanlamayı seçebilecektir.

#### **9.19. Taşınacak Birim Ulaştırma İhtiyaçları Analizi**

Ulaştırma ihtiyaçları analiz fonksiyonu şu alt işlemlere sahip olacaktır:

- Kargo Kategorisi
- Araç İhtiyaçları
- Ayrıntılı Analiz
- Ulaştırma ihtiyaçları analiz sonuçları

Uygulama taşımaya esas boyutlar ve brüt ağırlığa bağlı özete dayanarak taşınabilme kategorilerine göre gruplandırılmış yolcu, tekerlekli malzeme, konteyner, diğer, tonaj, metre, metrekare, metreküp toplamlarını verecektir. Uygulama aynı zamanda bu toplamlar için toplam envanter miktarlarını da gösterecektir.

Uygulama Kullanıcının kaç tane taşınacak birim seçtiğini gösterecek ve analiz raporu çıktısını alabilecektir.

Uygulama, bir veya daha fazla taşınacak birim için araç tipine göre gruplandırılmış yolcu, tekerlekli malzeme, konteyner, diğer, tonaj, metre, metrekare, metreküp toplamlarını verecektir.

Uygulama Kullanıcının kaç tane taşınacak birim seçtiğini gösterecek ve analiz raporu çıktısını alabilecektir.

Araç ihtiyaç raporu sonunda, kargonun ebatlarının belirlenmesine yardımcı olan maksimum boyutlar sunulacaktır.

Kullanıcı, hava, deniz, demir yolu ve kara yolu sorti/sefer ihtiyaçlarının ortaya konulması için araç seçimini yapabilecek ve yaptığı seçimleri tekrar kullanabilmek üzere kaydedebilecektir.

Kullanıcı sorti/sefer ihtiyaçlarını hesaplayabilecek ve bunu araç atama algoritması ile yapabilecektir.

Kullanıcı, analizini hem taşıma hem de envantere dayalı olarak gerçekleştirebilecek ve kendi kendine taşıma edebilen malzemeleri analize/analizden katıp/çıkarabilecektir.

Envantere göre analiz seçeneği seçildiğinde, taşıma güzergâhı için belirlenmiş ulaştırma moduna bakılmaksızın seçilen bileşenin envanterine göre hesaplamalar yapılacaktır.

Ulaştırma moduna göre analiz seçeneği seçildiğinde, sadece bileşen taşıma güzergâhında belirlenmiş ulaştırma modları için sorti ihtiyaçları hesaplanacaktır.

Bileşenin iki veya daha fazla hava taşınması varsa, ihtiyaçlar dikkatsiz bir plancının ihtiyacı iki veya daha fazla kat fazla hesaplamasına neden olmaması için iki veya daha fazla kez tekrarlanacaktır.

Ulaştırma moduna bağlı olarak araç tip seçimi varsayılan araç tercihlerine göre yapılacaktır.

Yük atama kuralları için taşıma planı yük tahsis parametrelerinin dikkate alınması yeterli olacaktır.

Hesaplama sonucunda uygulama, araç seçimindeki tüm ulaştırma modlarına ait sortileri sunacaktır.

Uygulama, seçilen taşınacak birimlerin sayısını verecektir.

Uygulama, araç tipine göre gruplandırılmış şekilde tahsisli sortiler ve kullanılan sortileri gösterecektir.

Uygulama, kargo atamasına göre gruplandırılmış yolcu, tekerlekli malzeme, konteyner, diğer, ağırlık, LaneMetre, metreküp, metrekare verilerini ve toplamalarını gösterecektir.

Uygulama, araç tipi ve kargo bölümü koduna göre gruplandırılmış kullanılan ve tahsis edilen sorti sayılarını verecektir.

#### **9.19.1. Ulaştırma İhtiyaçlarının Analizi Manifesto Sonuçları**

Hesaplama sonucu uygulama, araç seçiminde kullanılan tüm ulaştırma modları için manifestoları gösterecektir.

Kullanıcı araç tipi ve kargo bölümünü belirleyebilecektir.

Uygulama, tahsisli sortiler ve kullanılan sortileri gösterecektir.

Seçilen kargo bölümündeki malzeme listeleri için uygulama, REK, tanım, miktar, araç kodu, ayrılış, varış, kategori, ağırlık, uzunluk, genişlik ve yükseklik alanlarını kullanım/taşımaya esas boyutlar ile net/brüt ağırlık seçimine göre görüntüleyebilecektir.

Listedeki her bir malzeme için malzeme bilgileri ara yüzüne ulaşılabilecektir.

Seçilen malzemeler ve tüm malzemeler için, yolcu, malzeme, tonaj, LaneMetre, metrekare, metreküp verilerini kullanım/taşımaya esas boyutlar ile net/brüt ağırlık seçimine göre gösterecektir.

#### **9.19.2. Ulaştırma İhtiyaçlarının Tünel Kısıtları Sonuçları**

Uygulama, araç seçiminde işaretlendiği takdirde tünel kısıtlarını hesaplayacaktır. Hesaplama sonucunda, uygulama kısıt tipine göre gruplandırılmış yolcu, tekerlekli malzeme, konteyner, diğer, tonaj, metre, metrekare, metreküp verilerini ve bunların toplamalarını verecektir.

Kullanıcı bir tünel tipi (kısıt tipi) seçebilecektir. Seçimi müteakip, uygulama seçilen tünel tipine uygun malzemeleri belirlemeli ve her bir malzeme için, REK, tanım, miktar, taşınacak birim kodu, bileşen kodu, ayrılış, varış, kategori, ağırlık, uzunluk, genişlik ve yükseklik verilerini sunacaktır.

Uygulama kullanım/taşımaya esas boyutlar ile net/brüt ağırlık değerlerini listedeki malzemeler için gösterecek ve toplamlarını da verecektir. Bu fonksiyon, tahsis kuralları fonksiyonunu kullanacaktır.

Uygulama çıktısı alınabilir bir sorti raporunu Kullanıcıya sunacaktır.

### **9.19.3. Taşınacak Birim Envanterinin Tahsis Edilen Ulaştırma Aracından Boşaltılması**

Kullanıcı seçilen bir veya daha fazla taşınacak birim/bileşen taşıması için tahsis edilmiş olan tüm araçlardan tüm malzemeyi bir seferde boşaltabilecektir.

### **9.19.4. Taşınacak Birime Ulaştırma Aracı Tahsisinin Kaldırılması**

Kullanıcı bir seferde araç tahsislerinin iptalini bir veya daha fazla taşınacak birim/bileşen için gerçekleştirebilecektir.

## **10. Lokasyonlara ve Ulaştırma Alt Yapısına İlişkin Özellikler**

### **10.1. Duraklama Noktası Ekleme ve Özellik Düzenleme**

Bu fonksiyonla Kullanıcı sisteme yeni duraklama noktası ekleyebilecektir.

Kullanıcı mevcut Sorumluluk Sahası Lokasyonları Listesinden;

- Tüm Lokasyon Verileri (Havaalanları, limanlar, tren istasyonları, depolama bölgeleri, sınır geçiş noktaları ve diğer)
- Veri Kaynağı (Veri Tabanı veya Taşıma Planı)
- Ait Olduğu Ülke bilgilerine göre filtreleme yapabilecektir.

Kullanıcı duraklama noktasındaki bekleme süresinin saat olarak sisteme tanımlayabilecektir. (Varsayılan değer=0)

Kullanıcı ülke veya taşıma modu kısıtlarını belirleyebilecektir.

Kullanıcı duraklama noktasının, yükleme noktası merkezli taşımaların ve indirme noktasına kadar olan taşımaların yerlerini belirleyebilecektir.

Kullanıcı bu fonksiyonu kullanarak daha önce tanımlanmış bir duraklama noktası özelliklerinde değişiklik yapabilecektir.

### **10.2. Grafik Gösterimde Lokasyon Ekleme**

Uygulama, ülkeleri ve kullanılacak lokasyonlarını Kullanıcıya sunacaktır.



Kullanıcı, lokasyon tipini (Havaalanları, limanlar, tren istasyonları, depolama bölgeleri, sınır geçiş noktaları ve diğer) seçebilecektir.

Kullanıcı verinin kullanılacağı veri kaynağını (Veri Tabanı veya Taşıma Planı) belirleyebilecektir.

Kullanıcı belirli bir aktif bir bölgeyi enlem ve boylam bilgilerini sisteme girerek kullanabilecektir.

Kullanıcı grafiğe bir veya daha fazla lokasyon ekleyebilecektir.

### **10.3. Duraklama Noktası Silme**

Kullanıcı bu fonksiyonu kullanarak tanımlanmış bir duraklama noktasını silebilecektir.

### **10.4. Lokasyon Planlama**

Kullanıcı lokasyonlara ilişkin planlama yapabilecektir. Bu maksatla;

"İntikal Bacağı Düzenleme, Taşınacak Birim İntikal Başlangıç ve Bitiş Alt ve Üst Sınırının Düzenlenmesi, Lokasyon Silme" fonksiyonlarına ait özellikler bu fonksiyon için de kullanılabilir olacaktır.

### **10.5. Lokasyon Özellikleri**

Uygulama şu coğrafi lokasyon özelliklerini Kullanıcıya sunacaktır: "Kodu, adı, ait olduğu ülke, tipi, enlem/boylamı, UTM koordinatları, müsait olduğu ulaştırma modları ve açıklamalar".

## **11. Taşımalara İlişkin Özellikler**

### **11.1. Ulaştırma Modunun Değiştirilmesi**

Kullanıcı bir taşıma bacağının ulaştırma modunu değiştirebilecektir.

### **11.2. Taşıma Ekleme**

Kullanıcı iki lokasyon arasına taşıma ekleyebilecektir.

### **11.3. Taşıma Bacağı Düzenleme**

Kullanıcı, bir taşımanın zamansal verilerini güncelleyebilecektir.

- Uygulama,şu taşıma verilerini Kullanıcıya sunacaktır:
- Bileşen adı,
- Taşınacak birim Kodu,

- Taşınacak birim Kontrol Numarası,
- Seri Numarası,
- Taşınacak birim Adı
- TÇN
- TBN

Kullanıcı en erken ve en geç hazır olma zamanı, en erken ve en geç ayrılış zamanı, en erken ve en geç varış zamanı alanlarını “Zamanı Değiştir/Gir” fonksiyonu ile güncelleyebilecektir.

Kullanıcı en erken ve en geç zamansal verileri eş zamanlayabilecektir (senkronize edebilecektir).

Kullanıcı otomatik zamanlama fonksiyonunu kullanabilecektir.

Uygulama, tahsis edilen araçların programlarına uygun olarak taşınacak birim taşıma programlarını eş zamanlayabilecektir.

Uygulama, ulaştırma modunu ve mesafeyi (deniz mili, kara mili, km) verecektir.

Kullanıcı bileşenin kendi kendine taşıma edip edemediğini işaretleyebilecektir.

#### **11.4. Ulaştırma Aracı Taşıma Bacağı Düzenleme**

Kullanıcı bir aracın taşıma bacaklarına ilişkin zamansal verileri güncelleyebilecektir.

Uygulama, araç kodu, araç çeşidi, araç adı, araç tip kodu, bağlı olduğu terminal, gideceği terminal/nokta ve mesafe (deniz mili, kara mili, km) verilerini sunacaktır.

Kullanıcı hazır olma zamanı, ayrılış zamanı ve varış zamanı alanlarını “Zamanı Değiştir/Gir” fonksiyonu ile güncelleyebilecektir.

Kullanıcı otomatik zamanlama fonksiyonunu kullanabilecek/devre dışı bırakabilecektir.

Kullanıcı araçların bağlı oldukları terminalleri vb. tanımlarken veri tabanından veya mevcut taşıma planında kullanılan yerlerden seçim yapmasını sağlayacak bir lokasyon belirleme fonksiyonu kullanacaktır.

## **11.5. Taşıma Güzergâh Planlaması**

### **11.5.1. Otomatik Taşıma Planlama**

Uygulama, seçilen bir taşınacak birimin/bileşenin tüm taşımalarını gösterecektir.

Uygulama, bağlı olduğu terminal adı, gideceği yer adı, ulaştırma modu, kendi kendini taşıyıp taşıyamadığı, en erken ve en geç hazır olma zamanı, en erken ve en geç ayrılış zamanı, en erken ve en geç varış zamanı, taşınacak birim adı, taşınacak birim kodu, bileşen adı alanlarını Kullanıcıya sunacaktır.

Kullanıcı, taşımanın başlatılmasına ilişkin olarak hangi zamandan (en erken ve en geç hazır olma zamanı, en erken ve en geç ayrılış zamanı, en erken ve en geç varış zamanı) itibaren başlamak istediğini sisteme girecektir.

Kullanıcı taşımanın en erken zamandan ileriye doğru mu yoksa en geç zamandan geriye doğru mu yapmak istediğini sisteme tanımlayacaktır.

Kullanıcı mümkün olan boş zaman fasıllarının (available slack times) kullanılmasını sağlayabilecektir

### **11.5.2. Öncesine Lokasyon Ekleme**

Kullanıcı taşımanın seçilen her hangi bir bacağının çıkış lokasyonu öncesine yeni bir taşıma bacağı oluşturmak üzere yeni bir lokasyon ekleyebilecektir.

### **11.5.3. Sonrasına Lokasyon Ekleme**

Kullanıcı taşımanın seçilen her hangi bir bacağının varış lokasyonundan sonra yeni bir taşıma bacağı oluşturmak üzere yeni bir lokasyon ekleyebilecektir.

### **11.5.4. Taşıma Listesinin Çıktısını Alma**

Uygulama taşıma listesinin çıktısını alabilecektir.

Uygulama çıktıda şu verileri gösterecektir: Taşınacak birim Kodu, taşınacak birim adı, barış konuş yeri, sefer görev yeri, ulaştırma modu, en erken ve en geç hazır olma zamanı, en erken ve en geç ayrılış zamanı, en erken ve en geç varış zamanı.

### **11.5.5. Kaydetme**

Fonksiyon, "Plan Kaydet" fonksiyonunu kullanabilecektir.

Fonksiyon, Taşıma Bacağı Düzenleme fonksiyonunu kullanabilecektir.

Fonksiyon, Zaman Düzenleme/Girme fonksiyonunu kullanabilecektir.

Fonksiyon, Zaman Öteleme fonksiyonunu kullanabilecektir.

Fonksiyon, Taşıma Silme fonksiyonunu kullanabilecektir.

#### **11.6. Herhangi bir Taşıma Görevinin Tamamını Veya İlgili Taşıma Bacağını Silme**

Kullanıcı bir taşımayı ya da taşıma bacağını silebilecektir.

#### **11.7. Taşıma Planı Düzenleme**

Kullanıcı taşıma planını yeniden düzenleyebilecektir. Bu lokasyonları ve taşıma bacaklarını ilk planlama öncesindeki duruma dönüştürecektir.

#### **11.8. Taşımayı Tersine Çevirme**

Kullanıcı taşıma güzergâhını tersine çevirebilecektir. Bu fonksiyonun kullanılabilmesi için söz konusu taşımaya her hangi bir araç tahsisi yapılmamış olması gereklidir. Şayet tahsis yapılmışsa önce yüklü malzemelerin söz konusu araçtan boşaltılıp ve araç tahsisi kaldırılıp tersine çevrilebilecektir.

#### **11.9. Taşımaya Araç Tahsis Edilip Edilmediğinin Gösterilmesi**

Kullanıcı seçilen bileşenin taşınmasına ilişkin araç tahsislerini güncelleyebilecektir.

### **12. Plana İlişkin Özellikler**

#### **12.1. Taşınacak Birim Bileşen Envanterlerinin Değişik Araç Tiplerine Atanması**

Uygulama, taşınacak birim bileşen envanterlerindeki malzemenin değişik araç tipleri ve onların kargo bölümlerine tahsis edilmesini sağlayacaktır.

İlk prensip olarak, en yüksek öncelik olmak üzere yazılım ilk olarak öncelik derecesi en yüksek araç tiplerinin kargo bölümlerini doldurmaya çalışacaktır. Birden fazla araç tipi kullanan bir analiz yapıldığında, mevcut sorti miktarları sınırlandırılmadığı takdirde küçük araçların büyük araçlardan daha yüksek öncelikli kılınması önemlidir. Örneğin; Jumbo Kargo uçağı sortilerine öncelik derecesi olarak 1 girildiğinde ve Orta Kargo Uçağı sortileri için de 2 girildiğinde ikinciye sığan tüm kargo ilkinde de sığabildiğinden bütün kargo ilk uçağına yüklenmeye çalışılacaktır.

İkinci prensip olarak, benzer şekilde, birden fazla kargo bölümü olan araçlar için önceliği yüksek olan kargo bölümünden itibaren yüklenmeye başlanacaktır. Bu durumda da yüklemenin uygun yapılabilmesi için küçük kargo bölmelerine büyük olanlara göre daha büyük öncelik verilecektir. Örneğin, C-130 uçağının ana kargo bölümüne öncelik derecesi 1 ve rampa bölümüne 2 verilirse rampa bölümünde taşınabilen her türlü malzeme ana bölmede de tüm kargo ana bölme yükleneyecek ve rampa bölümüne yükleme yapılmayacaktır. Kargo bölümüne öncelik verilmesi işlemi VTYM içerisinde yapılacaktır.

Üçüncü prensip olarak, her bir araç tipi doldurulmaya çalışılırken, mevcut kargo bölmeleri tipleri ile envanter tiplerini optimum şekilde örtüştürebilmek için şu yaklaşım uygulanacaktır: (Muhtelif, yardımcı malzeme, ikmal maddesi ve tüketim malzemelerinin şekilsiz yük veya genel kargo olarak nitelendirilecektir)

- Yolcu: Sadece yolcu yüklenebilen kargo bölmelerine ve yolcu yükleme kapasitesine sahip her hangi bir kargo bölümüne (tamamen kargo ile doldurulmadığı takdirde),
- Tekerlekli Malzeme: Lane Metre ve Metrekare kapasiteli kargo bölmelerine ve Ton, Metreküp ve 20 Ayaklık Malzeme Birimi (TEU) kapasiteli kargo bölmelerine,
- Konteyner: 20 Ayaklık Malzeme Birimi (TEU) kapasiteli kargo bölmelerine ve Lane Metre, Metrekare ve Metreküp kapasiteli kargo bölmelerine,
- Dağınık Malzeme/Yük: Ton ve Metreküp kapasiteli kargo bölmelerine ve Lane Metre ve Metrekare kapasiteli kargo bölmelerine yüklenebilecektir.

Dördüncü prensip olarak, kargo bölümünün kapasitesi yüklenecek malzeme tipine ilişkin bir yükleme faktörü ile de düzenlenecektir. Örneğin, kargo bölümünün kapasitesi tekerlekli bir malzeme yüklendiği takdirde kargo bölümünün metreküp kapasitesi o malzemeye ilişkin yükleme faktörü ile çarpılmalı, şekilsiz veya dökme malzeme yüklendiğinde de kargo bölümünün metreküp kapasitesi o malzemeye ilişkin farklı bir yükleme faktörü ile çarpılacaktır.

Son prensip olarak, farklı birimlere göre kapasitelendirilmiş kargo bölümleri için, sistem tanımlanan her bir birime göre sığabilecek malzeme miktarını hesaplamalı ve sığabilecek en küçük miktarı raporlayacaktır. Bu nedenle, örneğin 100 LM'lik 225 metrekarelik yükleme alanına sahip bir kargo bölümüne 15 metrekare yer kaplayan 5 LM'lik bir kamyon yüklenmeye çalışıldığında sistem 20

yerine 15 araç yükleyecektir (Yükleme faktörü %100 kabul edildiğinde). Sistem, en küçük değer Metrekare opsiyonunda olduğundan, ona göre işlem yapmış olacaktır.

## **12.2. Taşıma Planının Kapatılması**

Kullanıcı bu fonksiyonu kullanarak aktif taşıma planını kapatacaktır.

## **12.3. Taşıma Planlarının Birleştirilmesi**

Bu fonksiyonu kullanarak Kullanıcı iki planı birleştirebilecektir.

Kullanıcı önce aynı taşıma planı altındaki bir taşıma planını seçecektir.

Kullanıcı birleştirmeyi yaparken benzer veya aynı taşınacak birimler için şu iki seçenekten birini seçecektir:

- Plandaki benzer veya aynı taşınacak birimler için taşınacak birim profilini aynen kullan yenileri kabul etme
- Taşınacak birim profilini sil ve yenileri ile değiştirme.

Kullanıcı birleştirmeyi yaparken benzer veya aynı ulaştırma araçları için şu iki seçenekten birini seçecektir:

- Plandaki benzer veya aynı ulaştırma araçları için mevcut araçları aynen kullanma
- Manifesto ve araç programlarını birleştir ile mevcut araçları sil ve yenileri ile değiştirme.

Şu taşıma planı detayları görüntülenecektir: Oluşturma Tarihi, Oluşturan, Oluşturan Birim, Değiştirme Tarihi, Değiştiren, Değiştiren Birim, Açıklamalar.

## **12.4. Taşıma Planı Silme**

Kullanıcı bu fonksiyonu kullanarak taşıma planı silebilecektir.

Kullanıcı önce mevcut bir taşıma planı ve daha sonara da silinecek taşıma plan(lar)ını seçebilecektir.

Şu plan detayları görüntülenecektir: Plan adı, plan tipi, taşımanın başlama zamanı ve adı, gizlilik derecesi ve açıklamalar.

Şu taşıma planı detayları görüntülenecektir: Oluşturma Tarihi, Oluşturan, Oluşturan Birim, Değiştirme Tarihi, Değiştiren, Değiştiren Birim, Açıklamalar.

## 12.5. Sistemden Taşıma Planı Aktarma

Taşıma planı aktarma fonksiyonu ile, Kullanıcı taşıma planını bir veri dosyası olarak kaydedecektir.

Kullanıcı bir dosya ismi belirleyebilecektir.

Kullanıcı taşınacak birim kuruluş ve malzeme bilgilerini de sadece ilk kez aktarıldığında veya taşınacak birim yapısına ilişkin veriler değiştiğinde taşıma planı içinde saklayabilecek ve bu opsiyon sistem tarafından varsayılan olarak seçili gelecektir.

Kullanıcı, istediğinde opsiyonel olarak tüm BİL ve BKL bilgilerini plan içinde saklayabilecektir.

Kullanıcı opsiyonel olarak sadece söz konusu planı ilgilendiren hususları kısmi opsiyonu altında aktarabilecektir.

Kullanıcı durum raporu bilgi seviyesi belirleyecektir. Bu opsiyon tüm seviyeler ile Level 1-4 arası değerlere sahip olacaktır.

Aktarma esnasında text rapor şeklinde sistem bir durum raporu oluşturacaktır. Kullanıcı bu raporu kaydedebilecek veya çıktısını alabilecektir.

Kullanıcı aktarmayı yaparken açıklamalar hanesini düzenleyebilecektir.

Aktarmada şu veri alanları görülecektir: taşıma planı adı, taşıma planı adı, Kullanıcı adı, aktarma tarihi

## 12.6. Sisteme Taşıma Planı İthal Etme

Kullanıcı taşıma planını ithal etmeden önce ne yapması gerektiğine ilişkin bir bildirim alacaktır. Müteakiben ithal edilecek taşıma planını seçecektir.

Kullanıcı ithal edilen taşıma planını sistemde kayıtlı her hangi bir idari plan ile ilişkilendirebilir/ithal edilen plandaki idari plan verilerini kabul edebilir ve mevcut ismi ile ithal edebilir/yeniden isimlendirebilir. Bu yapılırken seçilen taşıma planının taşıma başlangıç tarih ve saati görüntülenecektir.

Kullanıcı, aynı idari plana ilişkin bir dosyayı ithal etmek istediğinde "Mevcut İdari Plan ve Mevcut Taşıma Planı" seçeneğini işaretleyecektir. Bu opsiyon, "Taşıma Planına Yeni İsim Ver" veya "Mevcut Taşıma Planını komple değiştir" seçeneklerinden birine göre ayarlanacaktır.

Kullanıcı aynı zamanda;

- “Aynı Taşıma Başlama Tarih ve Saatini Kullan”
  - “ Taşıma Planındaki günleri ... gün ötele)”
  - “Taşıma Başlama Tarih ve Saatini Belirsiz Bırak/Belirsiz Çevir”
- seçeneklerinden birini işlem yapılmak üzere işaretleyecektir.

Kullanıcı, ithal esnasında çatışmaların giderilmesi için şu seçeneklerden birini işaretleyecektir:

- “Sadece Yeni Verileri İthal Et, Güncelleme”
- “Yeni Verileri İthal Et ve Eskileri Güncelle”.

Kullanıcı durum raporu bilgi seviyesi belirleyecektir. Bu opsiyon tüm seviyeler ile Level 1-4 arası değerlere sahip olacaktır.

Aktarma esnasında metin (text) rapor şeklinde sistem bir durum raporu oluşturacaktır. Kullanıcı bu raporu kaydedebilecek veya çıktısını alabilecektir.

İthal edilecek Taşıma Planına ilişkin olarak şu veriler görüntülenecektir: ithal edilen dosyanın adı, taşıma planı adı, taşıma planı adı, ithal eden, ithal tarihi, açıklamalar, taşımanın başlama zamanı, Taşınacak birim ve Malzeme verilerinin dâhil edilip edilmediği.

## **12.7. Yeni Taşıma Planı Oluşturma-1**

Kullanıcı; Taşınacak Birim İhtiyaç Listesi/Taşınacak Birim Yer Değiştirme Listesi kullanarak yeni plan oluşturma fonksiyonu yardımıyla yeni bir taşıma planı oluşturabilecektir.

Kullanıcı Taşınacak Birim Yer Değiştirme Listesi veya taşınacak birim listesi seçeneklerinden birini kullanarak taşıma planında kullanılacak taşınacak birim listesini belirleyebilecektir.

Kullanıcı şu zorunlu hususları belirleyebilecektir:

- Bir idari plan
- Mevcut bir Taşıma Planı
- Taşınacak Birim İhtiyaç Listesi içinden en az bir taşınacak birim
- Emsalsiz bir Taşıma Planı ismi
- Açıklamalar (opsiyonel)



Şu idari plan ayrıntıları görüntülenecektir: Plan Adı, Plan Tipi, Taşımanın Başlama Tarih ve Saati, Gizlilik Derecesi, Açıklamalar

Şu taşıma planı ayrıntıları görüntülenecektir: Oluşturma Tarihi, Oluşturan, Oluşturan Birim, Değiştirme Tarihi, Değiştiren, Değiştiren Birim, Açıklamalar.

Kullanıcı Taşınacak Birim Yer Değiştirme Listesinden taşıma planına taşınacak birim ekleyebilecektir.

## **12.8. Yeni Taşıma Planı Oluşturma-2**

Bu fonksiyonu kullanarak Kullanıcı yeni bir taşıma planı oluşturabilir.

Kullanıcı şu zorunlu hususları belirleyecektir:

- Bir taşınacak birim profili,
- İdari planlardan biri,
- Taşınacak Birim profilinden en az bir taşınacak birim,
- Emsalsiz bir Taşıma Planı ismi
- Açıklamalar (opsiyonel)

Şu İdari planı ayrıntıları görüntülenecektir: Plan Adı, Plan Tipi, Taşımanın Başlama Tarih ve Saati, Gizlilik Derecesi, Açıklamalar

Şu taşıma planı ayrıntıları görüntülenecektir: Oluşturma Tarihi, Oluşturan, Oluşturan Birim, Değiştirme Tarihi, Değiştiren, Değiştiren Birim, Açıklamalar.

## **12.9. Mevcut Bir Taşıma Planını Açma**

Kullanıcı bu fonksiyonu kullanarak mevcut taşıma planlarından birini açabilecektir.

Kullanıcı şu hususları belirlemek zorundadır:

- İdari Planlardan biri,
- Taşıma Planlarından biri,

Şu idari plan ayrıntıları görüntülenecektir: Plan Adı, Plan Tipi, Taşımanın Başlama Tarih ve Saati, Gizlilik Derecesi, Açıklamalar

Şu taşıma planı ayrıntıları görüntülenecektir: Oluşturma Tarihi, Oluşturan, Oluşturan Birim, Değiştirme Tarihi, Değiştiren, Değiştiren Birim, Açıklamalar.

## 12.10. Taşıma Planı Parametreleri

Kullanıcı bu fonksiyonu kullanarak parametreleri belirleyebilecektir. Bunu yaparken de sistem varsayılan değerlerini kabul edebileceği gibi kendisi yeni değerler de atayabilecektir.

Planlama parametreleri şu fonksiyonları yerine getirecektir:

- Plan Parametreleri: Planlama Faktörleri
- Plan Parametreleri: Grup araç Faktörleri
- Plan Parametreleri: Tahsis Kuralları
- Plan Parametreleri: Genel Araç Tercihleri

### 12.10.1. Taşıma Planlama Faktörleri

Kullanıcı şu planlama faktörlerini sisteme girebilecek veya sistem varsayılan değerlerini kullanabilecektir.

Hava yolu, iç su yolu, demir yolu, kara yolu, deniz yolu ulaştırma modlarına ait;

- Yükleme Erteleme/Rötar (sa)
- Boşaltma Erteleme/Rötar (sa)
- Ortalama Hız (km/sa)
- Yüklemede Malzemeler Arası Mesafe (m)
- Araç Grubu ve Araç Büyüklüğü.

### 12.10.2. Taşıma Planı Kara yolu Parametreleri

Kullanıcı şu grup araç faktörlerini sisteme girebilecek veya sistem varsayılan değerlerini kullanabilecektir.

Kara yolu grup araç özellikleri :

- Araç Sayısı ve yürüyüş taşınacak birimleri, yürüyüş grupları arasındaki zamanca mesafe,
- Molalar arasındaki zaman aralığı (dak.), ve ilk, ikinci ve üçüncü mola süreleri,
- Araçlar arasındaki mesafe (m),
- Araç başına düşen yolcu sayısı,

- Araç başına düşen taşınacak birim miktarı.

### 12.10.3. Taşıma Planı Yük Tahsis Kuralları Parametreleri

Kullanıcı şu tahsis kurallarını belirleyebilecektir:

- Tünel kısıtlamaları : Vagon Platform Yüksekliği (cm)
- Tonaj Tahsisi : malzeme tonaj limitlerini ve malzemeler arasındaki mesafeleri (cm) kullan/kullanma
- Konteynerizasyon: Kullan/Kullanma (araçlar ve maksimum taşıma tonajı) Kullan/Kullanma (Dökme/Şekilsiz Yük ve maksimum taşıma tonajı)
- Araçlara, konteynerlere ve dökme/şekilsiz yüklere ilişkin yükleme faktör değerleri (LaneMetre, Metrekare, Metreküp, Ağırlık (ton) yüzdeleri)
- LaneMetre Algoritması: Ortalama Malzeme Geniliği (cm).
- Tahsis etme opsiyonlarından (kullanım/taşımaya esas boyutlar ile net/brüt ağırlık) birer tanesini belirle.

Kullanıcı yükleme faktörlerini farklı malzeme tiplerinin farklı kargo bölmelerine tahsis edilebilmesi için güncelleyecektir.

İki yükleme faktörü seti mevcut olacaktır: İlk sette, sistem varsayılan değerleri bulunacak ve yetkili personel haricinde değiştirilemeyecektir. Kullanıcı tarafından belirlenen ikinci set değerler Kullanıcının ihtiyacına cevap verebilecek şekilde değiştirilebilecektir. Kullanıcı bu ikisinden birini seçmek zorunda olacaktır. Seçim yapılmadığında varsayılan değerler kullanılacaktır. Mesela, kapasitesi LaneMetre ve Metrekare olarak tanımlanmış kargo bölmelerine şekilsiz/dökme yükleri yüklemek istemediğinde Kullanıcı kendisi tarafından belirlenen faktörleri kullanabilecek ve LaneMetre ve Metrekare şeklindeki kargo için söz konusu bölümün yükleme faktörlerini sıfır olarak değiştirebilecektir.

Demir yolu tünel kısıtlarının değerlendirilebilmesi için vagon platform yüksekliği de kullanılabilir.

**Sınırlama:** Bu opsiyon seçildiğinde, yazılım tonaj kapasite değeri girilmiş kargo bölümlerine malzeme yüklenirken ilave kontroller yapması sağlanacaktır. Yazılım, maksimum tonaja ulaşılmadan önce mevcut kargo bölümü büyüklüğünün aşılıp aşılmadığını kontrol edecektir. Bu, özellikle kargo bölümünün taban büyüklüğü maksimum tonaja ulaşılmadan önce dolduğunda geçerli olacaktır. Yazılım, bunu

yaparken malzemeler arasındaki istif mesafesinin de belirlenmesini sağlayacaktır. Bu mesafe ne kadar büyük olursa kargo bölümünün maksimum tonaj ulaşılmadan dolması da o kadar erken olacaktır. Bu opsiyon uygulandığında, kargo bölümünü ebatlarının doğru olarak belirlenmesi büyük önem arz etmektedir.

**Konteynerizasyon:** Bu opsiyon etkinleştirildiğinde, ekipman ve şekilsiz/dökme yükler kargo bölmelerine ebatları konteynere sığıdığı farz ve kabul edilerek 20 Ayaklık Malzeme Birimi (TEU) olarak yüklenecektir. Aksi takdirde yükleme yapılmayacaktır.

**Tahsis Esası:** Kullanım/taşımaya esas boyutlar ve net/brüt ağırlık opsiyonları dikkate alınarak bu iki farklı durum için envanterin araçlara sığıp sığmadığı kontrol edilecektir.

#### **12.10.4. Taşıma Planı Taşınabilirlik Analizi Yapma**

Kullanıcı, şu genel araç tercihlerini belirleyecektir:

- Kullanıcı, kara yolu, demir yolu, hava yolu, deniz yolu ulaştırma modlarından birine ait bir araç tipi belirleyebilecektir
- Kullanıcı her bir araç tipi için öncelik ve maksimum sorti miktarlarını belirleyecektir (varsayılan değer sorti miktarları için 999 olacaktır)

#### **12.10.5. Taşıma Planı Tercihleri**

Kullanıcı bu fonksiyonu kullanarak tercihleri belirleyebilecektir.

- Kullanıcı, şu opsiyonları belirleyebilecektir:
  - İlgili sahası (\*.pro)
  - Deniz, hava, kara, iç su yolu, demir yolu ulaştırma ağı görüntülenebilecektir.
- Şu planlama ayarları yapılabilecektir:
  - Otomatik zamanlama
  - Planlama Faktörlerini Göster
  - Araçlar tam sefer yapabilsin
  - Çok modlu taşımaya izin ver
  - Manifesto uygunluğunu kontrol et
- Diğer kontroller

- Araç mesafe uygunluk kontrolü
- Aracın mevcut konumuna göre mesafe uygunluk kontrolü
- Araç geri dönüş noktası mesafe uygunluk kontrolü
- Şu veri tabanı opsiyonları seçilebilecektir:
  - Veri tabanına kaydetmeden önce uyarı kontrolünü devre dışı bırak
  - Otomatik zaman uyarıcısını kullan
  - Otomatik zaman uyarıcısı uyarı aralıklarını belirle (dak.)
- Şu görüntü ayarları yapılabilecektir:
  - Zaman gösterim fonksiyonunu kullan (referans tarihine göre ayarla, gerçek zamana göre ayarla). (Gerçek zaman ZULU ve lokal zamana göre otomatik düzenlenebilecektir)
  - Mesafe birimi deniz mili, km ve kara mili şeklinde seçilebilecektir.
  - Malzeme/ekipman/taşınacak birim için isminin mi yoksa kısa isminin mi kullanılması seçilebilecektir.
  - Bileşen için bileşen numarasının mı yoksa bileşen adının mı kullanılacağı belirlenebilecektir.
  - Malzeme için milli koda göre mi yoksa REK koduna göre mi görüntüleme yapılacağı belirlenebilecektir
  - Bu opsiyon durum çubuğunda verinin bulunup getirilmesini sağlayacaktır.

#### **12.10.6. Taşıma Planı Çıktılarının Alınması**

Planın çıktısının alınması fonksiyonu ile Kullanıcı seçilen taşınacak birime ait verilerin çıktısını alabilecektir.

Kullanıcı aşağıdaki seçeneklerden seçim yapabilecektir.

- Taşınacak Birim
- Bileşen
- Taşıma Güzergâhı

- Araç Programları
- Araç Manifestoları

Kullanıcı, “Envanteri dâhil et” opsiyonunu seçebilecektir.

Taşıma planında taşınan birimlere ilişkin çıktılar aşağıdaki esaslar dâhilinde alınabilecektir:

- Uygulama, Kullanıcıya “taşınacak birim kodu, adı, ülkesi, seri numarası, konuş lokasyonu, önceliği, istenen son varış tarihi” verilerini sunacaktır.
- Taşınacak birim envanterine ilişkin çıktıların alınması:
- Uygulama, envanter çıktısı olarak değerlendirilecek bir raporu görüntüleyebilecektir.
- Uygulama bu kapsamda, kodu, adı, ait olduğu birim, miktarı, brüt ve net ağırlık, taşımaya esas ve kullanım boyutlarını (uzunluk, genişlik ve yükseklik olarak (cm)) Kullanıcıya sunacaktır.
- Uygulama, Kullanıcı tarafından belirlenecek taşınabilme kategorilerine göre toplamları ve genel toplamı verecektir.

Taşıma planında taşınan birim bileşenlerine ilişkin çıktılar kapsamında uygulama, Kullanıcıya bileşenin “taşınacak birim kodu, adı, ülkesi, seri numarası, konuş lokasyonu, önceliği, istenen son varış tarihi” verileri yanında birim kontrol numarası, bileşen adı verilerini de sunacaktır.

Taşıma planı taşıma güzergâhına ilişkin çıktılar kapsamında uygulama, Kullanıcıya güzergâh planlama raporu sunacaktır. Bu raporda her bir taşınacak birim için birim kodu ve adı ile bu birim bileşenlerinin adı kontrol numaraları yer alacaktır. Bu kapsamda; tüm bileşen taşımaları için “konuş yeri, varış yeri, ulaştırma modu, kendi kendine taşıma etme durumu (E/H), ilk ve son hazır olma zamanları, ilk ve son ayrılış ile ilk ve son varış zamanları da bu raporda yer alacaktır. Bu rapor kapsamında;

- Uygulama, Envanterli güzergâh raporunu Kullanıcıya sunacaktır. Bu rapor güzergâh planlama raporu ile aynı özelliklere sahip olacaktır. Bu kapsamda, bileşen envanterindeki malzemelere ilişkin olarak, “REK kodu, malzeme tanımı, kategori tipi, miktarı, brüt ve net ağırlık, taşımaya esas ve kullanım boyutlarını (uzunluk, genişlik ve yükseklik olarak (cm)) verileri sunulacaktır.

- Uygulama her bir bileşen için yolcu, malzeme, ton, lane matre, metrekare, metreküp olarak toplamları verecektir.

Plana tahsis edilen araç programına ilişkin çıktılar kapsamında;

- Uygulama Kullanıcıya araç programı raporu sunacaktır. Bu raporda her bir taşınacak birim için birim kodu ve adı ile bu birim bileşenlerinin adı kontrol numaraları yer alacaktır. Bu kapsamda; tüm bileşen taşımaları için “konuş yeri, varış yeri, ulaştırma modu, kendi kendine taşıma etme durumu (E/H), ilk ve son hazır olma zamanları, ilk ve son ayrılış ile ilk ve son varış zamanları da bu raporda yer alacaktır.

- Uygulama, bileşenlere tahsisli tüm ulaştırma araçları için “araç kodu, adı, ayrılış ve varış tarih ve zamanları” verilerini sunacaktır.

- Envanteri de gösteren araç programları raporu:

- Araç programları raporu ile aynı özelliklere sahip olacaktır. Bu kapsamda, bileşen envanterindeki malzemelere ilişkin olarak, “REK kodu, malzeme tanımı, kategori tipi, miktarı, brüt ve net ağırlık, taşımaya esas ve kullanım boyutlarını (uzunluk, genişlik ve yükseklik olarak (cm) verileri sunulacaktır.

- Uygulama her bir bileşen için yolcu, malzeme, ton, lane matre, metrekare, metreküp olarak toplamları verecektir.

Plana tahsis edilen araç manifestosuna ilişkin çıktılar kapsamında:

- Uygulama Kullanıcıya araç manifestosu raporu sunacaktır. Bu raporda her bir taşınacak birim için birim kodu ve adı ile bu birim bileşenlerinin adı kontrol numaraları yer alacaktır. Bu kapsamda; tüm bileşen taşımaları için “konuş yeri, varış yeri, ulaştırma modu, kendi kendine taşıma etme durumu (E/H), ilk ve son hazır olma zamanları, ilk ve son ayrılış ile ilk ve son varış zamanları da bu raporda yer alacaktır.

- Uygulama, bileşenlere tahsisli tüm ulaştırma araçları için “araç kodu, adı, ayrılış ve varış tarih ve zamanları” verilerini sunacaktır.

- Envanteri de gösteren araç manifestosu raporu

- Bu rapor araç programları raporu ile aynı özelliklere sahip olacaktır. Bu kapsamda, bileşen envanterindeki malzemelere ilişkin olarak, “REK kodu, malzeme tanımı, kategori tipi, miktarı, brüt ve net ağırlık, taşımaya esas ve kullanım boyutlarını (uzunluk, genişlik ve yükseklik olarak (cm)) verileri sunulacaktır.

o Uygulama her bir bileşen için yolcu, malzeme, ton, lane matre, metrekare, metreküp olarak toplamları verecektir.

#### 12.10.7. İdari Plan Özellikleri

İdari Plan ayrıntıları Kullanıcıya sunulacaktır:

- İdari Plan Kodu
- İdari Plan Tipi
- Gizlilik Derecesi
- İdari Plan Adı
- İdari Plan Başlama Zamanı (Referans Tarihi)(Gün, saat, dakika)
- Taşıma Başlama Tarihi (G-Günü) (Gün, saat, dakika)
- Açıklamalar

Şu Taşıma Planı ayrıntıları Kullanıcıya sunulacaktır:

- TP Kodu
- Kullanıcı Kodu
- Oluşturma Tarihi
- Oluşturan Personel
- Oluşturan Birim
- Değiştirme Tarihi
- Değiştiren Personel
- Değiştiren Birim

Kullanıcı TP'in açıklamalar bölümünü değiştirebilecektir. Şu özet bilgiler görüntülenecektir:

- Taşınacak birim Miktarı
- Bileşen Miktarı
- İcra Edilen Taşıma Miktarı
- Kullanılan Ulaştırma Aracı Miktarı
- Tahsisli Araç Programı Miktarı



- Taşınan Ekipman Malzeme Miktarı
- Taşınan Ekipman Tonajı
- Taşınan Yolcu Miktarı

#### **12.10.8. Taşıma Planı Veri Tabanına Kaydedilmeden Önce Değişikliklerin Geri Alınması**

Bu fonksiyon kullanılarak en son kayıttan sonra ve yazılım kapatılmadan önce yapılan değişikliklerin veya kaydet butonuyla kaydedilmesi istenen değişikliklerin veri tabanında değiştirilmeyerek geri alınmasını sağlayacaktır.

#### **12.10.9. Taşıma Planının Kaydedilmesi**

Bu fonksiyon, yeni oluşturulan veya değiştirilmek üzere açılmış olan bir taşıma planının ayrıntılarının kaydedilmesini sağlayacaktır.

#### **12.10.10. Taşıma Planının Başka Bir İsimle Kaydedilmesi**

Bu fonksiyon, mevcut bir taşıma planının veri tabanında farklı bir isimle kaydedilmesini sağlayacaktır.

### **13. Görüntülemeye İlişkin Özellikler**

#### **13.1. Taşınacak Birimlerin Hiyerarşik Yapılarının Görüntülenmesi**

Taşınacak birimin yapısal hiyerarşisi ve taşıma ihtiyaç hiyerarşisi uygulama tarafından Kullanıcıya verilecektir.

Kullanıcı bu birimi ve bağlı birimlerini listeden seçebilecektir. (Bknz. TAŞINACAK BİRİM LİSTESİNİN GÖRÜNTÜLENMESİ)

Yapısal Hiyerarşi: Uygulama, bütün birimleri Kullanıcıya varsa aralarındaki yapısal hiyerarşik yapıyı da gösterecek şekilde sunacaktır.

Taşıma İhtiyaç Hiyerarşisi: Bütün taşıma ihtiyaçları hiyerarşinin kökünde gösterilecektir.

#### **13.2. Taşınacak Birim Listesinin Görüntülenmesi**

Uygulama taşınacak birimlerin bir listesini sunacaktır

Uygulama Kullanıcıya şu verileri sunacaktır: Taşınacak birim kodu, adı ve bağlı olduğu ülke, en geç varış zamanı, seri numarası, bileşen sayısı, gideceği son nokta, plan profil adı, en erken ayrılış zamanı.

Kullanıcı bir veya daha fazla birimi taşınacak birim taşıma listesinden seçebilecektir. (Bknz. TAŞINACAK BİRİMLERİN LİSTEDEN SEÇİLMESİ)

Taşınacak birime ilişkin olarak grafik ve GANTT gösterim Kullanıcıya sunulacaktır. (Bknz. GANTT GÖSTERİMİNDE ZAMANSAL GÖSTERİMİN DÜZENLENMESİ ve TAŞIMA PLANI İLGİ SAHASI GRAFİK (HARİTA) GÖSTERİMİ)

### **13.2.1. Taşınacak Birimlerin Listeden Seçilmesi**

Kullanıcı taşınacak birimleri şu kriterlere göre seçebilecek/seçimi kaldırabilecektir: Taşınacak birim tipi, ait olduğu ülke, seri numarası, önceliği, birim belirleme kodu, istenen son dağıtım tarihi, en erken ayrılış zamanı, bileşen adı, bileşen ulaştırma modu, bileşen konuş yeri, bileşen varış yeri ve son gidilecek nokta.

- Kullanıcı seçimi ters çevirebilecektir.
- Kullanıcı bütün birimleri seçebilecektir.
- Kullanıcı tüm birimlerin seçimini iptal edebilecektir.

### **13.2.2. Taşınacak Birimlere İlişkin Filtreleme Yapılabilmesi**

Kullanıcı taşınacak birimleri şu kriterlere göre filtreleyebilecektir: Taşınacak birim tipi, ait olduğu ülke, seri numarası, önceliği, birim belirleme kodu, istenen son dağıtım tarihi, en erken ayrılış zamanı, bileşen adı, bileşen ulaştırma modu, bileşen konuş yeri, bileşen varış yeri ve son gidilecek nokta.

Kullanıcı filtrelemeyi tek adımda temizleyebilecektir.

### **13.2.3. Taşınacak Birim Taşımalarının Hiyerarşik Olarak Görüntülenmesi**

Uygulama, Kullanıcıya taşınacak birim taşımalarının hiyerarşik olarak şu özellikleri gösterecek şekilde listelenmesini sağlayacaktır: Taşınacak birim adı, bileşenleri, ulaştırma aracı taşıma bacağı.

## **14. Gantt Gösterim Özellikleri**

### **14.1. Gantt Gösteriminde Zamansal Gösterimin Düzenlenmesi**

Kullanıcı GANTT gösteriminde zamansal gösterime ilişkin düzenleme yapabilecektir.

### **14.2. Gantt Gösteriminde Zaman Göstergesinin Görüntülenmesi**

Kullanıcı GANTT gösteriminde zaman göstergesini görüntüleyebilecek, görüntüyü iptal edebilecektir.

#### **14.3. Gantt Gösterimi Seçim Modu**

Kullanıcı GANTT görünümünü seçim moduna getirebilecektir.

#### **14.4. Gantt Gösterimi Zamansal Düzenleme**

Kullanıcı GANTT gösteriminde zaman değişikliği yapabilecektir.

#### **14.5. Gantt Gösterimi Zaman Girme**

Kullanıcı, GANTT gösteriminde zaman girebilecektir.

#### **14.6. Gantt Gösterimi Eş Zamanlaması**

Yazılım, Kullanıcının “Araç Programı ile eşzamanla” seçeneğini kullanarak planlı taşıma bileşen zamanlamalarının otomatik olarak tahsis edilen araç programı zamanlamalarıyla eş zamanlamasını yapabilecektir.

Olası bir çatışma durumunda, tahsis edilen aracın ayrılış, varış ve hazırlık zamanları değişmezken, bileşene ilişkin zamanlamalar yazılım tarafından düzenlenecektir.

İhtiyaç duyulduğunda, ulaştırma aracı zamanlamaları Kullanıcı tarafından değiştirilecektir.

Uygulama, aynı bileşene ait müteakip taşıma bacağı programlarında ortaya çıkabilecek önceki taşıma bacağının varışından önce sonraki bir taşıma bacağının taşınmaya başlatılması gibi çatışma durumlarında Kullanıcıyı ikaz edecektir.

İstenen en son varış zamanından sonra şayet bir taşıma tamamlanıyorsa Kullanıcı yazılım tarafından ikaz edilecektir.(Bknz. KURAL-1)

#### **14.7. Gantt Gösteriminin Yolcu (Pax) ve Malzemeye (Ton) Göre Düzenlenmesi**

Uygulama, her bir taşınacak birim ve bileşen için Tonaj ve yolcu GANTT gösterimini sunabilecektir.

Bu işlev, GANTT gösteriminde her hangi bir taşımanın üzerine sağ tıklanarak yapılabilecektir.

#### **14.8. Gantt Gösteriminde Taşınacak Birim Taşıma Programının Görüntülenmesi**

Uygulama taşınacak birimin taşıma programını GANTT gösteriminde taşınacak birimler, bileşenler, taşımalar ve programlar şeklinde seviye seviye sunacaktır. Yaklaştır/Uzaklaştır seçeneği de burada Kullanıcıya sunulacaktır.

## **15. Grafik Gösterim Özellikleri**

### **15.1. Taşıma Planı Sorumluluk Sahası Grafik (Harita) Gösterimi**

Kullanıcı bir harita seçebilecektir. Bu işlem bir dünya haritası üzerinde sorumluluk sahası belirlenerek yapılabilecektir.

### **15.2. Taşıma Planının Harita Üzerinde Grafik Gösterimi**

Uygulama, üzerinde taşıma bacaları ile lokasyonlarında yer alacağı bir harita görüntüsünü Kullanıcıya sunacaktır. Uygulama, harita üzerinde farenin geldiği her hangi bir noktanın enlem ve boylamını sunacaktır. Bu nokta Kullanıcı tarafından işaretlenebilecektir.

## **16. Yazılım Kuralları**

### **16.1. Kural-1**

Taşıma bacağıının varış zamanı CRD'den sonra olmayacaktır.

Taşıma bacağıının varış zamanı, aynı bileşenin bir sonraki taşımalarının ayrılış zamanından sonra olmayacaktır.

Taşıma bacağıının ayrılış zamanı, aynı bileşenin bir önceki taşımalarının varış zamanından önce olmayacaktır. (Bknz. GANTT GÖSTERİMİ EŞZAMANLAMASI)

### **16.2. Kural-2**

Bu seçenek işaretlendiğinde, tahsis fonksiyonu taşımanın TÇN ile TBN arasındaki mesafenin tahsis edilen aracın hareket sırasından fazla olup olmadığını kontrol edecektir. Fazla olduğu durumda, fonksiyon mevcut duraklama noktaları içinden uygun bir duraklama noktası seçmeye çalışacaktır. Mesafe aşıldığında ve sistem uygun bir duraklama noktası bulamadığında bunu Kullanıcıya rapor edecektir.

Bu seçenek işaretlenmediğinde tahsis fonksiyonu mesafe uygunluğunu dikkate almayacaktır.

- Araç Pozisyonuna Göre Mesafe Kontrolü: Bu opsiyon araç mesafe kontrolü seçeneği işaretlendiğinde kullanılabilir olacaktır. Bu opsiyon işaretlendiğinde tahsis fonksiyonu aracın mevcut konumu ile TBN arasındaki mesafeyi dikkate alacak ve bu mesafenin aracın sırasını aşıp aşmadığını kontrol edecektir. Aştığı durumda, fonksiyon mevcut duraklama noktaları içinden uygun bir duraklama noktasını seçmeye çalışacaktır. Mesafe aşıldığında ve sistem uygun bir

duraklama noktası bulamadığında bunu Kullanıcıya rapor edecektir. Şayet araç belli bir zaman aralığında tahsis edilmemişse, bu seçenek boş bırakılacaktır. Örneğin; bir uçak Amsterdam ile Cezayir arasındaki bir taşımaya ayın 5'inde tahsis edilmiş olsun. Aynı uçak ayın 15'inde de Los Angeles ile Tokyo arasındaki bir taşımaya tahsis edilmiş olsun. Bu durumda Cezayir ile Tokyo arasına bir duraklama noktası koymak mantıksız olacaktır. Zira söz konusu araç ayın 5'i ile 15'i arasında başka taşımalara da tahsis edilebilir. Bu nedenle bu opsiyon sadece uçağın belirtilen taşımalarından başka bir taşıma programı olmadığı durumlarda kullanılacaktır.

- Gerİ Dönüş Noktasına Göre Mesafe Kontrolü: Bu opsiyon araç mesafe kontrolü seçeneği işaretlendiğinde kullanılabilir olacaktır. Bu opsiyon işaretlendiğinde tahsis fonksiyonu dönüş mesafesinin (aracın döneceği noktayı dikkate alarak) aracın sığasına uygun olup olmadığını kontrol edecektir. Uygun olmadığı durumda, fonksiyon mevcut duraklama noktaları içinden uygun bir duraklama noktasını seçmeye çalışacaktır. Mesafe aşıldığında ve sistem uygun bir duraklama noktası bulamadığında bunu Kullanıcıya rapor etecektir.

- Araç Gidiş-Gelişinin (Sorti/Sefer) Kontrolü: Bu opsiyon, tahsisin tekrarlanıp tekrarlanmayacağını kontrol etmek üzere Kullanıcı tarafından tahsis et/ata düğmesine kaç kez basıldığını belirlemek için kullanılır. Bu opsiyon seçildiğinde, araç tahsisi şayet halen taşınabilir konumda kargo var ise yeni araç program(lar)ı oluşturularak tekrarlanacaktır. Bu opsiyon seçilmediğinde tahsis et düğmesine bir kez basıldıktan sonra ara yüz kapanacaktır.

- Uygunluk Kontrolünün İptali: Bu opsiyon işaretlenmediğinde, sistem bütün uygulanabilirlik (fizibilite) kontrollerini yaparak uygunsuz bir aracın tahsis edilmesini engelleyecektir. Seçildiğinde ise, hiçbir fizibilite kontrolü yapılmayarak işlem gerçekleştirilecektir.

- Manifesto Kontrolleri: Bu opsiyon işaretlenmediğinde sistem, bileşen taşımalarının ilk hazır olma zamanından itibaren bütün kargonun yüklemeye müsait olduğunu anlayacaktır. Seçildiğinde ise, her bir kargoya önceden tahsis edilen araç programlarına bağlı olarak yüklenebilme zamanlarının doğrulamasını yapacaktır. Örneğin; bu kontrol işaretlendiğinde, bir tank yükleme bölgesine ayın 10 unda varıyorsa ve söz konusu tankın yüklenmesi planlanan gemi 8 inde ayrılıyorsa bu tank yüklenmeyecektir. Bu kontrol işaretlenmediğinde ise yüklenecektir.

### **16.3. Kural-3**

Bir aracın taşıma programları manifestoları varsa yeniden düzenlenemez. Sadece manifestosu olmayan programlar silinebilir. Toptan bir deęişiklik için kargonun söz konusu araçtan boşaltılması gereklidir.(Bknz. ULAŞTIRMA ARACI GÜZERGÂH DÜZENLEME)

### **16.4. Kural-4**

Araç tahsis edilmiş taşımaları bulunan bileşenler alt bileşenlere ayrılamaz. Yeniden düzenlenebilmeleri için önce söz konusu araçtan boşaltılmaları gerekir.

### **16.5. Kural-5**

Bir bileşen, alt bileşenlerine ayrılırsa, ana bileşenin tüm malzemeleri yeni bileşenlere dağıtılacaktır.

### **16.6. Kural-6**

Zamanları öteleyebilmek için, mutlaka bir hareket başlangıç tarih ve saati belirlenmiş olacaktır.

### **16.7. Kural-7**

Bileşenin araç tahsis edilmiş taşımaları varsa, taşıma güzergâhı tersine çevrilemez.

### **16.8. Kural-8**

Bileşenlerin araç tahsis edilmiş taşımaları geri çevrilemez. Bunun yapılabilmesi için önce söz konusu araçtan boşaltılması gereklidir

### **16.9. Kural-9**

Birden çok taşımaya yönelik araç tercihlerini yazmak için seçilen tüm taşımaların aynı ulaştırma modunda olması gereklidir.

### **16.10. Kural-10**

Yeni bileşenin oluşturulmasını müteakip, Kullanıcı sadece tercih edilen araçla taşınabilecek malzemelerin atamasını yapabilir.

## **17. Ulaştırma Planlamalarında Kullanılacak Yükleme Planlama Esasları**

### **17.1. Kara Yolu Yükleme Planlama Esasları**

#### **17.1.1. Ulaştırma Aracına İlişkin Esaslar**

- Kargo bölümünün tipi
- Yükleme /boşaltma süresi
- Yükleme Platformunun yüksekliği
- Kargo bölümünün yüklenme önceliği
- Kargo bölümünün adı
- Kargo bölümü depo tipi (tenteli, kapalı, açık, platform)
- Kargo bölümünün uzunluğu/yüksekliği/genişliği, (LM olarak tanımlama yapıldığında araç yüklenebilir anlamına gelecek ve malzemenin LM olarak değeri sisteme girildiğinde kaplayacağı toplam alan (taşımaya esas ebatlar+sabitlenme için kullanılacak alan) ile eşleştirilmesi sağlanacaktır)
- Kargo bölümüne yüklenebilecek maksimum kargo ebatları (uzunluğu, genişliği, yüksekliği) ve ağırlığı
- Kargo bölümü maksimum yük taban dayanım direnci
- Kargo bölümünün maksimum istif yüksekliği ve ağırlığı
- Kendi kendine yükleme boşaltma özelliği, kapasitesi ve mevcut ekipmanları
- Aracın yük denge durumu
- Tehlikeli madde taşıyabilme özelliği

#### **17.1.2. Malzeme Yükleneşine İlişkin Esaslar**

- Malzemenin ebatlarının yüklenecek bölmeye uygunluğunun kontrolü [en, boy, yükseklik, ağırlık)
- Taşıma koşulları kontrolü (İklim Kontrollü Bölme, açık bölme, kapalı bölme)
- Malzemenin taşınma önceliği (sevk önceliği),
- Malzeme Özelliği (Aynı özellikteki malzemeleri taşınacak birimde götürmek için)

- Tehlikeli madde olup olmadığı ve taşıma koşulları [bir arada taşınamayacak malzemelerin kontrol edilerek uygun şekilde yükleme yapılması (compatibility analysis)]
- Yükün cinsi (Dökme, paket, şekilsiz, ekipman, ikmal maddesi)
- Her bir yükün diğerlerinden ne kadar mesafede olması gerektiği bilgisinin değerlendirilmesi (tehlikeli maddeler, büyük ebattaki malzemeler)
- Taşınamayan yüklerin en maliyet etkin hangi ulaştırma vasıtasıyla taşınacağı belirlenmesi (cost-efficient transportability analysis),
- Araca malzeme yüklenmeye başlandığında yüklemenin ilgili kargo bölümünün neresinden başlayacağı ve devam edeceği bilgisinin kontrolü.

### **17.1.3. Malzemenin İstiflenmesine İlişkin Esaslar**

- Malzeme bağımlı üst üste maksimum istifleme (Aynı malzemenin üst üste istif miktarı).
- Malzemenin üstüne konulabilecek maksimum ağırlık miktarı (Farklı tip malzemeler için).
- Yükleme bölgesi üst üste maksimum istifleme yüksekliği ve ağırlığı.
- Tek başına taşınması gereken malzemenin belirlenerek ilgili araca yüklenmesi (örn.; dozer, tank)
- Yükler arası boşlukların belirlenmesi (yükleme esnasında her malzemenin zorunlu boşluk miktarı farklı olduğundan uygulamada en büyük değer esas alınacaktır)
- Dengenin sürekli korunması için aynı bölüme yüklenecek malzemelerin yükleme sırası (her bir malzemenin yüklenmesini müteakip genel araç denge kontrolü yapılması)
- Malzemenin yükleme önceliği (tahsis edildiği araca yüklenmede önceliği)

### **17.1.4. Genel Yükleme Esasları**

- Her bir yükün araç içerisinde yerleştirileceği platform, deck vs. üzerindeki yerinin koordinatlarına varıncaya kadar ayrıntılı olarak yükleme şeklinin görülmesi,



- Manuel Yükleme
- Otomatik Yükleme
- Ön Tahsisli Yükleme (Müteakip terminallerdeki yükleme boşaltma faaliyetleri dikkate alınarak gerektiğinde boş alan tahsisi)
- Önceki/Müteakip Terminal(ler)e göre yükleme (Malzeme Biniş/İniş Sırası)
- Ulaştırma aracına göre yükleme (Üstten yükleme, Yandan Yükleme, Arkadan Yükleme)

#### **17.1.5. Yükleme Alternatiflerine İlişkin Esaslar**

- İlk giren ilk çıkar (FIFO, LIFO) (Yandan yüklemede bir yandan yüklenip diğer yandan boşaltılma yapıldığında)
- İlk giren son çıkar (FILO, LIFO)

#### **17.2. Deniz Yolu Yükleme Planlama Esasları**

##### **17.2.1. Ulaştırma Aracına İlişkin Esaslar**

- Yolcu/yük taşıması yapacak güverte miktarı
- Yolcu/yük güvertelerinin yükleme öncelik sırası (örn.; 1 nci öncelik ana güverte, 2 nci öncelik tank güverte)
- Yolcu/yük güvertesinin adı
- Yolcu/yük güvertesinin depo tipi [(klimalı,kapalı bagaj vb.) (tehlikeli yükler için açık güverte)]
- Yük güvertelerinin araç taşıma kapasiteleri
- Yük taşıma kapasiteleri (ton/lane metre)
- Yolcu taşıma kapasitesi (Yataklı/Pulman/Kampetli/Oturarak/Ayakta)
- Araç ile taşınacak birim yolcu taşıma kapasitesi
- Yük güvertesinin uzunluğu /genişliği/yüksekliği
- Gemiye yüklenecek maksimum kargo uzunluğu/yüksekliği/ genişliği/ ağırlığı
- Kargo bölümü tabanının maksimum yük direnci(kg/m<sup>2</sup>)
- Maksimum istif ağırlığı

- Kendi kendine yükleme boşaltma kapasitesi ve ekipmanları
- Malzeme sabitleme ekipmanlarının konumuna göre gemi bordasından içeriye doğru, yüklemede bırakılacak mesafe
- Yüklemeden sonra meyil (yalpa açısı) ve trim durumu görülerek balast tanklarının ayarlanması.
- Geminin yükleme durumuna göre ne kadar draftta sahip olduğunun görülmesi,
- Yük ve yakıt dengesi durumunun planlamalarda dikkate alınması (gidilecek mesafeye göre yakıt durumunun kontrolü ve yükleme için yakıt doluluk durumu buna göre dikkate alınarak yüklenebilecek maksimum malzeme ağırlık, hacim ve LM hesaplanması)
- Deniz yolu araçlarına yükleme yapılırken güverte ara/yükleme rampası/rampaları olarak kullanılan kapağın yüklenecek malzeme dikkate alındığında taşıyabilme durumunun değerlendirilmesi (kapak/rampa meyil açısı) (gemiye yükleme yapılacak ana rampa/kapak taşıma kapasitesi ve meyil durumunun, yükleme güverteleri arasındaki rampa/asansörlerin taşıma kapasitesi/ebatları)
- Kalkış limanı draftı ile yükleme sonrası gemi draftının karşılaştırılması
- Konteyner yükleme bölmelerinin belirlenmesi (Soğuk iklim konteynerlerinin bağlanacağı yerler (elektrifikasyon için) vb.)
- Güvertelerin doluluk/boşluk miktarı ve boş alanın konumu
- Tehlikeli madde taşıyabilme özelliği (var/yok)

### **17.2.2. Malzeme Yükleneşine İlişkin Esaslar**

- Malzemenin ebatlarının yüklenecek bölmeye uygunluğunun kontrolü [en, boy, yükseklik, ağırlık, taşıma koşulları (İklim Kontrollü Bölme, açık bölme, kapalı bölme)]
- Malzemenin taşınma önceliği (İneceği yere göre),
- Malzemenin yükleme önceliği (tahsis edildiği araca yüklenmede önceliği)
- Malzemenin ikmal sınıfı (Aynı ikmal sınıfına ait malzemeleri taşınacak birimde götürmek için)

- Tehlikeli madde olup olmadığı ve taşıma koşulları
- Yükün cinsi (Dökme, paket, şekilsiz, ekipman, ikmal maddesi)
- Her bir yükün diğerlerinden ne kadar mesafede olması gerektiği bilgisinin değerlendirilmesi (tehlikeli maddeler, büyük ebattaki malzemeler)
- Yüklenecek araçların/malzemelerin arasında bırakılacak mesafe.

### **17.2.3. Malzemenin İstiflenmesine İlişkin Esaslar**

- Malzeme bağımlı üst üste maksimum istifleme (Aynı malzemenin üst üste istif miktarı).
- Malzemenin üstüne konulabilecek maksimum ağırlık miktarı (Farklı tip malzemeler için).
- Yükleme bölmesi üst üste maksimum istifleme yüksekliği ve ağırlığı.
- Tek başına taşınması gereken malzemenin belirlenerek ilgili araca yüklenmesi (örn.dozer)
- Yükler arası boşlukların belirlenmesi
- Dengenin sürekli korunması için aynı bölüme yüklenecek malzemelerin yükleme sırası.
- Suhunet kontrolü yapılacak.

### **17.2.4. Genel Yükleme Esasları**

- Her bir yükün araç içerisinde yerleştirileceği platform, deck vs. üzerindeki yerinin koordinatlarına varıncaya kadar ayrıntılı olarak yükleme şeklinin görülmesi,
- Manuel Yükleme
- Otomatik Yükleme
- Ön Tahsisli Yükleme (Müteakip terminallerdeki yükleme boşaltma faaliyetleri dikkate alınarak gerektiğinde boş alan tahsisi)
- Önceki/Müteakip Terminal(ler)e göre yükleme (Malzeme Biniş/İniş Sırası)
- Liman lokasyon durumu göz önüne alınarak söz konusu limandan kapak atma durumuna göre yükleme yapılıp yapılamayacağını belirlenmesi

### 17.3. Hava Yolu Yükleme Planlama Esasları

#### 17.3.1. Hava Yolu Aracına İlişkin Esaslar

- Kargo bölümünün tipi (Yolcu, Yolcu/yük, Yük vb.)
- Kargo bölümünün yükleme önceliği
- Kargo bölümünün depo tipi
- Kargo bölümünün uzunluğu /geniřlięi/yükseklilięi
- Uçaaęa yüklenebilecek maksimum kargo uzunluęu, geniřlięi, yükseklięi, aęırlılıęı
  - Kargo bölümü tabanının maksimum yük direnci (kg/m<sup>2</sup>), maksimum istif aęırlılıęi(kg), kendi kendine yükleme/bořaltma kapasitesi ve ekipmanların özellikleri
    - Maksimum yük kapasitesi
    - Maksimum istif aęırlılıęı
    - Kendi kendine yükleme /bořaltma kapasitesi
    - Rampa üzerine yüklenebilecek yük kapasitesi (Bu bölüm ayrı bir bölme olarak tasarlanacaktır)
  - Yüklenecek araçların ve malzemelerin aęırlık noktaları hesabı,
  - Malzeme sabitleme ekipmanlarının konumu
  - Belirlenmiř kompartımanlara göre yük ve yakıt dengesi durumunun planlamalarda dikkate alınması.
  - Gidilecek yere göre depolanacak yakıt miktarı dikkate alınarak yükleme yapılabilecek malzeme ebatlarının yeniden deęerlendirilmesi ve kontrolü.
  - Hava yolu araçlarına yükleme yapılırken yükleme rampası olarak kullanılan kapaęın yüklenecek malzeme dikkate alındıęında taşıyabilme durumunun deęerlendirilmesi.
  - Uçak gövdesi içindeki en dar bölüm (çerçeve ebatları)
  - Tehlikeli madde taşıyabilme kontrolü (Sıvı ve katı)

### 17.3.2. Malzeme Yükleneşine İlişkin Esaslar

- Malzemenin ebatlarının yüklenecek bölmeye uygunluğunun kontrolü [en, boy, yükseklik, ağırlık, taşıma koşulları]
- Malzemenin taşıma önceliğı (İneceğı yere göre),
- Malzemenin yükleme önceliğı (tahsis edildiğı araca yüklemede önceliğı)
- Malzemenin ikmal sınıfı (Aynı ikmal sınıfına ait malzemeleri taşıyacak birimde götürmek için)
- Tehlikeli madde olup olmadığı ve taşıma koşulları (bir arada taşınamayacak malzemelerin kontrol edilerek uygun şekilde yükleme yapılması (compatibility analysis))
- Yüknün cinsi (Dökme, paket, şekilsiz, ekipman, ikmal maddesi)
- Her bir yükün diğerlerinden ne kadar mesafede olması gerektiğı bilgisinin değerlendirilmesi (tehlikeli maddeler, büyük ebattaki malzemeler)
- Ekipman (Tekerlekli araç ve benzeri) taşımasında ekipman ağırlıklarının belirlenmiş bölgelerdeki güçlendirilmiş tabanlı bölümlere (treadways) yükleşinin kontrolü
- Uçağın gövdesinden (iç kısım) malzemelerin uzak bulunması gereken minimum mesafe
- Bağlama halkalarının yeri ve özelliklerine göre yükleme yapılması.
- Yüklenecek her bir araç için dingil mesafesi ve ağırlık merkezi hesabı (havadan atılacak malzemeler için değışikse belirtilecek)
- Yüklenecek her bir araç için tekerlek basınçları

### 17.3.3. Malzemenin İstiflenmesine İlişkin Esaslar

- Malzeme bağımlı üst üste maksimum istifleme (Aynı malzemenin üst üste istif miktarı).
- Malzemenin üstüne konulabilecek maksimum ağırlık miktarı (Farklı tip malzemeler için).
- Yükleme bölmesi üst üste maksimum istifleme yüksekliğı ve ağırlığı.

- Tek başına taşınması gereken malzemenin belirlenerek ilgili araca yüklenmesi (örn.dozer)
- Yükler arası boşlukların belirlenmesi
- Dengenin sürekli korunması için aynı bölüme yüklenecek malzemelerin yükleme sırası (her bir malzemenin yüklenmesini müteakip genel araç denge kontrolü yapılması)

#### **17.3.4. Genel Yükleme Esasları**

- Her bir yükün araç içerisinde yerleştirileceği platform, deck vs. üzerindeki yerinin koordinatlarına varıncaya kadar ayrıntılı olarak yükleme şeklinin görülmesi,
  - Manuel Yükleme
  - Otomatik Yükleme
  - Ön Tahsisli Yükleme (Müteakip terminallerdeki yükleme boşaltma faaliyetleri dikkate alınarak gerektiğinde boş alan tahsisi)
  - Müteakip Terminal(ler)e göre yükleme (Malzeme İniş Sırası)
  - Ulaştırma aracına göre yükleme (Yandan Yükleme, Arkadan Yükleme)

#### **17.3.5. Yükleme Alternatiflerine İlişkin Esaslar**

- İlk giren ilk çıkar (FIFO, LIFO) (her iki yönden yüklenip boşaltılabilen uçaklar için)
- İlk giren son çıkar (FILO, LIFO)

### **17.4. Demir yolu Yükleme Planlama Esasları**

#### **17.4.1. Demir Yolu Aracına İlişkin Esaslar**

- Vagon Ağırlığı (Dara)
- Vagon Tipi
- Vagon Maksimum Yük Kapasitesi (Taşıyabileceği Toplam Yüklerin Ağırlığı)
- Yükleme Platformu Yüksekliği (Maksimum/Minimum)
- Yükleme/Boşaltma Süreleri

- Yükleme Yeri (Önden, Arkadan, Yandan vb.)
- Kargo Bölümü Tipi
- Kargo Bölümü Yükleme Önceliği
- Kargo Bölümü Depo Tipi
- Kargo Bölümü Uzunluğu, Geniřliđi, Yüksekliđi
- Yüklenebilecek Maksimum Kargo Ebatları (En, Boy, Yükseklik ve Ađırlık)
- Maksimum İstif Ađırlıđı
- Kargo Bölümü Tabanı Maksimum Dayanım Direnci
- Kendi Kendine Yükleme/Bořaltma Kapasitesi ve Ekipmanları
- Tehlikeli madde Tařıması İin Tehlikeli madde Tařıması Planlanan Vagonların Arasına İlave Edilecek Boř Vagon Miktarı
- Katardaki Vagon Sayısı (Tařınacak malzemelerin özellikleri de dikkate alınarak 60 vagonu geçmeyecek řekilde planlanacaktır)
- Katar Yükl Kapasitesi (Bu Kapasite Katarın Bütün Vagonları Dikkate Alındıđında Planlama Faktörü Olarak Toplam 1000 Tonu Geçemez)
- Yandan kapaklı kargo yük vagonları verileri

#### **17.4.2. Malzeme Yükleneşine İlişkin Esaslar**

- Malzemenin ebatlarının yüklenecek bölmeye uygunluđunun kontrolü [en, boy, yükseklik, ađırlık, tařıma kořulları (İklim Kontrollü Bölme, açık bölme, kapalı bölme)]
- Malzemenin tařınma önceliđi (İneceđi yere göre),
- Malzemenin yükleme önceliđi (tahsis edildiđi araca yükleneş önceliđi)
- Malzemenin sınıfı (Aynı ikmal sınıfına ait malzemeleri tařınacak birimde götürmek için)
- Tehlikeli madde olup olmadıđı ve tařıma kořulları (bir arada tařınamayacak malzemelerin kontrol edilerek uygun řekilde yükleme yapılması (compatibility analysis))

- Ykn cinsi (Dkme, paket, Őekilsiz, ekipman, ikmal maddesi)
- Her bir ykn diđerlerinden ne kadar mesafede olması gerektiđi bilgisinin deđerlendirilmesi (tehlikeli maddeler, byk ebattaki malzemeler)
- İstasyonlardaki yan ve baŐ rampa durumlarının dikkate alınması.

#### **17.4.3. Malzemenin İstiflenmesine İliŐkin Esaslar**

- Malzeme bađımlı st ste maksimum istifleme (Aynı malzemenin st ste istif miktarı).
- Malzemenin stne konulabilecek maksimum ađırlık miktarı (Farklı tip malzemeler iŐin).
- Ykleme blmesi st ste maksimum istifleme yksekliđi ve ađırlıđı.
- Tek baŐına taŐınması gereken malzemenin belirlenerek ilgili araca yklenmesi (rn.dozer)
- Ykler arası boŐlukların belirlenmesi
- Dengenin srekli korunması iŐin aynı blme yklenecek malzemelerin ykleme sırası,

#### **17.4.4. Genel Ykleme Esasları**

- Her bir ykn araŐ iŐerisinde yerleŐtirileceđi platform, gverte (deck) vs. zerindeki yerinin koordinatlarına varıncaya kadar ayrıntılı olarak ykleme Őeklinin grlmesi,
- Manuel Ykleme
- Otomatik Ykleme
- n Tahsisli Ykleme (Mteakip terminallerdeki ykleme boŐaltma faaliyetleri dikkate alınarak gerektiđinde boŐ alan tahsisi)
- Mteakip Terminal(ler)e gre ykleme (Malzeme İniŐ Sırası)
- UlaŐtırma aracına gre ykleme (stten ykleme, Yandan Ykleme, Arkadan Ykleme)

#### **17.4.5. Ykleme Alternatiflerine İliŐkin Esaslar**

- İlk giren ilk Őıkar (FIFO, LIFO) (Her iki ynden yklenebilen/boŐaltılabilen vagonlar iŐin)



- İlk giren son çıkar (FILO, LIFO)

#### **17.5. İç Su yolu Yükleme Planlama Esasları**

Deniz yolu yükleme planlama esasları uygulanacaktır.

#### **17.6. Genel Görüntüleme ve Raporlama Esasları**

Yüklemeyi müteakip malzemenin yükleme sonrası halinin iki boyutlu/üç boyutlu görülmesi sağlanacaktır.

Hangi malzemenin nerede olduğu bilgisinin raporlanabilecektir.

Aynı yere gidecek malzemelerin işaretlenerek görüntülenebilmesi için filtreleme yapılabilecektir.

Her bir yükleme bölgesine ilişkin ayrıntılı yükleme raporu alınabilecektir.

Her bir ulaştırma çeşidi için genel yükleme raporlarının Kullanıcının istediği formatta sistemden alınması sağlanacaktır.

#### **17.7. Manuel Yükleme Planlama Modülü Teknik Özellikleri**

Manuel Yükleme Planlama Modülünün (MYPM) geliştirilmesinde aşağıdaki hususlar dikkate alınacaktır:

Madde 17.9'daki araç tipleri üç boyutlu modellenecektir ve kargo bölümleri gerçeğine uygun bir şekilde tanımlanacaktır. Bu araç tiplerinde kargo kapağı görselleştirilecektir.(3 boyutlu modelde kargo kapağının konumunun doku ile görselleştirilmesi beklenmektedir) Kargo bölümleri ve malzemeler aynı ölçekte olacak ve yakınlaştırma ve uzaklaştırma özelliği bulunacaktır.

Araç tipi önceden belirlenmiş, modellenmiş ve kararlaştırılmış araç tiplerinden biri olabileceği gibi farklı özel bir araç da olabilir.

Özel bir araç verilmiş ise:

- Aracın görsel şekli ve kargo bölümü listesi de verilecektir.
- Bir araç bir veya birden fazla kargo bölümüne sahip olabilecektir.
- Kargo bölümleri yüklenebilir alanı dikkate alacak şekildeki en büyük dikdörtgenler prizması şeklinde olacak oval ve yuvarlak alanların maksimum kullanımı için gerektiğinde aynı yükleme bölgesinde birden çok dikdörtgenler prizması tanımlanabilecektir.

- Kargo bölümleri aracın görsel şekline göre sıralı bir şekilde görünecektir.

- Kullanıcı yükleme yaparken öncelikle yükleme yapmak istediği kargo bölümünü seçecektir.

Modellenmiş araç tiplerinden biri verilmiş ise, kargo bölümü listesi verilmeyecektir. Araç tipi uçak ise ve paletli taşıma yapıyor ise:

- Önce yüklenecek palet sayısı belirlenecek,
- Paletler birleştirilebilecek,
- Yükleme bu paletler üzerine yapılacak,
- Yüklenmiş paletler kargo bölümü tabanında yanlara, ileri ve geri hareket ettirilerek (yatay ve dikey konumlama) yerleştirilebilecek,
- Bazı paletlerin kullanımı engellenebilecek,
- Arkadan yüklenen araçlar için yükleme önden arkaya doğru yapılacak,
- Hem önden hem arkadan yüklenebilen araçlar için yükleme ortadan öne veya ortadan arkaya doğru yapılacak
- Kullanıcı yükleme yaparken öncelikle yükleme yapmak istediği kargo bölümünü seçecektir.

Malzemeler, geliştirilecek ekranda bir tabloda listelenecektir. Tabloda görünecek malzeme alanları aşağıda verilmiştir.

- Kod
- Stok No
- İsim
- Taşınacak Birim Kodu
- Taşımaya esas ebatları (en, boy, yükseklik, ağırlık)
- Malzemenin miktarı (Toplam miktar ve her bir yükleme sonrası kalan miktar şeklinde)

Tablodan seçilen malzemeye ait tüm bilgiler ekran içerisinde başka bir tabloda/alanda gösterilecektir.

Kullanıcı seçtiği malzemeyi kargo bölümünde belirlenmiş kriterlere uygun bir şekilde yerleştirebilecektir. Yerleştirilmiş malzemeler malzeme tablosundan çıkarılır. Yerleştirme kriterleri aşağıda verilmiştir;

Yükleme sırasında üst üste konulan malzemelerin dikey ekseninde birbirlerine yapışık olması sağlanacaktır. Yapışık olarak yüklenmediği takdirde sistem uyarı verecektir.

Malzemelerin asgari dengeli yüklenmesi sağlanmalı ve gerektiğinde Kullanıcı dengesiz yükleme yaptığı konusunda ikaz edilecektir. Malzemeler yerleştirildikleri yerde durabileceklerdir.

Her bir malzemenin yüklenmesi, malzemeyi kapsayan en küçük dikdörtgenler prizması (bounding box) üzerinden planlanır.

Malzeme-malzeme ve malzeme-kargo bölümü iç yüzeyi arasında bırakılması gereken boşluk dikkate alınacaktır. Malzeme üzerindeki ve parametrelerdeki mesafe bilgisinden büyük olan dikkate alınacaktır.

Tehlikeli maddeler birbirlerine parametrelerde tanımlanmış mesafeden daha yakın olmaacaktır.

Yerleştirilmiş malzeme kaldırılabilir. Kaldırılan malzeme tekrar malzeme tablosuna eklenir.

Malzemeler tiplerine göre çizilecektir.

Malzemeler döndürülebilme kısıtlarına [x sabit (y,z değişken), y sabit (x,z değişken), z sabit (x,y değişken)] uygun biçimde 90 ve 180 derece açılarla döndürülebilecektir.

Araç yüklenmiş haldeyken; kargo bölümleri ve araçlar kamera açısı değiştirilerek görüntülenebilir. Ayrıca üstten, yandan, önden gibi hazır kamera açıları da Kullanıcının hizmetine sunulacaktır.

Malzemeler Code alanına göre benzerliklerinin anlaşılması için aynı renge boyanacak, ya da benzeri bir gruplama yapılabilecektir.

Kargo bölümlerinin boş ve yüklü ağırlık merkezleri gösterilecektir.

Modellenmiş araçlar için aracın boş ve yüklü ağırlık merkezleri gösterilecektir.

Malzemelerin, kargo bölümü içindeki koordinatları yükleme sonucunda güncellenecektir.

Malzemelerin ağırlık merkezleri sisteme girilebilecek ve aracın boş ve yüklü ağırlık merkezleri gösterilecektir.

### **17.8. Ara yüzler ve Özellikleri**

Sistem, yüklenmemiş malzemeleri ve verilerini gösteren ara yüze sahip olacaktır.

Hangi kargo bölümüne hangi malzemelerin yüklendiğini gösteren bir grafik ara yüz sunulacaktır.

Yüklemeler şablon olarak kaydedilebilecek, şablon olarak kaydedilmiş bir yükleme daha sonra yeniden aynı şekilde yükleme yapılmasına imkân tanıyacak şekilde kullanılabilir.

Kargo bölümü bazında istenen yükleme adımındaki yükleme durumunun görüntüsü grafik olarak sunulabilecektir.

### **17.9. Modeller**

#### **17.9.1. Modellenen araç tipleri:**

- Hava Araçları
- Yolcu uçakları (Yolcu ve yük bölümleri görülebilecektir)
- Tek katlı kargo uçakları
- Çok katlı kargo uçakları
- Askeri kargo uçakları (C-130 Kargo Uçağı, C-160 Kargo Uçağı, CN-235 Kargo Uçağı (CASA), C-5 Kargo Uçağı, C-17 Kargo Uçağı, A 400M Kargo Uçağı, An 124 Antonov, Ilyushin 76)

#### **17.9.2. Deniz Araçları**

- Ro-Ro Gemisi
- Ro-Pax Gemi (Yolcu ve yük bölümleri görülebilecektir)
- Ro-Ro Gemisi (Üst güvertede konteyner taşıyabilen)
- Kuruyük Gemisi

### 17.9.3. Modelleneyecek "Özel" araç tipleri:

- TIR (bir veya iki römorklu/yarı römorklu)
- Kamyon (2,5 ton, 5 ton, 10 ton, 20 ton, 25 ton)
- Tren (1 lokomotif ve kargo bölümü sayısınca vagon)
- Konteyner (dörde, üçe, ikiye bölünebilir dikdörtgenler prizması 20'lik ve 40'lık)

### 17.9.4. Modelleneyecek Malzemeler:

Aşağıdaki malzemeler modelleneyecektir:

- Kutu (ebatları Kullanıcı tarafından girildiğinde şekli otomatik değişecek)
- Silindir (varil) (ebatları Kullanıcı tarafından girildiğinde şekli otomatik değişecek)
- Kamyon (2,5 ton, 5 ton, 10 ton, 20 ton, 25 ton) (Sabit bölgesi)
- TIR (bir veya iki römorklu/yarı römorklu)
- İş Makinesi 1 (Ekskavator)
- İş Makinesi 2 (Kepçe)
- İş Makinesi 3 (Dozer)
- İş Makinesi 4 (Kreyn)
- Helikopter (Ulaştırma)
- Dorse / Römork
- Konteyner

Modu yukarıda açıklanan araçlar modelleneyecek ve özel araç tipleri ve malzeme tipleri için 3 boyutlu görsel modeller oluşturulacaktır. Bu 3 boyutlu görsel modellerin teknik özellikleri aşağıda belirtilmiştir:

- Her bir 3 boyutlu model en fazla 50.000 noktadan (vertex) oluşacaktır.
- Her bir 3 boyutlu model en fazla 25.000 üçgenden oluşacaktır.

- Her bir 3 boyutlu modele ait en fazla 3 adet 1024x1024 çözünürlükte doku dosyası olabilecektir.
- 3 boyutlu modeller üzerindeki hareketli parçaların animatif gösterimi yapılmayacaktır. (Örnek: helikopterlerin pervanesi dönmeyecektir.)

#### **17.10. Otomatik Yükleme Planlama Modülü Teknik Özellikleri**

Yazılım, manuel yükleme planlama modülü özelliklerini taşıyacak ancak planlamaları araçlara yük tahsisinin tamamlanmasını müteakip otomatik olarak yapacaktır. Bu nedenle araçlara yük tahsis edilirken Kullanıcı tarafından belirlenmiş ön kriterler (minimum %90 doluluk, istif özelliği, vb.) dikkate alınarak sistem tarafından ön bir değerlendirme yapılacaktır.

Yapılan otomatik planlamaların gerektiğinde tamamen veya Kullanıcı tarafından belirlenen kısımlarının iptali ve Kullanıcı tarafından yeniden düzenlenmesi mümkün olacaktır.

#### **17.11. Dağınık Yük Paketleme Sistemi (Breakbulk-Shell/Containerization) Özellikleri**

Dil seçimi yapılabilecek ve istendiğinde değiştirilebilecektir (Çoklu dil seçimini sağlayan yapı oluşturulacaktır).

Ölçü birimleri metrik veya İngiliz ölçü birimi olarak belirtilebilecek gerektiğinde tek işleme dönüştürülebilecektir.

İlk kullanımda kullanıcıya bir sihirbaz yardımı ile programın ana ve anahtar fonksiyonlarına ilişkin gösterim turu yaptırılacaktır.

##### **17.11.1. Kullanım Tasarım Özellikleri**

Malzemeler manuel olarak yüklenebilecek ve bu kapsamda sürükle-bırak (drag&drop) yöntemiyle listeden seçilen bir malzemenin fare kullanılarak üç boyutlu olarak yüklenebilmesi sağlanacaktır. özellikleri fare yardımı ile hareket ettirilerek ve/veya gerektiğinde tüm yükleme tasarımı manuel olarak gerçekleştirilebilecektir.

Yükleme modülü her açıldığında istenildiği ve varsayılan olarak seçildiği takdirde son taşımaya ilişkin yükleme bilgileri ekrana gelecektir.

##### **17.11.2. Kullanıcı Ara Yüzü Özellikleri**

Her ara yüzde kullanımı kolay ve belli mantığa göre yerleştirilmiş sekmeler (tabs) kullanılacak ve bu sayede kullanıcı her aşamada programın neresinde olduğunu anlayabilecektir.

Girilmesi zorunlu alanlar pembe, isteğe bağlı alanlar ise açık mavi renkte gösterilecektir. Zorunlu alanlara veri girilmeden kayıt yapılamayacaktır.

Veriler MS Office programlarına aktarılabilecektir.

Yükleme üstten (Top view) ve açılı (angled view) olacak şekilde iki grafik görünümle kullanıcıya sunulacaktır (sağdan açılı veya soldan açılı ve açı derecesi varsayılan değeri istenildiğinde kullanıcı tarafından değiştirilebilecektir). Bu görünümünün yanında grafik olarak yüklemenin ve hangi malzemelerin hâlihazırda yüklenmediğinin anlık görülebilmesi için yükleme ilerleme durumu bölmesi (stage area) de kullanıcıya sunulacaktır.

Kargo etiketleri “Gelişmiş görüntüler sekmesi kullanılarak görüntülenebilecektir. Bu görüntülemeye Kısım Numarası, Kimlik Kodu ve Tanımı gösterilecektir. Görüntülenen şekil büyütülüp küçültülebilecek ve istenildiğinde herhangi bir detayın görülebilmesi bakımından döndürülebilecektir.

Her hangi bir paketin içinde yer alan aynı cins malzeme miktarı istenildiğinde gösterilecek ve mevcut paket miktarı ile çarpılmış değeri de tablolara yansıtılacaktır.

Herhangi bir paketin üzerine fare ile gelindiğinde paketin özellikleri ve içeriğine ilişkin bilgi otomatik olarak görüntülenebilecektir. Yükleme için bekleyen paketler için de bu özellik sağlanacaktır.

Yükleme sonrası görüntüde görülmesi mümkün olmayan malzemelerin de görüntülenebilmesi bakımından “Adımları Göster” benzeri bir buton yardımıyla (butona her basışta bir sonraki aşama görüntülenecek, istendiğinde de tüm aşamalar görüntülenebilecektir) yüklemenin aşamalandırılması ve aşama aşama kontrolünün yapılabilmesi sağlanacaktır.

Yüklemelerde kullanılacak ebat verilerinde hem nokta hem de virgül kullanılabilir. Aynı husus binler şeklinde ayırmalarda da sağlanacaktır.

### **17.11.3. Herhangi Bir Verinin İthal Edilmesi:**

Ya veri tabanına direkt veri girilerek ya da MS Office (Word, Excel, Powerpoint, Access) programlarından birinde tablo olarak hazırlanmış veriler geçerlik ve uygunluk kontrolü program tarafından yapılmak ve yanlış girilmiş veri bulunmamak (rakamsal veri alanlarında harflerin olması gibi) kaydıyla sisteme bir ara yüz ile aktarılabilecektir.

### **17.11.4. Yükleme Kuralları Seçeneği:**

Bu seçenek istenildiğinde devre dışı bırakılabilecektir.

İstifleme limitleri girilebilecektir.

Bir arada taşınması gerekli veya takım şeklindeki yüklerin (Her bir masa için 4 sandalye gibi) istenildiğinde bir arada taşınması sağlanacaktır.

"Taboo on top" mantığı ile bazı yüklerin üzerine istif yapılması yasaklanabilecektir.

İstenildiğinde bazı malzemelerin sadece tabana yüklenebilmesi (Bottom Only) sağlanacaktır.

Ürünlerin gruplanabilmesi ya da taşıma sırasının belirlenebilmesi bakımından yükleme öncelikleri belirlenebilecektir.

Herhangi bir malzemenin istenildiğinde sadece dik, sadece belirtilen yüzü alta gelecek şekilde yüklenmesi gerektiği gibi bütün alternatifler dikkate alınarak ve bu alternatifler kullanıcıya sunulacak ve gerektiğinde en alta/en üste/en sağa/en sola (herhangi bir uç/dip noktaya) yüklenemeyeceği kullanıcı tarafından belirtilebilecektir.

Yüklenmesi yeterli görülen toplam ağırlık sisteme belirtilebilecek ve belirtilen ağırlığa ulaşıldığında ve/veya yüklemeye müsait başka malzeme kalmadığında sistem otomatik olarak yüklemeyi durduracaktır.

Yükleme şablonlarının kaydedilmesi ve rapor şeklinde sistemden istenilen formatta ve dosyalama şekliyle alınabilmesi sağlanacaktır.

Konteyneri taşıyacak aracın dingil başına düşen ağırlığı ve aracın ağırlık merkezinin yeni yerine ilişkin bilgiler sistem tarafından hesaplanacak ve istenildiğinde kullanıcıya sunulacaktır. Kullanıcı boş yer olduğu takdirde istediğinde bu ağırlık merkezini kaydıracak şekilde yük kaydırması ve yer değiştirmesi yapabilecektir.

İstenildiğinde yükleme belirlenen yükleme kriterleri karşılanarak tamamlanmış olsa bile istenildiği şekilde gerçekleştirilmemiş olabilir. Bunu sağlamak için boş yer olduğu takdirde "yükü tabana yay" benzeri bir düğme kullanılarak konteyner dengesi (stability) sağlanabilecektir.



**EK-7: Yazılım Donanım, İletişim Altyapısı Mimarisi**

# **Yazılım, Donanım, İletişim Altyapısı Mimarisi**

**Versiyon .....**

## 1. GİRİŞ

### 1.1. Amaç ve Kapsam

Sistem; karayolu, denizyolu, havayolu ve iç su yolu ulaştırma ihtiyaçlarının en etkin, süratli ve ekonomik olarak karşılanmasına yönelik ulaştırma planlamalarının Kullanıcı kaynakları ve dış ulaştırma kaynaklarının, araç boş kapasitelerinin ve faaliyetlerinin tüm ulaştırma planlamacıları tarafından karşılıklı olarak zamanında görülerek, paralel ve bütünleşik olarak yapılabilmesi ve karar verici makamlara karar desteği sağlanması ana amacını taşımaktadır.

Bu dokümanla; sistemin donanım, iletişim altyapısı ve uygulama yazılımına yönelik, sistem ihtiyaçları doğrultusunda oluşturulan, genel mimari yapısının özetlenmesi amaçlanmıştır. Bu kapsamda proje genel bir özeti yanında, yazılım geliştirme süreçlerinde kullanılacak geliştirme araç ve teknolojileri özetlenmiştir.

Geliştirici tarafından uygulanan proje yönetim faaliyetleri, MSF/SDD, PMI ve IEEE 12207 standartlarına uygun olarak gerçekleştirilmektedir. Diğer proje yönetimi yaklaşımlarından farklı olarak, uygulanan proje yönetim yaklaşımı Rational, Prince2, UML gibi farklı metodolojilerden de faydalanabilmeyi mümkün kılmakta ve onlarla bütünleşerek CMM, SPICE, ISO 9001:2000 vb. kalite yönetim sistemlerini uygulamayı etkinleştirmektedir. Bu doküman ile, genel olarak, projenin izlenebilmesi, kontrolü ve planlamasının yapılabilmesine yönelik yönetsel bir araç sağlanması hedeflenmiştir. Yazılım Donanım, İletişim Altyapısı Mimari Dokümanı tümüyle, uygulanacak proje yönetim metodolojilerine de uygun olarak, uluslararası kabul görmüş PMI ve IEEE 12207 süreç ve standartlarına uygun olarak hazırlanmıştır.

### 1.2. Proje Özeti

Sistem; Kullanıcının günümüzdeki ve gelecekteki ihtiyaçlarına cevap verecek şekilde:

- Kurye/ring seferlerinin, taşımaların süratle ve maliyet-etkin olarak planlanması,
- Uygulanması ve uygulamaların gerçek ve gerçeğe yakın zamanlı olarak takibi,
- Mevcut ve dış kaynak kullanımı (outsourcing) sayesinde elde edilecek ulaştırma kaynaklarının değerlendirilmesi
- Bu planlama ve değerlendirme faaliyetleri konusunda her seviyedeki karar vericilere karar desteği sağlanması,

- Planlamacıların taşımaya esas bilgilere eş zamanlı olarak ulaşması ile paralel ve bütünlük planlama yapılması imkanı sağlayan müşterek ulaştırma planlama ve koordinasyon sistemi olarak tanımlanabilir.

Sistem; taşımaya esas malzeme bilgileri ve ulaştırma alt yapısının planlanmasına yönelik bütünlük ortak bir veri tabanı (VTYM) ve bu veri tabanını kullanarak, her koşulda taşımaların en maliyet etkin bir şekilde planlanması, programlanması ve koordinasyonuna yönelik ulaştırma ihtiyaçlarının ortaya konması (TPM ve KURPM) ve yapılan planlamaları müteakip işlerliğin ve olası problem alanlarının belirlenebilmesi için gerçek ve gerçeğe yakın zamanlı araç/malzeme iz takibinin yapılabilmesi (ETARİTM) amacıyla geliştirilecek dört ayrı modülden oluşmaktadır.

### **1.2.1. Veri Tabanı Yönetim Modülü (VTYM)**

VTYM, sistemin diğer modülleri (KURPM, TPM, ETARİTM tarafından kullanılacak, Kullanıcı mevcut ve temin edilecek sistemleri ile veri değişimi yapabilecek, veri tabanını yönetecek, işleyecek, değiştirebilecek, güncelleştirebilecek işlevler içeren bir sistem olarak tanımlanmıştır.

VTYM, sistem kapsamındaki diğer modüllerin ihtiyaç duyabileceği taşınacak birim, yolcu, malzeme bilgileri, ulaştırma araç bilgileri, altyapı bilgileri, iz bilgileri, planlı kurye/ ring seferleri bilgileri, plansız ulaştırma aracı tahsis bilgileri, planlı/plansız seferlere planlanmış yolcu/yük bilgileri, KURPM'de kullanılan ulaştırma araç/altyapı bilgileri, taşıma plan bilgilerini içermesi hedeflenen genel bir veritabanıdır.

### **1.2.2. Taşıma Planlama Modülü (TPM)**

TPM, VTYM veri tabanını kullanarak, her koşul altında gerçekleştirilecek taşımaların maliyet etkin bir şekilde planlanabilmesi, programlanabilmesi ve koordinasyonuna yönelik ulaştırma ihtiyaçlarının ortaya konması için geliştirilen bir karar destek sistemi olarak tanımlanabilir. Bu çerçevede, TPM aşağıda listelenen ana hedefleri kapsamaktadır;

- Kullanıcının mevcut ve dış kaynaktan temin edeceği taşıma yeteneklerinin, maliyet etkin, hızlı ve ani taşıma planlamaları ile enuygunlanması (optimization),
- Planlama kapsamındaki her bir taşınacak birimin istenilen zamanda istenilen yerde olmalarının sağlanması,

- Hangi taşınacak birimin, ne kadar sürede, ne kadar uzağa, hangi ulaştırma araçları ile ve hangi ulaştırma altyapısını kullanarak uygun bir planlama süresi içinde taşınabileceğine yönelik planlamaların sağlanması.

### **1.2.3. Kurye/Ring Planlama Modülü (KURPM)**

KURPM, Kullanıcının yurt içi ve yurt dışında süreklilik arz eden taşımalarının maliyet etkin kurye/ring seferleri ile yapılabilmesi amacıyla, yeni kurye/ring seferleri ihdas edilmesinde ve mevcut seferlerinin birleştirilmesi veya uygulamadan kaldırılmasında karar desteği sağlayacak ve ihdas edilmiş kurye/ring seferlerine yönelik taleplerin maliyet-etkin olarak karşılanmasının planlanmasını sağlayacak alt sistem olarak tanımlanabilir.

### **1.2.4. Etkin Araç İz Takip Modülü (ETARİTM)**

ETARİTM, sistem kapsamında gerçekleştirilecek planlamaların işlerliğinin sağlanması ve olası problem alanlarının belirlenebilmesi için gerçek ve gerçeğe yakın zamanlı araç/malzeme iz takibinin yapılabilmesini sağlayacak modüldür.

Proje ihtiyaçları, Kullanıcı ile Geliştirici arasında imzalanan ..... tarih ve ..... Nu.ılı Sözleşme eklerinde yer alan idari ve teknik şartnamelerde belirtilmiştir. İdari ve teknik şartnamelerde sistemin;

- Mevcut ulaştırma yeteneklerinin her koşul altında gerçekleştirilecek taşımaların ve çeşitli maksatlarla görevlendirilen kuryelerin en maliyet etkin şekilde planlanması ve programlanması,

- Anılan optimizasyon sürecinin gerçekleştirilmesine ve karşılaşılan ulaşım taleplerinin yerine getirilmesine yönelik ön değerlendirmelerin yapılabilmesi için gerekli veri setlerini içeren ulaştırma sistemi veri tabanı alt yapısının ve veri tabanının geliştirilmesi,

- Müteakiben planın işlerliğinin ve olası problem alanlarının belirlenebilmesi için etkin araç iz takip planlama sistemlerinin geliştirilmesine yönelik ihtiyaç duyulan sistem, endüstri ve lojistik alanlarındaki mühendislik hizmetleri, danışmanlık, yazılım, donanım, eğitim ve işletim desteği ile ilgili mal ve hizmetlerin tedarikine ilişkin hususlar detaylı bir biçimde açıklanmıştır.

### **1.3. Tanımlar, Kısa Adlar ve Kısaltmalar**

Bu bölüm EK-1'de sunulmuştur.

### **1.4. Referanslar**

- Kullanıcı ile Geliştirici arasında imzalanan ..... tarih ve ..... Nu.ılı Sözleşme
- Sistem Teknik ve İdari Şartnameleri.

- Proje, Yazılım, Donanım, İletişim Altyapısı Mimarisi Sürüm .....,
- IEEE Yazılım Geliştirme Standartları; "Institute of Electrical and Electronics Engineers. IEEE Standard for Software Development Plan, ANSI/IEEE Standard"

## 2. YAZILIM MİMARİSİ

Yazılımı oluşturan uygulamalar nesne tabanlı metodoloji kullanılarak üç katmanlı mimaride geliştirilecektir. Proje kapsamında geliştirilmesi talep edilen tüm yazılımlar grafik ara yüzü, dördüncü kuşak ve nesne (object-oriented) / bileşen (component) tabanlı yapıda olup, kodlama standardı olarak tekrar kullanılabilir (re-usable) yapıda kod teknikleri ile, bu özellikleri birebir destekleyen yazılım geliştirme araçları kullanılarak geliştirilecektir. Proje kapsamındaki uygulama yazılımları İnternet Teknolojileri tabanlı bir yapıda .NET teknolojileri kullanılarak tasarlanacaktır. Detayları aşağıdaki paragraflarda açıklanan yazılım mimarisi modüler, esnek ve ölçeklenebilir bir platform sunacaktır.

### 2.1 .NET ve İnternet Teknolojileri

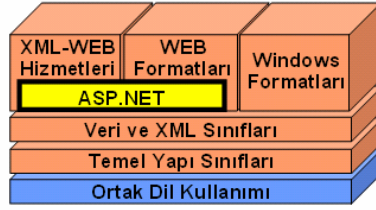
Sistem Microsoft .NET teknolojileri kullanılarak geliştirilecektir. Microsoft .NET teknolojileri günümüzün ve yakın geleceğin en modern kurumsal sistem geliştirme tekniklerini ihtiva etmektedir.

.NET Framework farklı programlama dillerinin ve programlama kütüphanelerinin bir arada uyum içinde çalıştırılmasını sağlayan bir geliştirme ve yürütme ortamıdır. .NET Framework, Windows ve Ağ Uygulamalarını oluşturmak, yönetmek, kurmak, ve diğer ağ tabanlı sistemlerle entegre etmek için, kullanıcıya çok gelişmiş olanaklar sunmaktadır.

.NET Framework aşağıdaki iki önemli bileşenden oluşmaktadır;

- **Ortak Programlama Dili Geliştirme Ortamı (The Common Language Runtime-CLR):** Uygulama yürütümünün yönetilmesini sağlayan servisler sunan programlama dilinden bağımsız geliştirme ve yürütme ortamıdır.

- **Bütünleşik Hazır İşlev Kütüphanesi (The Framework Class Libraries -FCL):** Uygulama geliştirme amaçlı kullanılacak, çok gelişmiş hazır işlevsellikler sunan bütünleşik kütüphanedir.



**Şekil-E7.1:** ASP.NET Yapılanma

.NET Framework, Windows ve Web tabanlı uygulamaların bilgiyi, insanları, sistemleri ve cihazları bağlayarak bütünlük içinde bir sistem oluşturulmasını sağlayan aşağıdaki temel altyapı özelliklerini de beraberinde sunmaktadır;

- **Standard Ağ Protokol ve Özellikleri Desteği:** .NET Framework en geniş kapsamda bilgi, sistem, insan ve cihaz entegrasyonu ve bütünlüğünü sağlamak için TCP/IP, SOAP, XML ve HTTP gibi standard internet protokol ve özelliklerini kullanır.

- **Farklı Programlama Dilleri Desteği:** .NET Framework birçok programlama dilini destekler. Bu sayede, kullanıcılar yaptıkları işe, tecrübelerine ve kullanım alışkanlıklarına en uygun programlama dilini serbestçe seçerek geliştirmelerini yaparlar.

- **Farklı Dillerde Geliştirilmiş Programlama Kütüphaneleri Desteği:** .NET Framework farklı dillerde geliştirilmiş fonksiyon paketlerinin (kütüphaneler) bir arada uyum içinde çalışmasını sağlar. Bu sayede, her fonksiyon için en uygun programlama dili seçilerek hızlı ve kolay geliştirim sağlanır.

- **Farklı Platformlarda Çalışma Desteği:** .NET Framework farklı Windows platformları üzerinde çalışarak farklı uygulama platformlarını kullanan insanları, sistemleri ve cihazları bağlar ve bütünlük sağlar. Örneğin; Windows XP tabanlı PC'ler kullanan kişiler veya Windows CE tabanlı el bilgisayarı kullananlar, Windows Server tabanlı sunumculara bağlanarak uyum içinde çalışabilir.

Proje kapsamında geliştirilecek uygulama yazılımlarında, .NET Framework altyapısı ile ilişkili olarak aşağıda özetlenen teknolojilerden yararlanılacaktır:

- **ADO.NET:** Veritabanı seçiminden bağımsız olarak, OLE DB ve ODBC üzerinden veriye XML tabanlı erişim sunan programlanabilir uygulama arayüzüdür. Oracle ve SQL Server için OLE DB veya ODBC'den çok daha yüksek veritabanı erişimi performansı sağlayan arayüzler de sağlar.

- **ASP.NET (.NET Active Server Pages):** Sunumcu tarafında önceden derlenmiş ve saklı kod mantığını (Code-Behind) ve istemci tarafında Java program

parçacıkları ve dinamik HTML (DHTML) yapısını destekleyerek web-tabanlı uygulamalar ve web servislerinin geliştirilmesini sağlar. .NET platformunun sağladığı Gerektiği Anda Derleme (Just-in time compilation – JIT) ve caching teknolojileri web uygulamalarında kayda değer performans artışı sağlamaktadır. (Bkz. Şekil-E7.1)

- **COM+:** Uygulamalara Dağıtık İşlem (Transaction) Yönetimi, Nesne Havuzu ve Nitelikli Bileşen özellikleri sağlar.
- **MSMQ:** Verinin ve programı tetikleyici olayların güvenilir ve asenkron iletimi için kullanılan mesaj hattı servisedir.
- **XML Web Servisleri:** İstemcilerle açık mimaride, XML tabanlı standartlar ve iletişim protokolleri kullanılarak veri alışverişi sağlayan web-tabanlı kurumsal uygulamalardır. XML Web Servisleri İnternette dağıtık bilgi işlemeye giden yolun temel yapıtaşlarıdır.
- **.NET Remoting:** Nesnelerin farklı uygulama alanları (application domain) üzerinden iletişim ve entegrasyonunu sağlayan iskelet altyapıyı sağlar. Bu altyapı nesnelere aktivasyon ve yaşam boyu destek hizmetlerinin yanı sıra uzak uygulamalara farklı tipte ve özellikte mesaj iletimi yeteneği sağlayan iletişim kanalları sunar.
- **Active Directory:** Bir iletişim ağındaki tüm nesnelere (kullanıcılar, bilgisayarlar, yazıcılar, servisler, vb.) güvenli, yapısal, hiyerarşik veri modeli oluşturulabilmesi için kullanılan bir servistir. Active Directory bu nesnelere saptamak ve bu nesnelere çalışmak için gerekli ortam ve servisleri sağlar.

Sistem Yazılım Altyapısı internet teknolojileri tabanlı olarak tasarlanacaktır. İnternet teknolojileri; sistemi oluşturan donanım ve yazılım unsurlarının, yerel intranet ağı (WLAN) üzerinden veri aktarımını, istenildiğinde sistem yöneticileri tarafından uzaktan erişimle yönetilebilmesini, web hizmetleri tabanlı çalıştırılabilmesini ve tüm uygulamaların hizmet bazlı çok katmanlı yapıda geliştirilerek merkezî, yerel ve dağıtık olarak ölçeklenebilmesini sağlayacak her türlü yazılım, donanım ve protokoller olarak tanımlanabilir. Başka bir ifade ile İnternet teknolojileri; web servisleri (.NET, XML, SOAP, WSDL, UDDI vb.) bazlı geliştirilen çok katmanlı, zenginleştirilmiş internet uygulamalarını (Rich İnternet Applications) içeren ölçeklenebilir gelişmiş yazılım teknolojileridir.

Sistem kapsamında öngörülen .NET platformunun en önemli avantajlarından birisi de, bu altyapıda birbiriye tutarlı, uyumlu, bütüncül yapı ve mekanizmaların “tek bir firma” tarafından sağlanabilmesi ve böylelikle yazılım yaşam döngüsü süresince Java ortamlarının aksine, çok sayıda firma tarafından temin edilen birbirinden

tümüyle farklı ürünler arasında yaşanabilecek uyumluluk ve birlikte çalışabilirlik sorunlarının hiç bir suretle yaşanmamasıdır. Farklı firmaların farklı ürünleri ile meydana getirilebilecek bir Java ortamının, projenin gerçekleştirilmesi ve sonrasındaki destek ve idame sürecinde ciddi uyum, entegrasyon ve birlikte çalışabilirlik problemleri doğurması; öte yandan tüm bileşenleri tek bir firma tarafından ve birbiriyle tümüyle uyumlu, entegre ve birlikte çalışabilir olarak sağlanan .NET ortamında bu sorunların yaşanmaması .NET geliştirme ortamının seçilmesinde çok etkili bir rol oynamıştır. .NET platformunun sadece bu özelliğinin dahi, bu projede olduğu gibi, kullanıcı/saha/uygulama sayıları, süreçler, kaynaklar, roller, vb. kritik özellikleriyle son derece kapsamlı (kullanıcı, site sayısı, uygulama sayısı, entegre işleyiş, süreçler, kaynaklar, roller) ve büyük ölçekli bir projenin sürekliliği ve idamesi açısından çok önemli bir avantaj ve verimlilik sağlayacaktır. Böylelikle, Kullanıcıya yüksek performanslı, genişlemeye açık ve yapılan teknolojik ve insan gücü yatırımlarının gelecekte de korunacağı bir sistem sağlanması hedeflenmiştir.

## **2.2. Nesne Tabanlı Analiz ve Tasarım**

Proje kapsamında uygulanacak yazılım geliştirme süreçlerinin her aşamasında, MSF/SDD ve IEEE 12207 standartlarına da uygun olarak, nesne tabanlı yazılım analiz ve tasarım teknikleri kullanılacaktır. Yazılımı oluşturan tüm bileşenlerin geliştiriminde kalıtım (inheritance), şablonlama (templates), keşeleme (encapsulation) gibi temel nesneye yönelik programlama konseptleri ağırlıklı kullanılacaktır.

Proje kapsamındaki uygulama yazılımları, her biri kendi içerisinde bütünlük arz eden özel işlevleri sağlayacak şekilde, modüler ve hizmet (service) tabanlı olarak tasarlanacaktır.

Nesneye yönelik geliştirme metodolojisi modüler tasarımla birleştiğinde yazılım tasarım biçimine katkıda bulunmakta ve modüllerin tekrar kullanımına imkân sağlamaktadır.

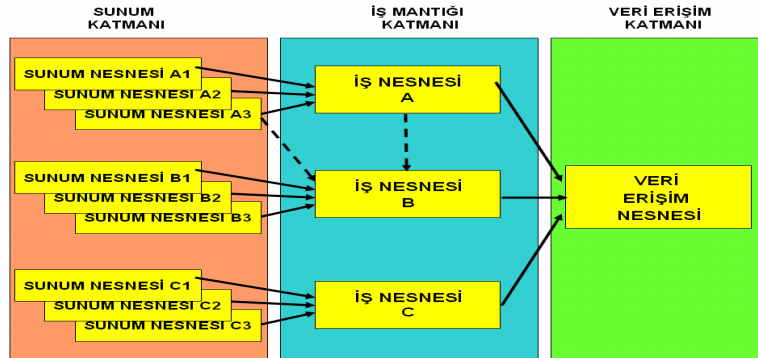
Nesnelerin özellik ve davranışları istenildiğinde sistem bütünlüğünü bozmayacak şekilde değiştirilebilir. Yazılımın temel unsurları sınıf (class) olarak tanımlanmaktadır. Her sınıf bünyesinde özellikleri (attributes) ve davranışları (methods) belirleyen kriter ve metotlar tanımlanır. Bu tanımlama nesne tabanlı yazılımın en önemli özelliklerinden biri olan keşeleme (encapsulation) sağlamakta, bileşenlerin kendi içlerinde birbiriyle uyumlu işlevsel bir bütün olarak geliştirilmesini (consistency and unity), tekrar kullanılmasını (reusability), ve hataların önlenmesini sağlamaktadır. Bu sayede yazılım içerisinde yaratılacak her bir nesnenin tanımı,



üyesi olduğu sınıf ve o sınıfın üzerinde olan sınıflardan kalıtım (inheritance) yolu ile sahiplendiği özelliklerden oluşabilecektir. Bu nitelikte bir yapı oluşturulabilmesi için, tasarım aşamasında oldukça iyi bir planlama yapılması ve yine bu süreçte çok kapsamlı bir ileri yazılım mühendisliği çalışması gerekmektedir. Ancak “iyi tasarım”, gerçekleştirme sırasında hata olasılığının önemli derecede azalmasını, test verimliliğinin ve gerçekçiliğinin artırılmasını ve yazılım uygulamaya geçtikten sonraki idame aşamasında da ihtiyaç duyulabilecek düzeltme ve iyileştirmelerin klasik bir yazılıma göre son derece hızlı (ortalama 10 kat) yapılabilmesini sağlamaktadır. Bununla birlikte yazılıma yapılan müdahale, genellikle sadece ilgili sınıfın tanım ve metotlarına yapıldığı için yazılım bütünlük ve işleyişini klasik sistemlerin aksine bozmamaktadır. Yazılımda kullanılacak nesne tabanlı yazılım geliştirme metodolojisi modüler tasarım ile birleştiğinde yazılım tasarım dokularını (design patterns) oluşturabilmekte ve modüllerin tekrar ve çok amaçlı kullanımını sağlamaktadır.

### 2.3 N-Katmanlı Mimari (Şekil-E7.2)

Sistem yazılım modeli 3 katmanlı mimariye uygun olarak tasarlanmıştır. Kullanıcı Ara yüzü, İş Mantığı ve Veri Tabanı işlemleri farklı katmanlarda aşağıda özetlendiği şekliyle gerçekleştirilecektir.



Şekil-E7.2: N Katmanlı Mimari

#### 2.3.1 Katman 1: Sunum Katmanı (Presentation Layer):

Bu katman tüm kullanıcı ara yüzünü içerir. Bu katmanın sağladığı grafiksel ara yüzler (GUI) aracılığı ile kullanıcının program ile bire bir etkileşimi sağlanır, kullanıcı veri girişini yapar ve sonuçları izler. Sunum katmanı, aynı zamanda, istemcinin sunumcudan aldığı istek sonuçlarını işler ve kullanıcıya sunmak üzere formatlar. Bir web servisinde bu katman kullanıcı isteklerinin işlenmesi ve bu isteklerin sonuçların sunulmasını sağlar.

#### 2.3.2 Katman 2: İş Mantığı Katmanı (Business Logic Layer):

Kullanıcı ara yüzü ve veri katmanı arasında uygulamanın ana mantığının gerçekleştirildiği katman bulunur. İş mantığı, uygulamadaki her tür işlem için kuralları

içerir ve birinci katmandaki kullanıcıyı üçüncü katmandaki veriye bağlar. Bu katmandaki işlevler belirlenen ihtiyaçları ve özellikleri iş mantığı kuralları çerçevesinde gerçekleyen bir veya birden fazla görevden oluşur. Bu katmanda geliştirilen kod aynı zamanda kapsamlı güvenlik mekanizmalarını ihtiva eder. Çünkü, kullanıcı girdisi güvenilir sınırlardan geçmektedir.

#### **2.3.4 Katman 3. Veri Hizmetleri (Data Services):**

Veri hizmetleri yapısal (SQL, Oracle veritabanı, XML veritabanı) veya yapısal olmayan (MS Exchange, MS Message Queuing) veri tabanına erişim ve veri yönetimini içerir. Bir uygulama bir ya da daha çok veri tabanına erişebilir.

Üç katmanlı mimari bahsi geçen temel işlevleri birbirinden ayırır. Bu sayede, sunum katmanı (kullanıcı ara yüzü) kuralların işlenmesi ve iş mantığından bağımsızdır. Aynı şekilde ilk ve ikinci katmanlar veriye erişim biçimi ile hiç bir şekilde ilgilenmezler.

Bu model, başlangıçta daha detaylı ve kapsamlı analizi gerektirmesine rağmen bakım maliyetlerini kayda değer derecede düşürür ve uzun vadede işlevsel esnekliği ciddi derecede artırır.

Proje kapsamındaki uygulama yazılımları, üç katmanlı mimaride geliştirilecektir. Bu sayede, uygulama ve iş mantığı tümüyle orta katmanda gerçekleştirilecek, böylece bütünlük ve tutarlılık elde edilmesinin yanı sıra, grafiksel ara yüzün bağımsızlığı da sağlanabilecektir. Veri Hizmetleri katmanı tüm veritabanı işlemlerini gerçekleştirecek, iş mantığının veritabanından bağımsız olmasını sağlayacaktır.

#### **2.4 Sistem Yazılım Mimarisi**

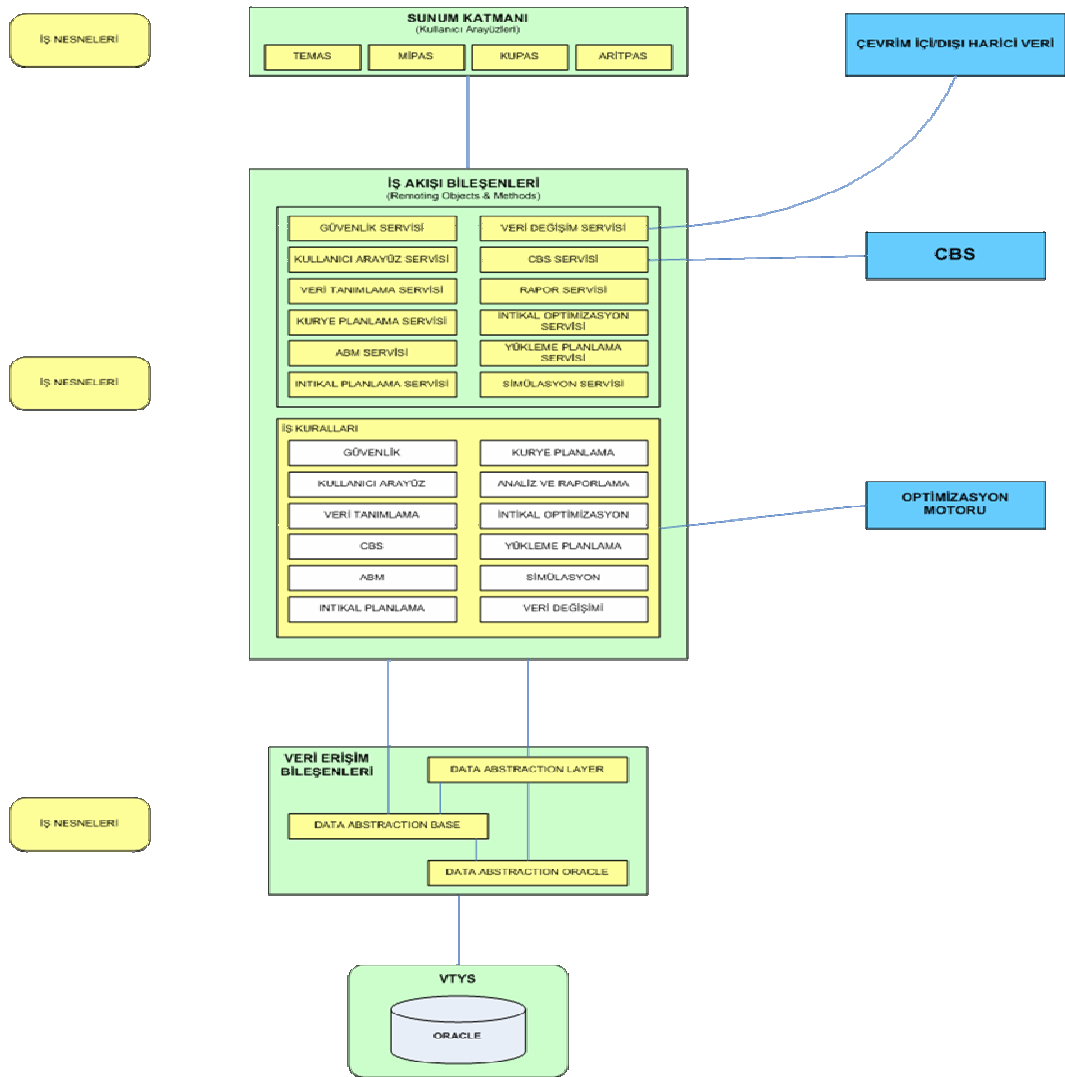
Geliştirilecek uygulama yazılımı üç katmandan oluşacaktır (Şekil-E7.3).

Ön katmanda (sunum katmanı / kullanıcı ara yüzü) iş mantığından bağımsız olan grafik kullanıcı ara yüzü uygulamaları bulunacaktır. Bu katmanda ince, kalın ve akıllı istemci (thin, thick veya smart client) yapısında tasarlanmış istemci uygulamaları yer alacak ve tüm sistem uygulamaları için işlenmiş veriyi kullanıcılar ve sistem yöneticilerine en uygun yapıda sunacaktır.

Orta katman, tüm iş mantığı uygulamaları yanında tüm güvenlik hizmetlerini, iletişim hizmetlerini, ağ hizmetlerini ve diğer sistemlerle entegrasyon amacıyla oluşturulan modülleri de ihtiva edecektir. Orta katmandan veriye erişim için üst seviye veri erişim fonksiyonları kullanılacaktır.

Arka katmanda ise veritabanı hizmetleri yer alacaktır. Veritabanı işlevlerinin ayrı bir katmanda yer alması veriye erişimi iş mantığından ayırarak, hata ayıklamayı ve bakımı kolaylaştıracak ve uygulamayı veri tabanından bağımsız hale getirecektir.

Sunum katmanında VTYM, TPM, KURPM ve ETARİTM Sistemlerinin kullanıcı ara yüzleri ve sistemin dış dünyaya açık olan servis ara yüzleri bulunacaktır. Sistemin diğer sistemlerle entegrasyonunu sağlayan web hizmetleri ve işlemlere bu katman üzerinden erişilecektir.



**Şekil-E7.3:** Uygulama Yazılımı Planlanan Katmanları

Orta katmanda İş Akışı Bileşenleri ve İş Kuralları yer alacak ve aşağıdaki hizmetler sunulacaktır:

- **Güvenlik Hizmetleri:** Kullanıcı, grup, yetki, rol tanımlama, kimlik doğrulama, kaynağa yetki / rol bazlı erişim, iletişim güvenliği ve değişiklik yapma (auditing) mekanizmalarının yönetimi ve kontrolünü sağlayacaktır.
- **Veri Değişim Hizmetleri:** Sistem ile diğer sistemler arası veri değişimlerini yönetecek, verilerin doğrulaması, eşleştirilmesi, dönüştürülmesi ve ortak kullanımını sağlayacaktır.
- **Kullanıcı Ara yüzü Hizmetleri:** Kullanıcı ara yüzünden gelen isteklerin ilgili modüllere / hizmetlere yönlendirilmesini ve sistem modül ve hizmetlerinden kullanıcı ara yüzüne verilerin iletilmesini kontrol eder. Kullanıcı ara yüzü seçeneklerinin iş mantığını etkileyebilecek özellikleri (verilerin istenen bir formatta filtre edilerek sunulması gibi) bu hizmet tarafından kontrol edilecektir. Kullanıcı ara yüzünde yer alacak dinamik elemanlar bu hizmetlerin kazandıracığı işlevler olacaktır.
- **CBS Hizmetleri:** Coğrafi Bilgi Sistemi ile çift yönlü veri iletişimini sağlayacak ve sisteme entegrasyon kapsamındaki tüm iş mantığı ve ulaşım ağı altyapı bilgilerinin tanımlanması / güncellenmesi / yönetilmesi bu servis tarafından gerçekleştirilecektir.
- **Veri Tanımlama Hizmetleri:** Sistemde yer alan verilerin kullanıcılar, on-line, off-line mekanizmalar tarafından tanımlanmasını sağlayacak ve tanımlanan verilerin doğrulaması, nitelendirilmesi ve saklanması işlemlerini yürütecektir.
- **Raporlama Hizmetleri:** Sistemdeki verilerin iş mantığına uygun olarak değerlendirilmesi, birleştirilmesi ve kullanıcıya karar desteği şeklinde uygun formatta sunulması işlemlerini yürütecektir.
- **Kurye Planlama Hizmetleri:** Sistemde yer alan tüm kurye, malzeme, yol, araç verilerinin değerlendirilmesi ve planlanması işlemlerini yürütecektir.
- **Taşıma Optimizasyon Hizmetleri:** İntikal işlemlerinde tüm parametreleri değerlendirerek en uygun alternatifi sunmak üzere tasarlanan algoritmaların çalıştırılacağı hizmet(ler)tir.
- **ABM Servisi:** Araç, (Taşınacak) Birim, Malzeme bilgilerinin doğrulaması, nitelendirilmesi ve saklanması işlemlerini yürütecektir.
- **Yükleme Planlama Hizmetleri:** Taşıma ve kurye planlama işlemlerinde yer alan yükleme planlamalarının gerekli parametreleri değerlendirerek en uygun alternatifin sunulması amacıyla tasarlanan algoritmaların çalıştırılacağı hizmetlerdir. Kullanıcı bu hizmetler sayesinde istediğinde otomatik yükleme

planlamasını görebilecek, değiştirebilecek veya tamamen manuel yükleme yapabilecektir.

- **Taşıma Planlama Hizmetleri:** Sistemde yer alan tüm taşıma planlama işlemlerinin, malzeme, yol, araç verileri ile birlikte değerlendirilmesi ve planlanması işlemlerini yürütecektir.

- **Benzetim (Simülasyon) Hizmetleri:** Taşıma planlama benzetimi sonucu ortaya çıkan verilerin kullanıcı ekranında alternatifler şeklinde gösterilmesini ve yapılan planlamanın doğruluğunun görülmesini sağlayacaktır.

Yukarıda açıklanan orta katmandaki bu hizmetler, Sistemin kural tabanında tanımlanan iş kurallarını kullanacaktır. Kullanılan kural-tabanlı yazılım sayesinde kullanıcı yönetmeliklerinde yapılacak değişiklikler çoğu zaman koda müdahale edilmeden ve sistemin çalışmasını kesintiye uğratmadan sisteme yansıtılacaktır. Sistemin kural tabanında aşağıdaki başlıklarda iş kuralları tanımlanacaktır:

- Güvenlik
- Kurye Planlama
- Kullanıcı Ara yüzü
- Analiz ve Raporlama
- Veri Tanımlama
- Taşıma Benzetimi (Simülasyonu)
- CBS
- Yükleme Planlama
- Araç, (Taşınacak) Birim, Malzeme (ABM)
- Plan Benzetimi (Simülasyonu)
- Taşıma Planlama
- Veri Değişimi

Arka katmanda veri erişim bileşenleri yer alacak ve veriye erişim çeşitli seviyelerde ele alınarak, iş mantığı veritabanından ve veri erişim mekanizmalarından bağımsız hale getirilecektir.

### **3. SİSTEM OPTİMİZASYONU MİMARİSİ**

Sistemde saklanan verinin kullanılabilir bilgi haline getirilmesi ve hedeflenen en uygun çıkarımların elde edilebilmesi için sistemin bütünü göz önüne alabilecek optimizasyon modellerine sahip bir karar destek sistemine ihtiyaç duyulmaktadır.

Optimizasyon, en genel anlatımı ile matematiksel bir fonksiyonun belli kısıtlar çerçevesinde alabildiği en düşük veya en yüksek değeri bulma sürecidir. Son

dönemde, ilerleyen bilişim teknolojileri optimizasyon sistemlerini giderek sık kullanılan bir karar destek aracı haline getirmiştir.

Optimizasyon modellerinin kullanımına ve çözüm sınırlarına dair verilebilecek bir örnek, en iyi bilinen optimizasyon problemlerinden biri olan Gezgin Satıcı Problemi'dir (Traveling Salesman Problem). Özetle "en kısa mesafeyi katedecek şekilde  $n$  şehri ziyaret edip başlangıç noktasına geri dönmek" şeklinde tanımlayabileceğimiz problemin yaklaşık 25,000 şehirlik bir çalışmasını Applegate, Bixby, Chvátal, Cook, ve Helsgaun 2004 yılı itibarı ile çözmeyi başarmışlardır. Söz konusu problemin uygulama modelini ele alan projeler arasında Amerikan Ulusal Sağlık Enstitüsü'ndeki Gen Sıralama Problemi, NASA'da gerçekleştirilen Uzay Araçları İçin Yakıt Kullanımı Azaltımı Projesi, Bell İletişim Araştırma'da gerçekleştirilen fiber optik ağ tasarımı projeleri de bulunmaktadır. Projenin konusu olan lojistik ve taşıma süreçleri konusunda ise 2003 yılında Amerika'nın Wisconsin eyaletinde Schneider Logistics isimli bir şirkette yapılan bir optimizasyon çalışmasında taşıma maliyetlerinde %5-25'lik düşüşler gözlemlenmiştir. Bu çalışma sonucunda firmanın günlük maliyet azalması yaklaşık 3000\$ olarak hesaplanmıştır.

Taşıma planlamasında optimizasyon tekniklerinin kullanımı ile ilgili ilk çalışmalar 1954 yılında Dantzig ve Fulkerson'un tanker zaman çizelgelemesi ile ilgili çalışmalarına kadar uzanmaktadır. Buna rağmen, her türlü taşıma planlaması ve bağlantılı filo genişleme kararları ile ilgili optimizasyon temelli nispeten az çalışma bulunmaktadır. Özellikle son 20 yılda bu alandaki çalışmaların miktarı artmışsa da, çalışmalar çoğunlukla benzetim (simulation) temellidir. Optimizasyon temelli çalışmalar hava kuvvetlerindeki uygulamalar ile sınırlı kalmıştır.

Problemin boyutuyla beraber veri gereksinimleri de büyümektedir. Veri tabanındaki ham veri parçaları, uygun sorgularla işlenerek kullanılabilir bilgi haline getirilip optimizasyon model(ler)ine girdi olacaktır. Kurulması amaçlanan optimizasyon modelinin asgari olarak aşağıdaki girdileri kullanması gerekmektedir:

- Ulaşım altyapısı kapasite kısıtları
- Yapılan plana dâhil olan taşınacak birim mevcut konuş yerleri
- Yapılan plana dâhil olan taşınacak birimlerin ulaşması istenen son noktalar
- Yapılan plana dâhil olan taşınacak birimlerin en erken taşınmaya başlayabilecekleri zaman
- Yapılan plana dâhil olan taşınacak birimlerin son noktada bulunmaları gereken en erken zaman

- Yapılan plana dâhil olan taşınacak birimlerin son noktada bulunmaları gereken en geç zaman

- Bekleme yerlerinde bulunan ulaştırma araçları cins ve sayıları
- Araçların yükleme ve boşaltma zamanları
- Araçların taşıma kapasiteleri
- Araçların ulaşım altyapısı ile uyumluluğu
- Araçların taşınacak malzemeler ile uyumluluğu
- Araçların taşıma süreleri
- Araçların kiralama/işletme maliyetleri

Optimizasyon modelinin belirlemesi gereken kararlar aşağıda belirtilmiştir.

- Her bir taşınacak birimin hangi ulaşım noktasında hangi araçlara yüklendiği

- Her bir taşınacak birim bileşeninin hangi ulaşım noktasında hangi araçlardan indirildiği

- Her bir aracın izleyeceği rota

- Her bir aracın yükleme/boşaltma yaptığı ulaşım noktalarına varış zamanı

- Her bir aracın yükleme/boşaltma yaptığı ulaşım noktalarından ayrılış zamanı

- Yukarıda belirtilen kararlar verilirken uyulması gereken olası kısıtlar aşağıda listelenmiştir.

- Kullanılacak ulaştırma araçlarının sayısının aşılmaması,

- Kullanılacak ulaştırma araçlarının kapasitelerinin aşılmaması,

- Kullanılacak ulaştırma araçlarının uyumlu olmadığı malzemeleri taşıma işine atanmaması

- Kullanılacak ulaştırma araçlarının uyumlu olmadığı güzergahlara atanmaması,

- Taşınacak birimlerin en erken taşınmaya başlayabilecekleri tarihten önce taşınmaya başlanamamaları

- Taşınacak birimlerin en erken varış tarihinden önce gidecekleri hedef noktaya ulaşmamaları

- Taşınacak birimlerin en geç varış tarihinden sonra sefer/plan görev yerlerine ulaşmamaları

- Ulaştırma araçları arasında malzeme transferi yapılan noktalarda zaman çakışması yaşanmaması

Model, kullanıcı tarafından yeni kısıtlar girilebilecek ve söz konusu kısıtların hangilerinin plan dâhilinde geçerli olacağına kullanıcının karar vereceği şekilde tasarlanacaktır.

Geliştirilecek model için iki ayrı olası amaç fonksiyonu belirlenmiştir.

- En az maliyet
- En kısa sürede ulaşımın tamamlanması

Bahsedilen hedeflere ulaşacak bir model kurulması yönündeki çalışmalarda, ilk etapta doğrusal tamsayısal programlama metotlarının uygulanabilirliği incelenecektir. Bir fikir vermesi açısından ön çalışma olarak kurulan bir model aşağıda verilmiştir. Model çok modlu ulaşım olasılığı göz önünde bulundurularak kurulmuştur. Ulaştırma modu  $m$  indeksi ile ifade edilmiştir. Modelde ulaşım modunun 0 olması, taşınacak birime ait taşımaya esas malzemenin yola çıkmış olduğu fakat hiçbir araca yüklenmemiş olduğu durumu ifade etmek için kullanılmıştır. Anlatım kolaylığı açısından değişken indeksleri tekrarlanan (redundant) bilgi taşımaktadır. Örneğin, bir ulaşım arki başlangıç noktası, bitiş noktası, ve ulaşım modu ile tanımlı iken, bazı değişkenlerde her üç özellik de indeks olarak kullanılmaktadır.

#### Parametreler:

|              |  |
|--------------|--|
| $s_u$        | : $u$ biriminin mevcut konuş lokasyonu                             |
| $t_u$        | : $u$ biriminin taşınacağı hedef lokasyon                          |
| $e_u$        | : $u$ biriminin yola çıkmaya hazır olabileceği en erken zaman      |
| $a_u$        | : $u$ biriminin istenen en erken varış zamanı                      |
| $b_u$        | : $u$ biriminin istenen en geç varış zamanı                        |
| $A_v$        | : $v$ tipi araçların uyumlu oldukları ulaşım ağı arklarının kümesi |
| $N_v$        | : $v$ tipi araçların uyumlu oldukları lokasyonların kümesi         |
| $N'_m$       | : Ulaşım modu $m$ ile uyumlu olan lokasyonların kümesi             |
| $C_i$        | : $i$ lokasyonunun uyumlu olduğu ulaşım modları kümesi             |
| $Begin_m(i)$ | : $i$ lokasyonundan çıkan arklar (ulaştırma modu $m$ 'de)          |
| $Final_m(i)$ | : $i$ lokasyonunda sonlanan arklar (ulaştırma modu $m$ 'de)        |
| $Asset_m$    | : Ulaşım modu $m$ olan ulaşım araçları kümesi                      |
| $TptAsset_m$ | : Ulaşım modu $m$ 'de taşınabilen ulaşım araçları kümesi           |
| $VCap_v$     | : $v$ tipi araçların hacim taşıma kapasitesi                       |
| $WCap_v$     | : $v$ tipi araçların ağırlık taşıma kapasitesi                     |



$PCap_v$  :  $v$  tipi araçların yolcu taşıma kapasitesi  
 $mode_v$  :  $v$  tipi araçların ulaşım modu  
 $Weight_u$  :  $u$  biriminin taşımaya esas malzemelerinin toplam kütlesi  
 $Volume_u$  :  $u$  biriminin taşımaya esas malzemelerinin toplam hacmi  
 $Pax_u$  :  $u$  biriminin toplam yolcularının sayısı

$Weight_v$  :  $v$  tipi bir aracın ağırlığı  
 $Volume_v$  :  $v$  tipi bir aracın hacmi

$t_{ij}^m$  :  $(i,j)$  arkının ulaşım modu  $m$  kullanılarak katedilme süresi  
 $\tilde{t}_{iu}^m$  :  $u$  biriminin  $i$  lokasyonunda  $m$  tipi ulaşım moduna yüklenme süresi  
 $\hat{t}_{iu}^m$  :  $u$  biriminin  $i$  lokasyonunda  $m$  tipi ulaşım modundan boşaltılma süresi

$AssetCap(m,(i,j))$ : Aynı anda  $(i,j)$  arkını ulaştırma modu  $m$ 'de kullanabilecek maksimum ulaştırma aracı sayısı.

$tc_{ijv}^1$  :  $v$  tipi bir aracın dolu olarak  $(i,j)$  arkından geçme maliyeti  
 $tc_{ijv}^2$  :  $v$  tipi bir aracın boş olarak  $(i,j)$  arkından geçme maliyeti

### **Karar değişkenleri:**

$y_{iut}^m$  : Eğer  $u$  Birimi  $t$  anından  $t+1$  anına kadar  $i$  lokasyonunda ulaştırma modu  $m$ 'de bekliyorsa 1, diğer durumlarda 0

$\tilde{y}_{iut}^m$  : Eğer  $u$  Birimi  $t$  anında  $i$  lokasyonunda ulaştırma modu  $m$ 'ye yüklenmeye başlanıyorsa 1, diğer durumlarda 0

$\hat{y}_{iut}^m$  : Eğer  $u$  Birimi  $t$  anında  $i$  lokasyonunda ulaştırma modu  $m$ 'den boşaltılmaya başlanıyorsa 1, diğer durumlarda 0

$y_{iut}^m$  şu indeksler için tanımlıdır  $(i,u,t,m) : e_u \leq t \leq b_u$

$\tilde{y}_{iut}^m, \hat{y}_{iut}^m$  şu indeksler için tanımlıdır  $(i,u,t,m) : e_u \leq t \leq b_u, m \neq 0$

$z_{ijut}^m$  : Eğer  $u$  Birimi  $t$  anında  $(i,j)$  arkında ulaştırma modu  $m$ 'de harekete geçiyorsa 1, diğer durumlarda 0.

$z_{ijut}^m$  şu indeksler için tanımlıdır  $((i,j),u,t,m) : e_u \leq t \leq b_u$   
 $i \in N_m, j \in N_m, m \neq 0$

$x_{iuv,t}$  :  $u$  Birimine tahsis edilmiş halde  $t$  anından  $t+1$  anına kadar  $i$  lokasyonunda bekleyen  $v$  tipi araç sayısı

$x_{iuv,t}$  şu indeksler için tanımlıdır  $(i,u,v,t) : i \in N_v, e_u \leq t \leq b_u$

$\hat{x}_{iv,t}$  :  $i$  lokasyonunda  $t$  anından  $t+1$  anına kadar bekleyen  $v$  tipi boş araç sayısı

$\hat{x}_{iv,t}$  şu indeksler için tanımlıdır  $(i,v,t) : i \in N_v$

$s_{ijuv,t}^m$  :  $u$  Birimine tahsis edilmiş olarak  $t$  anında  $(i,j)$  arkında ulaştırma modu  $m$ 'de harekete geçen  $v$  tipi araç sayısı

$\hat{s}_{ijv,t}$  :  $t$  anında  $(i,j)$  arkında harekete geçen  $v$  tipi boş araç sayısı

$s_{ijuv,t}^m$  şu indeksler için tanımlıdır  $((i,j),u,v,t,m) : v \in Comp_m, (i,j) \in A_v, e_u \leq t \leq b_u$

$\hat{s}_{ijv,t}$  şu indeksler için tanımlıdır  $(i,j,v,t) : (i,j) \in A_v$

$\tilde{w}_{iuv,t}$  :  $u$  birimine tahsis edilmiş olarak  $i$  lokasyonunda  $t$  anında yüklenmeye başlayan  $v$  tipi araç sayısı

$\hat{w}_{iuv,t}$  :  $u$  birimine tahsis edilmiş olarak  $i$  lokasyonunda  $t$  anında boşaltılmaya başlayan  $v$  tipi araç sayısı

$\tilde{w}_{iuv,t}, \hat{w}_{iuv,t}$  şu indeksler için tanımlıdır  $(i,u,v,t) : i \in N_v, e_u \leq t \leq b_u$

### **Model:**

$$\min \sum_{\substack{(i,j),m, \\ u,v,t}} tc_{ijv}^1 s_{ijuv,t}^m + \sum_{\substack{(i,j),m, \\ v,t}} tc_{ijv}^2 \hat{s}_{ijv,t} \quad (1)$$

s.t.

$$\sum_{\substack{m_1 \in C_i \\ m_1 \neq 0}} \hat{y}_{iu,t-\hat{t}_{iu}^{m_1}}^{m_1} + y_{iu,t-1}^0 = y_{iu,t}^0 + \sum_{\substack{m_2 \in C_i \\ m_2 \neq 0}} \tilde{y}_{iu,t}^{m_2}, \forall i,u,t \quad (2)$$

$$\sum_{j \in B_m(i)} z_{jii,t-t_{ji}^m}^m + \tilde{y}_{iu,t-\hat{t}_{iu}^m}^m + y_{iu,t-1}^m = \quad (3)$$

$$y_{iu,t}^m + \hat{y}_{iu,t}^m + \sum_{k \in F_m(i)} z_{iku,t}^m, \forall i,u,t,m : m \in C_i, m \neq 0$$

$$\sum_{j \in B_m(i)} \hat{s}_{jiv,t-t_{ji}^m} + \sum_u \hat{w}_{iuv,t-\hat{t}_{iu}^{\text{mod } e_v}} + \hat{x}_{iv,t-1} = \quad (4)$$

$$\hat{x}_{iv,t} + \sum_u \tilde{w}_{iuv,t} + \sum_{k \in F_m(i)} \hat{s}_{ikv,t}, \forall i,v,t : i \in N_v$$

$$\sum_{\substack{m, j: \\ v \in \text{Comp}_m \\ j \in B_m(i)}} s_{juv, t-t_{ji}^m}^m + \tilde{W}_{iuv, t-\tilde{t}_{iu}^{\text{mod } \epsilon_v}} + x_{iuv, t-1} =$$

$$x_{iuv, t} + \hat{w}_{iuv, t} + \sum_{\substack{m, k: \\ v \in \text{Comp}_m \\ k \in F_m(i)}} s_{ikuv, t}^m \forall i, u, v, t : i \in N_v \quad (5)$$

$$\sum_{v \in \text{Comp}_m} \text{Weight}_v s_{ijuvt}^m + \text{Weight}_u z_{iju, t}^m \leq \sum_{v \in \text{Carrier}_m} \text{WCap}_v s_{ijuvt}^m, \forall (i, j), u, t, m : m \neq 0 \quad (6)$$

$$\sum_{v \in \text{Comp}_m} \text{Volume}_v s_{ijuvt}^m + \text{Volume}_u z_{iju, t}^m \leq \sum_{v \in \text{Carrier}_m} \text{VCap}_v s_{ijuvt}^m, \forall (i, j), u, t, m : m \neq 0 \quad (7)$$

$$\text{Pax}_u z_{iju, t}^m \leq \sum_{v \in \text{Carrier}_m} \text{PCap}_v s_{ijuvt}^m, \forall (i, j), u, t, m : m \neq 0 \quad (8)$$

$$\text{Weight}_u \hat{y}_{iu, t}^m \leq \sum_{v \in \text{Carrier}_m} \text{WCap}_v \hat{w}_{iuv, t}, \forall i, u, t, m : m \neq 0 \quad (9)$$

$$\text{Volume}_u \hat{y}_{iu, t}^m \leq \sum_{v \in \text{Carrier}_m} \text{VCap}_v \hat{w}_{iuv, t}, \forall i, u, t, m : m \neq 0 \quad (10)$$

$$\text{Pax}_u \hat{y}_{iu, t}^m \leq \sum_{v \in \text{Carrier}_m} \text{PCap}_v \hat{w}_{iuv, t}, \forall i, u, t, m : m \neq 0 \quad (11)$$

$$\text{Weight}_u \tilde{y}_{iu, t}^m \leq \sum_{v \in \text{Carrier}_m} \text{WCap}_v \tilde{w}_{iuv, t}, \forall i, u, t, m : m \neq 0 \quad (12)$$

$$\text{Volume}_u \tilde{y}_{iu, t}^m \leq \sum_{v \in \text{Carrier}_m} \text{VCap}_v \tilde{w}_{iuv, t}, \forall i, u, t, m : m \neq 0 \quad (13)$$

$$\text{Pax}_u \tilde{y}_{iu, t}^m \leq \sum_{v \in \text{Carrier}_m} \text{PCap}_v \tilde{w}_{iuv, t}, \forall i, u, t, m : m \neq 0 \quad (14)$$

$$\text{Weight}_u y_{iu, t}^m \leq \sum_{v \in \text{Carrier}_m} \text{WCap}_v x_{iuv, t}, \forall i, u, t, m : m \neq 0 \quad (15)$$

$$\text{Volume}_u y_{iu, t}^m \leq \sum_{v \in \text{Carrier}_m} \text{VCap}_v x_{iuv, t}, \forall i, u, t, m : m \neq 0 \quad (16)$$

$$\text{Pax}_u y_{iu, t}^m \leq \sum_{v \in \text{Carrier}_m} \text{PCap}_v x_{iuv, t}, \forall i, u, t, m : m \neq 0 \quad (17)$$

$$y_{s_u u, e_u - 1}^0 = 1, \forall u \quad (18)$$

$$y_{t_u u, b_u}^0 = 1, \forall u \quad (19)$$

$$\hat{x}_{iv0} = \text{Fleet}(i, v), \forall i, v \quad (20)$$

$$x, \hat{x}, s, \hat{s}, \hat{w}, \tilde{w} \in Z^+$$

$$y, \hat{y}, \tilde{y}, z \in B$$

Yukarıda verilen modelin optimum çözümünün bulunması, problem boyutları büyüdükçe zorlaşacaktır. Buna rağmen, Yöneylem Araştırmasında sıkça rastlanan ve modelin kalitesini düşüren “büyük M”ler içermediğinden, çözüm sürecinin daha başlarında yüksek kalitede çözümlerin üretilebilmesi ihtimali vardır. Proje süresince yapılacak deneysel çalışmalar modelin doğruluğu ve üretilecek çözümlerin kalitesinin geliştirilmesini sağlayacaktır.

#### **4. SİSTEM VERİ TABANI VE DONANIM MİMARİSİ**

Sistem üç ayrı sunumcu üzerinde yer alan üç ayrı veritabanı sisteminden oluşacaktır

##### **4.1. Sistem Sunumcusu Üzerinde Bulunan Veri Tabanı Sistemi:**

Veri Tabanı Yönetim Modülü (VTYM), Taşıma Planlama Modülü (TPM), Kurye/Ring Planlama Modülü (KURPM), alt sistemlerinin bağlanacağı ORACLE veri tabanıdır. (Sözleşme doğrultusunda bu veritabanı Kullanıcıya ait sunumcu üzerindeki ORACLE VTYS üzerinde çalışacaktır)

##### **4.2. Etkin Araç İz Takip Modülü (ETARİTM) Sunumcusu Üzerinde Bulunan Veritabanı Sistemi:**

Etkin Araç İz Takip Modülü (ETARİTM) tarafından kullanılması planlanan ilişkisel bir veritabanı sisteminden oluşacaktır.

ETARİTM veritabanı ve Sistem Sunumcusu üzerinde bulunan veritabanı arasında on-line veri transferi olacak şekilde planlanacaktır (Güvenlik ön plana çıktığında internet ve GSM üzerinden alınacak verilerin sisteme zarar vermesinin önüne geçilmesi için gerektiğinde off-line kullanım da planlanacaktır). Bu sayede ETARİTM sunumcusu üzerinde takip edilecek olan aracın KURPM ve TPM modüllerinde yük/yolcu taşınması planlanan araçlardan birisi olması durumunda; Kullanıcı tarafından belirlenecek KURPM ve TPM bilgileri (taşınacak yük/yolcu, aracın tipi, sürücü/mürettebat kimlik bilgileri) ETARİTM sunumcusuna taşınabilecek ve izlenen araçlara ait ihtiyaç duyulan bilgilere erişim otomatik olarak sağlanabilecektir.

##### **4.3. Coğrafi Bilgi Sistemi (CBS) Sunumcusu Üzerinde Bulunan Veritabanı Sistemi:**

Sistemin bütün modülleri tarafından kullanılacak harita ve coğrafi ulaştırma alt yapı verileri bu veri tabanından alınacak verilerle sağlanacaktır. Kullanıcı tarafından temin edilecek bir paket program (MapInfo gibi) ve bu programın kullandığı harita veri setleri sistemle bütünleştirilmek (entegrasyon) üzere en kısa

sürede temin edilerek Geliştiriciye verilecektir. Ana Sistem ile karşılıklı veri transferini sağlayacak ara birim CBS'nin Geliştiriciye teslimini müteakip hazırlanacaktır.

Veritabanının yedeklenmesi ORACLE'ın yedekleme araçları kullanılarak hem on-line hem de off-line olacak şekilde tasarlanacaktır.

Kullanıcının hâlihazırda sahip olduğu sunumcular kullanılmaya devam edecektir.

#### **4.4. ETARİTM Sunumcusu ve Çevre Birimleri**

Kullanıcı tarafından temin edilecek ETARİTM Sunumcusu üzerinde bulunacak bir tanesi gömülü diğer ikisi kart olarak takılmış 3 adet LAN kartı aracılığı ile Kullanıcı yerel ağına dâhil edilecektir. ETARİTM sunumcu üzerinde bulunan 3 adet LAN kartı aynı yapıda olup iki adedi team olacak şekilde yapılandırılacaktır.

ETARİTM sunumcusu üzerinde ..... adet Hotswap Sabit disk yuvası bulunacak, sunumcunun mimari yapısı RAID 5 ve tüm RAID türevlerinin kurulmasına izin verecek ve üzerinde bulunan iki adet ..... GB büyüklüğündeki UW SCSI yapısındaki Sabit Disk RAID 1 olarak yapılandırılacaktır.

#### **4.5. Aktif İz Takip Kiti I (Dizüstü Bilgisayar, GPS Cihazı, GSM Telefon)**

Kullanıcı tarafından temin edilecek Aktif İz Takip Sistemi-I üç ayrı bölümden meydana gelecektir. GPS cihazı diz üstü bilgisayarın Seri Portuna ve Cep telefonu diz üstü bilgisayarın USB girişine bağlanacaktır.

#### **4.6. Aktif İz Takip Kiti II (GSM ve 3G Özellikli Cep Bilgisayarı, GPS Cihazı)**

Kullanıcı tarafından temin edilecek Aktif İz Takip Sistemi-II üç ayrı bölümden oluşacak, fakat cihazlar arasındaki bağlantı herhangi bir iletişim kablosu ile değil, Cep Bilgisayarının üzerinde bulunan soketler ile ve yekpare bir görünüm elde edilecektir.

Pocket PC aynı zamanda GSM ve 3G özelliğine sahip olacaktır. Cep Bilgisayarının arka kapağı çıkarılarak BackPack adı verilen cihaz takılacaktır. Bu cihaz aynı zamanda üzerinde 1 adet CF Slot ve VGA slotu da bulunduracaktır. BackPack cihaza ilave batarya imkanı kazandırmakta ve cihazın çalışma ömrünü oldukça uzatmaktadır. BackPack sayesinde Cep Bilgisayarına kazandırılacak CF slota ise CF arabirimine sahip olan GPS cihazı takılacaktır. Bu sistem de Dizüstü bilgisayar ile oluşturulacak sistem gibi çalışacak ve görevini yerine getirecektir.

#### **4.7. Sistem Kullanıcı Terminalleri (İstemciler), Yazıcıları ve Router ve Modemleri**

Kullanıcının elinde mevcut sistemler kullanılmaya devam edecek, ihtiyaç duyulduğunda ilave donanımlar Kullanıcı tarafından temin edilecektir. Barkod iletişim

sistemleri [Barkodlama (Binary ve Şerit) aparatları, barkod okuyucular ve yazıcılar, RFID özellikli etiketleme ve okuma sistemleri] kullanıcı tarafından temin edilecektir. Barkod Cihazlarına Barkod yazıcılar bağlanabilecek ve etiketler basılabilecektir.

## **5. SİSTEM İLETİŞİM ALTYAPISI MİMARİSİ**

### **5.1. Sistem Sunumcusu ile İstemciler Arasındaki İletişim Alt Yapısı**

Sistem sunumcusu hâlihazırda Kullanıcı tarafından kullanılmakta intranet/güvenli web ile sağlanmaktadır.

### **5.2. Aktif İz Takip Sistemi**

Aktif İz Takip sistemi iki farklı donanım altyapısıyla hizmet verecek şekilde tasarlanmıştır. Bu donanımlardan birincisi diz üstü bilgisayarı ile, diğeri de cep bilgisayarı ile tasarlanmıştır.

#### **5.2.1. Dizüstü Bilgisayar Sistemi:**

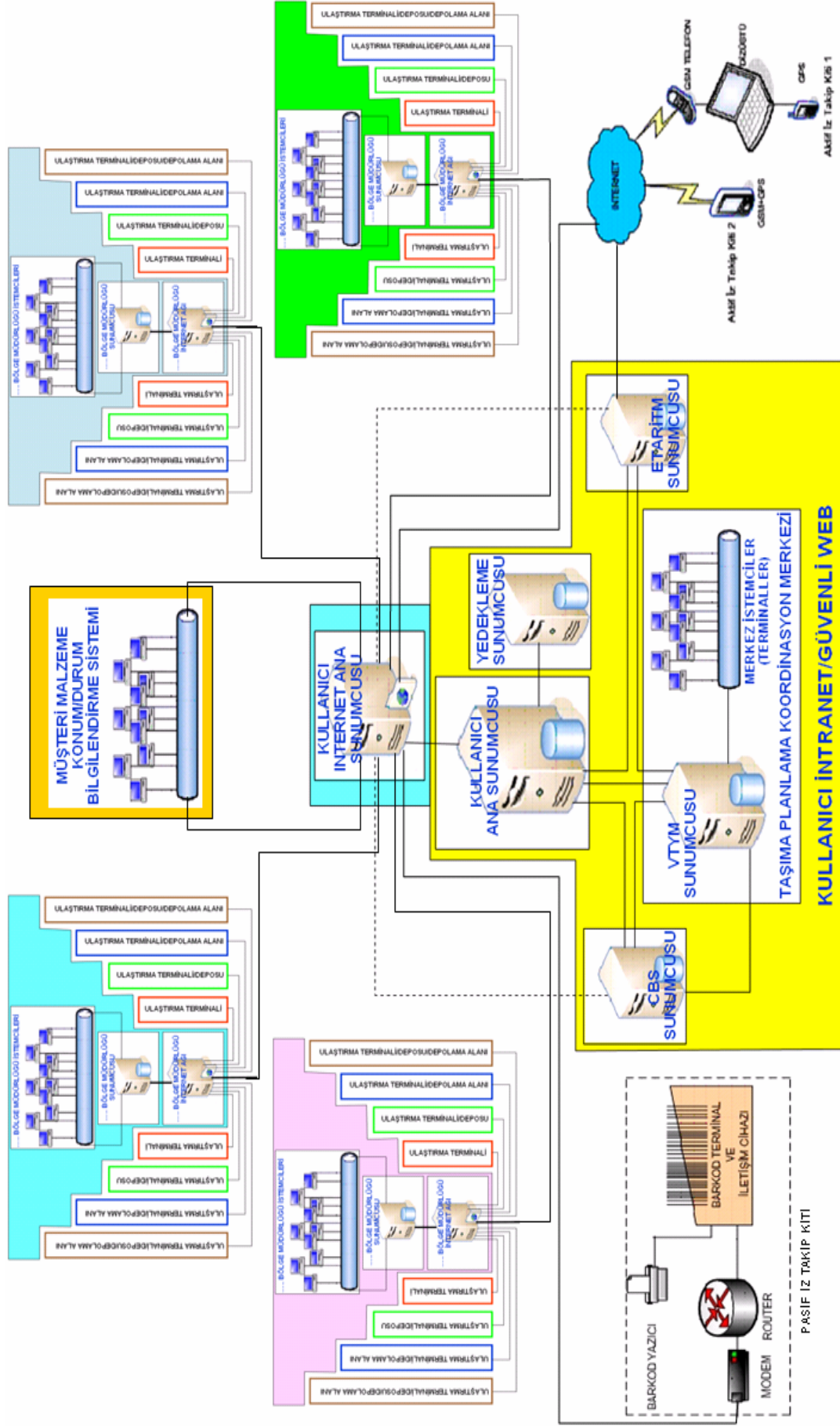
Bu sistem; bir dizüstü bilgisayarı, bu bilgisayara bağlanabilen GPRS özelliği bulunan bir cep telefonu ve bir GPS cihazından oluşmaktadır. Uydudan gelen GPS verileri GPS cihazı ile belirli bir iletişim portu üzerinden dizüstü bilgisayara aktarılmakta, dizüstü bilgisayar üzerinde bulunan yazılım ile okunmakta, gerekli olan veriler ayrıştırıldıktan sonra belirlenen bir formatta GSM operatörünün GPRS altyapısı kullanılarak ETARİTM sunumcusuna iletilecektir.

#### **5.2.2. Cep Telefonu Sistemi:**

Bu sistem; GSM-GPRS ve GPS sistemlerinin tümleşik olarak bünyesinde bulunduran bir cep bilgisayarından oluşmaktadır. Uydudan gelen GPS verileri cep bilgisayarı üzerinde bulunan yazılım ile okunup, ayrıştırıldıktan sonra belirlenen bir formatta GSM operatörünün GPRS altyapısı kullanılarak ETARİTM sunumcusuna iletilecektir.

### **5.2.3. ETARİTM Sunumcusu ile Sistem Ana Sunumcusu Veri İletişimi Alt Yapısı**

Sistem Ana Sunumcusu, Kullanıcı intranet/güvenli web iletişim ağı içerisinde yer alacaktır. ETARİTM sunumcusu ise internete bağlı olup iki sunumcu arasında fiziksel bağlantı kontrollü olarak sağlanacaktır. ETARİTM sunumcusu ile Sistem Ana Sunumcusu arasındaki veri iletişimi gerektiğinde dijital taşıma ortamları (CD/DVD/BRD/USB Yığın Depolama Aygıtları) kullanılarak gerçekleştirilecek şekilde de tasarlanacaktır.



Şekil-E7.4 Sistem Veri Tabanı, Donanım ve İletişim Altyapısı Mimarisi

## 6. SONUÇ

Sistem, icra edilecek her türlü taşımanın maliyet etkinliği, zaman, malzeme, araç ve personel tasarrufu sağlayacak şekilde faaliyetlerin tüm planlamacılar tarafından karşılıklı olarak zamanında görülerek paralel ve bütünleşik şekilde yapılabilmesi ve karar verici makamlara karar desteği sağlanması ana amacını taşımaktadır.

Bu doküman, proje kapsamında gerçekleştirilecek sistemin donanım, iletişim altyapısı ve uygulama yazılımına yönelik, sistem ihtiyaçları doğrultusunda oluşturulan, genel mimari yapısını özetlemektedir. Bu kapsamda projenin genel bir özeti sunulmuş, yazılım geliştirme süreçlerinde kullanılacak geliştirme araç ve teknolojileri özetlenmiş, bu süreç içerisinde teslim edilen/edilecek donanım ve iletişim altyapısına yönelik teknik özellikler listesi ile planlanan iletişim topolojileri sunulmuştur.

Sistem Mimari tasarımı aşaması, uygulanacak proje yönetim metodolojisinin çok önemli bir aşamasını oluşturmaktadır. Geliştirici proje yönetim metodolojisi olarak bu alanda dünyanın en gelişmiş metodoloji/yaklaşımlarından biri olan MSF/SDD metodolojisini uygulayacaktır. Uygulanacak MSF/SDD metodolojisi, faz yaklaşımli (phased approach) ve iterative-spiral (aşamalı-döngüsel) bir geliştirme süreci öngörmektedir. Bu çerçevede, proje yönetim sürecine dair tüm aşamalar döngüsel olarak, nihai ürün teslim edilene kadar devam edecektir. Bu yaklaşımın temel amacı; yazılımı aşamalı/döngüsel olarak geliştirmek ve erken sürümlerle birlikte müşteri ve dolayısı ile son Kullanıcının görüşlerini alarak, eski sürümlerin geliştirilmesi aşamasındaki alınan ders ve tecrübelerden de yararlanarak, sisteme yönelik yeni sürümler gerçekleştirebilmektir. Bu yaklaşımda teslim edilen ara sürümler için bir onay mekanizması /onay aşaması yoktur, sadece Kullanıcıdan gelen geri bildirim ve görüşler yeni sürümlere yansıtılmaktadır. Bu döngüsel ve aşamalı geliştirme süreci, nihai ürün teslim edilene kadar bu şekilde devam edecektir. Bu çerçevede, tasarım aşaması da dâhil tüm süreçlerin proje süresince yaşayan faaliyetler olduğu ve her aşamada müşteri/son kullanıcı-proje ekibi arasında etkileşimli bir ekip çalışmasına ihtiyaç olduğu unutulmamalıdır.

Uygulanacak proje yönetim metodolojisi, projeye Müşteri/Kullanıcı müdahale ve katkısını maksimum düzeyde sağlama amacını taşımaktadır. Bir projenin başarısı, ancak, o projeye dâhil olan tüm proje ortaklarının (müşteri, firma ve üçüncü parti kurum ve organizasyonlar) proje amaçları doğrultusunda, önceden belirlenen



rol ve sorumlulukları çerçevesinde projeye katılmaları ile mümkün olabilir. Başarılı projeler, keskin sınırlarla tanımlanmış müşteri-firma ilişkilerinden ziyade, takım çalışması ve proje ortaklığı oluşturabilecek bir yapının kurulması ile gerçekleştirilebilir.

## **EK-8: Genetik Algoritmalar ve Geliştirilen Sezgisel (Heuristic) Model**

### **GENETİK ALGORİTMALAR VE GELİŞTİRİLEN SEZGİSEL (HEURISTIC) MODEL**

#### **1. GENEL**

Günümüzün karmaşık ve zor koşulları problemlere hızlı ve kolay çözüm veren yeni çözüm yöntemleri arayışına neden olmuştur. Özellikle sert(hard) optimizasyon teknikleri yerine, yumuşak hesaplama (soft computing) ve evrimsel algoritma (evolutionary algorithm) kullanımı ön plana çıkmıştır. Evrimsel yaklaşımlardan olan genetik algoritmalar da, bu arayışlar içinde önemli bir yer tutmaya başlamıştır. Uygulama başarıları artan ve sürekli geliştirmeye çalışılan genetik algoritmalar diğer yumuşak hesaplama yöntemleri ile birlikte kullanılarak hibrid (hybrid) çözümler geliştirilmesine çalışılmaktadır.

Genetik algoritmalar, doğal seçim ilkelerine dayanan bir arama ve optimizasyon yöntemidir. Temel ilkeleri John Holland tarafından ortaya atılmıştır. Temel ilkelerinin ortaya atılmasından sonra, genetik algoritmalar hakkında birçok bilimsel çalışma yayınlanmıştır. Ayrıca, genetik algoritmaların teorik kısmı ve uygulamaları hakkında birçok uluslararası konferans da düzenlenmektedir. Genetik algoritmaların, fonksiyon optimizasyonu, çizelgeleme, mekanik öğrenme, tasarım, hücresel üretim gibi alanlarda başarılı uygulamaları bulunmaktadır. Geleneksel optimizasyon yöntemlerine göre farklılıkları olan genetik algoritmalar, parametre kümesini değil kodlanmış biçimlerini kullanırlar. Olasılık kurallarına göre çalışan genetik algoritmalar, yalnızca amaç fonksiyonuna gereksinim duyar. Çözüm uzayının tamamını değil belirli bir kısmını tararlar. Böylece, etkin arama yaparak çok daha kısa bir sürede çözüme ulaşırlar (Goldberg, 1989: 1-7). Diğer bir önemli üstünlükleri ise çözümlerden oluşan popülasyonu eş zamanlı incelemeleri ve böylelikle yerel en iyi çözümlere takılmamalarıdır.

“Reeves; ulusal hükümetler ve organizasyonlar tarafından genetik algoritma tabanlı projelere, tavlama benzetimi (simulated annealing) ve yasaklı arama (tabu search) tabanlı projelere göre daha fazla kaynak ayrılmakta olduğunu belirtmektedir” (Yeniay, 2001: 37). Bu ifadeden anlaşılacağı üzere, genetik algoritmaların geleneksel optimizasyon yöntemlerine olduğu gibi sözü edilen yapay zeka yöntemlerine göre de çeşitli alanlarda üstünlükleri bulunmaktadır. Bu üstünlükler genetik algoritmaların arama yapısı ile ilgilidir. Genetik algoritmaların arama yapısı ise, alt diziler teoremi ve yapı blokları hipoteziyle açıklanmaktadır.

Başlangıç çözümünden bağımsız olma, paralel çözüm arama ve hızlı çalışma özelliklerinden dolayı genetik algoritmalar çoğu yönetime göre, problemlere daha uygun çözüm üretebilen yöntemler olarak kabul görmektedir. Sözü edilen problemlerde genetik algoritmaların kullanımında parametrelerinin belirlenmesi gibi güçlükler de söz konusudur. Her problem çeşidinin parametreleri farklı olduğu için, bu parametreler genellikle deney tasarımı yöntemiyle belirlenmektedir. Ayrıca genetik algoritmalar kısıt içeren optimizasyon problemlerinin çözümünde de sorunlarla karşılaşabilmektedirler.

Rekabetin gittikçe arttığı günümüzün müşteri odaklı pazarlarında üretim işletmelerinin yaşamlarını sürdürebilmesi için öncelikle iki temel işletme fonksiyonunu en etkin şekilde yerine getirmesi gerekmektedir. Bu fonksiyonlar, üretim ve pazarlama fonksiyonlarıdır. Bu fonksiyonlara ait problemlerinin çözümleri için yoğun çalışmalar yapılmaktadır. Bu çerçevedeki genetik algoritma çalışmalarının, üretim ve tasarım alanlarında oldukça yaygın uygulama alanları mevcuttur. Pazarlama alanında ise üretim ve tasarım alanlarındaki kadar yaygın bir uygulaması görülmemektedir.

Genetik algoritmaların üretim problemlerinde, özellikle çizelgeleme, hücrenel üretim ve tasarım alanlarında geleneksel yöntemlere göre daha iyi sonuçlar verdiği görülmektedir. Bu tip işletme problemlerinin kısa sürede en uyguna (optimum) yakın olarak çözülmesi, üretim planlama ve tasarımda etkin çalışmayı sağlamaktadır. Zamanın maliyet olduğu günümüz pazar şartlarında üretim planlama, tasarım gibi kritik bölümlerin etkin çalışması, işletmenin müşterilerine daha çabuk yanıt vermesi demektir. Bu da önemli bir rekabet avantajı sağlamaktadır.

Pazarlama fonksiyonu açısından konuya bakıldığında ise, tüketici profillerinin ve satın alma kalıplarının bilinmesi işletme açısından son derece kritik bir bilgidir. Bu bilgiyi elde etmeye yönelik bir takım teknikler mevcuttur. Bunların başında, veri yığınları içerisinde belirli kalıpları çıkarmaya yarayan veri madenciliği tekniği gelmektedir. Genetik algoritmalar ise veri madenciliğinde arama amaçlı olarak kullanılmaktadır. Veri madenciliği uygulamalarında, genetik algoritmalar çözüm uzayının yalnızca belirli bir kısmını taraması nedeniyle diğer yöntemlere göre daha hızlı sonuçlar vermektedir. Bu nedenle de genetik algoritmaların veri madenciliğinde bir arama yöntemi olarak, özellikle diğer yumuşak hesaplama yöntemleriyle birlikte çalışarak önemli bir uygulama yeri bulacağı beklenebilir.

## 2. BASİT GENETİK ALGORİTMA

Bir çok alanda uygulama imkanı ve uygulamaları olan genetik algoritmaların işleme adımları şöyle açıklanabilir (Engin,2001:21):

- Arama uzayındaki tüm mümkün çözümler dizi olarak kodlanır.
- Genellikle rastsal bir çözüm kümesi seçilir ve başlangıç popülasyonu olarak kabul edilir.
- Her bir dizi için bir uygunluk değeri hesaplanır, bulunan uygunluk değerleri dizilerin çözüm kalitesini gösterir.
- Bir grup dizi belirli bir olasılık değerine göre rastsal olarak seçilip çoğalma işlemi gerçekleştirilir.
- Yeni bireylerin uygunluk değerleri hesaplanarak, çaprazlama ve mutasyon işlemlerine tabi tutulur.
- Önceden belirlenen kuşak sayısı boyunca yukarıdaki işlemler devam ettirilir.
- İterasyon, belirlenen kuşak sayısına ulaşıncaya işlem sona erdirilir. Amaç fonksiyonuna göre en uygun olan dizi seçilir.

Genetik algoritmalar bir çözüm uzayındaki her noktayı, kromozom adı verilen ikili bit dizisi ile kodlar. Her noktanın bir uygunluk değeri vardır. Tek bir nokta yerine, genetik algoritmalar bir popülasyon olarak noktalar kümesini muhafaza eder. Her kuşakta, genetik algoritma, çaprazlama ve mutasyon gibi genetik operatörleri kullanarak yeni bir popülasyon oluşturur. Birkaç kuşak sonunda, popülasyon daha iyi uygunluk değerine sahip üyeleri içerir. Bu, Darwin'in rassal mutasyona ve doğal seçime dayanan evrim modellerine benzemektedir. Genetik algoritmalar, çözümlerin kodlanmasını, uygunlukların hesaplanmasını, çoğalma, çaprazlama ve mutasyon operatörlerinin uygulanmasını içerir (Jang, 1997: 175-176). (Bknz. Şekil-E..-1)

### 2.1. Çözümlerin Kodlanması

Bir problemin çözümü için genetik algoritma geliştirmenin ilk adımı, tüm çözümlerin aynı boyutlara sahip bitler dizisi biçiminde gösterilmesidir. Dizilerden her biri, problemin olası çözümler uzayındaki rastsal bir noktayı simgeler (Yeniay, 2001: 38). Parametrelerin kodlanması, probleme özgü bilgilerin genetik algoritmanın kullanacağı şekle dönüştürülmesini sağlar (Jang, 1997: 176).

## 2.2. İlk Populasyonun Oluřturulması

Olası çözümlerin kodlandıđı bir çözüm grubu oluşturulur. Çözüm grubu populasyon, çözümlerin kodları da kromozom olarak adlandırılır. İkili alfabenin kullanıldıđı kromozomların gösteriminde, ilk populasyonun oluşturulması için rassal sayı üreticileri kullanılabilir. Rassal sayı üreticisi çağrılır ve deđer 0,5'den küçükse konum 0'a deđilse 1 deđerine ayarlanır (Yeo ve Agyel, 1998: 269). Birey sayısının ve kromozom uzunluđunun az olduđu problemlerde yazı-tura ile de konum deđerleri belirlenebilmektedir. Genetik algoritmalarda ikili kodlama yöntemi dışında, çözümlü aranan probleme bađlı olarak farklı kodlama yöntemleri de kullanılmaktadır (Goldberg, 1989: 9).

## 2.3. Uygunluk Deđerinin Hesaplanması

Bir kuřak oluşturulduktan sonraki ilk adım, populyondaki her üyenin uygunluk deđerini hesaplama adımıdır. Örneđin, bir maksimizasyon problemi için i. üyenin uygunluk deđerini  $f(i)$ , genellikle o noktadaki amaç fonksiyonunun deđeridir (Jang, 1997: 176). Çözümlü aranan her problem için bir uygunluk fonksiyonu mevcuttur. Verilen belirli bir kromozom için uygunluk fonksiyonu, o kromozomun temsil ettiđi çözümün kullanımıyla veya yeteneđiyle orantılı olan sayısal bir uygunluk deđerini verir. Bu bilgi, her kuřakta daha uygun çözümlerin seçiminde yol göstermektedir. Bir çözümün uygunluk deđerini ne kadar yüksekse, yaşama ve çođalma şansı o kadar fazladır ve bir sonraki kuřakta temsil edilme oranı da o kadar yüksektir (Yeniay, 2001: 38-39).

## 2.4. Çođalma İşleminin Uygulanması

Çođalma operatöründe diziler, amaç fonksiyonuna göre kopyalanır ve iyi kalıtsal özellikleri gelecek kuřađa daha iyi aktaracak bireyler seçilir. Üreme operatörü yapay bir seçimdir. Dizileri uygunluk deđerlerine göre kopyalama, daha yüksek uygunluk deđerine sahip dizilerin, bir sonraki kuřaktaki bir veya daha fazla yavruya daha yüksek bir olasılıkla katkıda bulunması anlamına gelmektedir. Çođalma, bireyleri seçme işleminden, seçilmiş bireyleri bir eşleme havuzuna kopyalama işleminden ve havuzda bireyleri çiftler halinde gruplara ayırma işleminden oluşur (Fıđlalı, 2002: 3).

Uygunluk deđerinin hesaplanması adımıdan sonra mevcut kuřaktan yeni bir populasyon yaratılmalıdır. Seçim işlemi, bir sonraki kuřak için yavru üretmek amacıyla hangi ailelerin yer alması gerektiđine karar vermektedir. Bu dođal seçimdeki en uygunun yaşaması durumuna benzerdir. Bu yöntemin amacı, ortalama

uygunluğun üzerindeki değerlere çoğalma fırsatı tanımaktır. Bir dizinin kopyalanma şansı, uygunluk fonksiyonuyla hesaplanan dizinin uygunluk değerine bağlıdır (Jang, 1997: 176). Seçim yöntemlerine rulet tekerleği seçimi, turnuva seçimi ve sıralama seçimi gibi seçim yöntemleri örnek verilebilir.

## **2.5. Çaprazlama İşleminin Uygulanması**

Mevcut gen havuzunun potansiyelini araştırmak üzere, bir önceki kuşaktan daha iyi nitelikler içeren yeni kromozomlar yaratmak amacıyla çaprazlama operatörü kullanılmaktadır. Çaprazlama genellikle, verilen bir çaprazlama oranına eşit bir olasılıkla seçilen aile çeşitlerine uygulanmaktadır (Jang, 1997: 176).

Genetik algoritmanın performansını etkileyen önemli parametrelerden biri olan çaprazlama operatörü doğal populasyonlardaki çaprazlamaya karşılık gelmektedir. Çoğalma işlemi sonucunda elde edilen yeni populasyondan rastsal olarak iki kromozom seçilmekte ve karşılıklı çaprazlama işlemine tabi tutulmaktadır. Çaprazlama işleminde dizi uzunluğu  $L$  olmak üzere,  $1 \leq k \leq L-1$  aralığında  $k$  tamsayısı seçilmektedir. Bu tamsayı değerine göre dizi çaprazlamaya uğratılır. En basit çaprazlama yöntemi tek noktalı çaprazlama yöntemidir. Tek noktalı çaprazlama yapılabilmesi için her iki kromozomun da aynı gen uzunluğunda olması gerekir. İki noktalı çaprazlamada ise kromozom iki noktadan kesilir ve karşılıklı olarak pozisyonlar yer değiştirilir (Fırlı, 2002: 3).

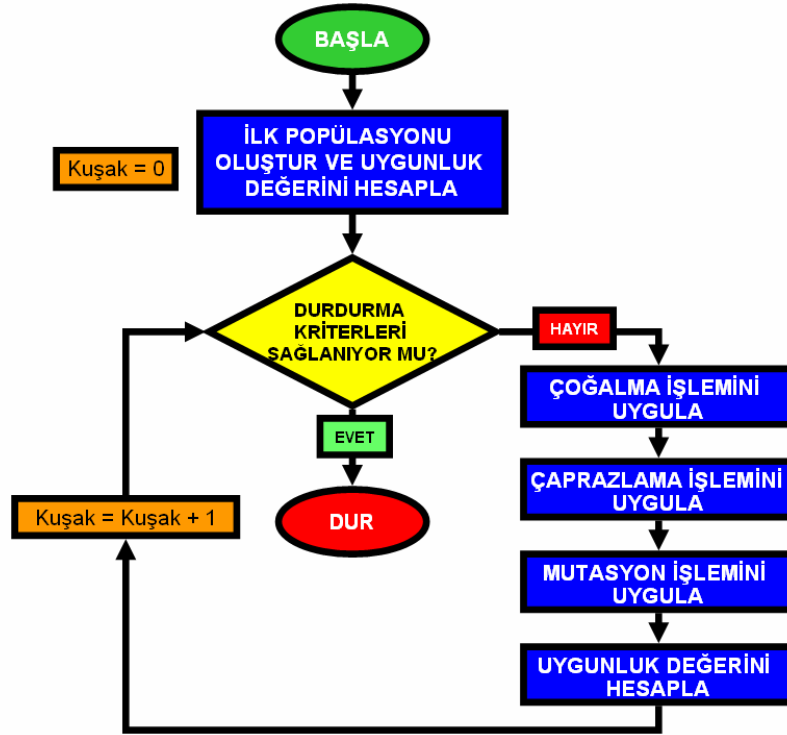
## **2.6. Mutasyon İşleminin Uygulanması**

Çaprazlama mevcut gen potansiyellerini araştırmak üzere kullanılır. Fakat populasyon gerekli tüm kodlanmış bilgiyi içermez ise, çaprazlama tatmin edici bir çözüm üretmez. Bundan dolayı, mevcut kromozomlardan yeni kromozomlar üretme yeteneğine sahip bir operatör gerekmektedir. Bu görevi mutasyon gerçekleştirir. Yapay genetik sistemlerde mutasyon operatörü, bir daha elde edilemeyebilir iyi bir çözümün kaybına karşı koruma sağlamaktadır (Goldberg, 1989: 14). İkili kodlama sisteminin kullanıldığı problemlerde mutasyon, düşük bir olasılık değeri altında bir bit değerini (0 veya 1 olabilir) diğer bit değerine dönüştürür. İkili kodlama sisteminin kullanılmadığı problemlerde ise daha farklı mutasyon yöntemleri kullanılmaktadır. Hangi yöntem kullanılırsa kullanılsın, mutasyonun genel amacı, genetik çeşitliliği sağlamak veya korumaktır (Braysy, 2001: 54).

## **2.7. Yeni Kuşağın Oluşması ve Döngünün Durdurulması**

Yeni kuşak çoğalma, çaprazlama ve mutasyon işlemlerinden sonra tanımlanmakta ve bir sonraki kuşağın ebeveynleri olmaktadır. Süreç yeni kuşakla

çoğalma için belirlenen uygunluk ile devam eder. Bu süreç, önceden belirlenen kuşak sayısı kadar veya bir hedefe ulaşıncaya kadar ya da başka bir durdurma kriteri sağlanana kadar devam eder (Yeo ve Agyel, 1998: 271). İstenen hassasiyet derecesine göre de maksimum iterasyon sayısı belirlenebilmekte ve iterasyon bu sayıya ulaştığında döngü durdurulabilmektedir. Durdurma kriteri iterasyon sayısı olabileceği gibi hedeflenen uygunluk değeri de olabilmektedir (Fung, Tang ve Wang, 2001: 270).



Şekil-E8.1: Genetik Algoritma Akış Diyagramı

### 3. GENETİK ALGORİTMALARDA PARAMETRE SEÇİMİ

Parametreler, genetik algoritma performansı üzerinde önemli etkiye sahiptir. Optimal kontrol parametreleri bulmak için bir çok çalışma yapılmıştır fakat tüm problemler için genel olarak kullanılacak parametreler bulunamamıştır (Altıparmak, Dengiz ve Smith, 2000: 61-62). Bu parametreler, kontrol parametreleri olarak adlandırılmaktadır. Kontrol parametreleri popülasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, kuşak aralığı, seçim stratejisi ve fonksiyon ölçeklemesi

olarak sayılabilir. Bu parametreler aşağıda açıklanmıştır (Yeniay, 2001: 40; Sinreich ve Samakh, 1999: 89; Deb:10):

- **Populasyon Büyüklüğü:** Genetik algoritma kullanıcısı tarafından verilen en önemli kararlardan birisidir. Bu değer çok küçük olduğunda, genetik algoritma yerel bir optimuma takılabilmektedir. Populasyonun çok büyük olması ise çözüme ulaşma zamanını arttırmaktadır. Bu konuda Goldberg 1985'de, yalnızca kromozom uzunluğuna bağlı bir populasyon büyüklüğü hesaplama yöntemi önermiştir. Ayrıca Schaffer ve arkadaşları 1989'da çok sayıda test fonksiyonları üzerinde yaptıkları araştırmalar sonucunda, 20-30 arası bir populasyon büyüklüğünün iyi sonuçlar verdiğini belirtmişlerdir.

- **Çaprazlama Olasılığı:** Çaprazlamanın amacı, mevcut iyi kromozomların özelliklerini birleştirerek daha uygun kromozomlar yaratmaktır. Kromozom çiftleri  $P(c)$  olasılığı ile çaprazlamaya uğramak üzere seçilirler. Çaprazlamanın artması, yapı bloklarının artmasına neden olmakta fakat aynı zamanda bazı iyi kromozomların da bozulma olasılığını arttırmaktadır.

- **Mutasyon Olasılığı:** Mutasyonun amacı populasyondaki genetik çeşitliliği korumaktır. Mutasyon  $P(m)$  olasılığı ile bir kromozomdaki her bitte meydana gelebilir. Eğer mutasyon olasılığı artarsa, genetik arama rastsal bir aramaya dönüşür. Fakat bu aynı zamanda kayıp genetik malzemeyi tekrar bulmada yardımcı olmaktadır.

- **Kuşak Aralığı:** Her kuşaktaki yeni kromozom oranına kuşak aralığı denmektedir. Genetik operatörler için kaç tane kromozomun seçildiğini gösterir. Yüksek bir değer bir çok kromozomun yer değiştirdiği anlamına gelmektedir.

- **Seçim Stratejisi:** Eski kuşağı yenilemenin çeşitli yöntemleri mevcuttur. Kuşaksal stratejide, mevcut populasyondaki kromozomlar tamamen yavrular ile yer değiştirir. Populasyonun en iyi kromozomu da yenilendiğinden dolayı bir sonraki kuşağa aktarılamaz ve bu yüzden bu strateji en uygun (elitist) stratejisiyle beraber kullanılmaktadır. En uygun stratejisinde, populasyondaki en iyi kromozomlar hiçbir zaman yenilenmemektedir, bundan dolayı çoğalma için en iyi çözüm her zaman elverişlidir. Denge durumu stratejisinde ise, her kuşakta yalnızca birkaç kromozom yenilenmektedir. Genellikle, yeni kromozomlar populasyona katıldığında en kötü kromozomlar yenilenir.



- **Fonksiyon Ölçeklemesi:** Doğrusal ölçekleme, üstsel ölçekleme gibi yöntemler mevcuttur. Probleme göre en uygun ölçekleme yönteminin seçilmesi genetik algoritmanın etkin işlemesi açısından önem taşımaktadır.

#### **4. TAŞIMA PROBLEMİ ÇÖZÜMÜ İÇİN GELİŞTİRİLEN ALGORİTMA**

##### **4.1. Genel**

Karar destek sistemleri, karar vericinin bilgi ve deneyimini karar yöntem ve teknolojilerinin sunduğu imkânlarla birleştirerek karar kalitesinde “hissedilir” bir artış hedefleyen bilgi sistemleridir.

Taşıma planlama problemleri akademik anlamda zor (NP-Hard, NonPolynomial-Hard) problemler kümesinde yer almaktadır. En uygun, gerçek/gerçeğe yakın ve karar vericiyi tatmin edici çözümler bulmada, karar vericinin sahip olduğu bilgi ve deneyim yanında geliştirilen karar destek sisteminin probleme özel çözücü yeteneği de eşit derecede önemlidir. Karar destek sistemi karar vericiye problemin en iyi çözümünü sunan bir matematik model sunabilse de bu çözüme zamansal bakımdan da istenilen sürede ulaşılabilmesi önem taşımaktadır. Bu nedenle karmaşık ve çözümü zor bir model üzerinde çalışılırken en iyinin yanında kısa sürede ulaşılacak ve en iyiye yakın çözümler bulabilecek yöntemlerin de sunulması bu tarz problemlerde kaçınılmaz olmaktadır. En iyiye yakın çözümlerin bulunmasını sağlayabilen ve son yirmi yılın eniyileme literatürünün en baskın ve yaygın örneklerini içeren bu yöntemler meta-sezgisel (sezgisel/yarı sezgisel) başlığı altında toplanmaktadır. Meta-sezgisel yöntemler içinde ise genetik algoritmalar ayrıca dikkat çekici bir yere sahiptir.

Bu kapsamda, çalışmaya konu olan taşıma problemine yönelik kısıt ve hedef gözetken çok-öbekli ve çok-amaçlı bir genetik algoritma geliştirilmiştir. Algoritma, en iyi veya en iyiye yakın çözümleri kısa bir sürede bulmayı ve kendisine tanınan arama zamanına bağlı olarak iyi çözümler bulmayı belirli bir güvenle gerçekleştirecek özelliklere sahiptir. Bunun yanı sıra öncelikle uygulanabilir (feasible) çözümlere ulaşılması ve akabinde maliyet etkin çözümlerin araştırılmasını/bulunmasını sağlayacak bir yöntem ortaya konmaya çalışılmıştır.

Taşıma problemi (TP) yolcu ve/veya malzemelerin buldukları konumdan ulaşmaları istenen hedef noktaya/noktalara verilen bir takvime uygun olarak taşınmalarının sağlanmasını hedeflemektedir. Bu hedef çerçevesinde;

- Taşımaya uygun yolların oluşturduğu bir rotanın belirlenmesi,

- Yolcu ve malzemeleri taşıyacak araçların elde mevcut imkânlar kapsamında tahsisinin yapılması ve uygun bir zaman içinde istenilen yerde hazır bulunmalarının sağlanması,

- Zorunlu hallerde farklı ulaşım modlarının kullanılması,

- Eldeki öz kaynakların yetersiz kaldığı durumlarda hizmet satın alınması,

- Bütün faaliyetlerin öncelikle takvim sonra da maliyet gözetilerek yerine getirilmesi öngörülmektedir.

Bu faaliyetler hem matematiksel hem de sezgisel modelin çözmeye çalıştığı küçük problemlerdir.

#### **4.2. Problemin Karmaşıklığı**

Taşıma problemi, belirli bir büyüklüğe kadar en iyi çözümü veren daha küçük problemlerden oluşmuştur. Bunlar:

##### **4.2.1. Rotalama**

Taşınacak unsurların buldukları konumlarından hedef noktaya ulaşmada kullanacakları yolun belirlenmesi problemidir. Bu problem taşınacak bir unsurun her ulaşım modelinde (karayolu, demiryolu, havayolu ve denizyolu) mevcut olan kaynaklarla, hedef noktaya verilen zaman kısıtlarına uygun olarak ulaşmasını sağlayacak faaliyetlerin (yollar ve kontrol noktaları) çizelgelenmesini hedeflemekte ve bu özelliği bakımından da **“kaynak kısıtlı proje çizelgeleme problemine”** karşılık gelmektedir. Yani problem, tek bir unsur ve yol sayısının tek olduğu durumlarda polinom değil üstel zamanlı bir çözüm süresine sahiptir. Başka bir ifade ile bu unsurun kullanabileceği yol ile o yolda harcayacağı zaman arasındaki ilişki doğrusal değildir. Bu özellik klasik kaynak kısıtlı proje çizelgeleme problemine ayrıca bir güçlük faktörü getirmektedir. Çünkü NP-hard olarak bilinen bu problemlerde ayrılan kaynaklar faaliyet sürelerini değiştirmemektedir.

##### **4.2.2. Güzergâha Uygun Ekonomik Araç Tahsisi**

Taşınacak unsurların belirli bir yol üzerinde kullanacakları araçların belirlenmesi problemidir. Taşınacak unsurların taşınmaya esas araç kapasite ihtiyaçlarının (ağırlık, hacim, yolcu, kendi taşıma edemeyen araçlar) belirlenmesi, daha ekonomik taşıma için araç doğrulamasının yapılması aslında bir kutu yükleme (bin-packing) problemi olup NP (non-polynomial) problemler kümesinde yer almaktadır.

Taşıma planlamasında taşınacak unsurlara tahsis edilecek araçların en uygun yük(ler)le yüklenmesi yanında verilen bir sürede hedef noktaya ulaşımının sağlanması da gerekmektedir.

Bunun yanında araç tahsisi sonucu oluşturulan taşıma filusunun, içlerindeki en yavaş aracın hızında hareket ettikleri farz ve kabul edilmekte ve farklı ulaştırma modellerinin beraber (aynı anda) veya birbirini takip eder şekilde kullanılabilirliği de eniyilemenin gerçekleştirilebilmesi bakımından dikkate alınmaktadır.

#### **4.2.3. Altyapı Kullanımı**

Konumların ve yolların taşınacak unsurlar tarafından belirli bir sıra dahilinde kullanımı problemidir. Bir konum, taşınacak unsurlara park ve elleçleme kapasitesi ve ayrıca ulaştırma modeli değişimi (aktarma) olanağı sunan bir ulaştırma altyapısı veya bir hizmet servis noktası gibi düşünülebilir. Bu noktaya ulaşan unsurların hizmet almaları aslında bir sıralama problemidir. Bu sıralama bahse konu noktaya unsurların istenen en geç varış zamanı içinde ne zaman gelmeleri gerektiği konusunda belirleyici olabilmektedir. Bu açıdan problem, istenenden erken ve geç varışın uygun olmadığı (minimized earliness and tardiness) bir çizelgeleme problemine dönüşmekte ve NP-zor problemler kümesinde yer almaktadır.

Bu alt problemlerin bir arada olduğu taşıma problemi örneğine az rastlanır bir kuramsal çözüm güclüğü taşımaktadır. Bu bağlamda, en iyi çözüme yakınlık her bir alt problemin güclüğü oranında değerlendirilmelidir.

#### **4.3. Önerilen Yöntem**

Önerilen yöntem, taşınacak unsurların önceliklerinin ve varış zamanlarının stokastik olarak işlendiği özel bir genetik algoritmadır. İçinde yerel iyileştirme rutinleri içerdiğinden literatürde bu tarz sezgisel yöntemler “memetik” algoritma başlığı ile de anılmaktadır.

Genetik algoritmalar, 1975 yılında Holland tarafından temel prensipleri ortaya konmuş, daha sonra öğrencileri tarafından geliştirilmiştir. Darwin'in doğal seçim (natural selection) kuramı ile genetik biliminin çaprazlama ve mutasyon esasları kullanılarak problemlere çözüm bulmayı amaçlayan bir yaklaşımdır. 1989'da Goldberg'in konu ile ilgili yayınladığı kitap konuyu geniş akademik ilgiye ulaştırmıştır. Herhangi bir algoritma çözüm uzayından rasgele  $n$  çözüm alıp değerlendirdiğinde, aynı koşullarda genetik algoritmalarda bu değerlendirme  $n^3$  olarak gerçekleşmektedir. Bu özelliği ile “gizil paralel (implicit parallelism)” bir algoritma olarak nitelendirilmektedir.

#### 4.4. Tanımlar ve Varsayımlar

Önerilen yöntem aşağıdaki değerlendirmelere dayandırılmıştır:

- Taşınacak unsurlar planlamadan önce birlikte taşınması gerekli küçük birimlere bölünebilir (bileşenlerine ayrılabilir). Bileşenler birlikte hareket etme esasına dayanan en küçük taşınacak unsurdur ve bütün olarak taşınırlar.

- Bileşenler önceliklendirilebilir. Bu önceliklendirme yol ve kaynak tahsisinde ilgili bileşene üstünlük hakkı verir.

- Karar verici, bir bileşen için yol veya bileşenin taşımada kullanabileceği ulaştırma modelini/modellerini belirleyebilir. Bu durumda yol ve ulaştırma modeli tercihlerine göre yol ve/veya araç tahsisi çalıştırılır.

- Araçların bileşenlere tahsisinde aşağıdaki kapasite ihtiyaçları belirlenen sıraya göre göz önüne alınır:

- Ağırlık (Ton),
- Hacim (m3)
- Yolcu,
- Paletli Araç (Tracked Vehicle-Non self-transportable)
- Tekerlekli Araç (Wheeled Vehicle- Self-transportable, Non self-transportable)
- Römork (Trailer- Towed Vehicle)
- Konteyner (TEU- Twenty-Foot Equipment Unit)
- Paletli araç taşıyan ulaştırma araçları ile tekerlekli araçlar da taşınabilir.

- Farklı ulaşım modelleri için bileşenlerin kapasite ihtiyaçları değişebilir. Bunun nedeni ilgili modellerde kendi kendini taşıyabilen bileşen envanterindeki araçlardır. Bu model değişimi ve araç tahsisinde göz önüne alınır.

- Denizyolu, demiryolu ve havayolunda hız, ilgili model için aynı anda kullanılan araçlardan en düşük seyir hızına sahip olan aracın hızıdır.

- Karayolunda ise, gündüz ve gece seyir hızları, mola/bekleme süreleri ve zaman aralıkları kullanıcı tarafından belirlenecektir.

- Karayolunda gece taşıma yapılmaması durumunda bileşen bulunduğu noktada bekleyecektir (Çalışma saatleri kullanıcı tarafından belirleneceğinden gece taşıma yapılıp yapılmayacağı veya taşımaya geceleyin devam edilip edilmeyeceği de bu sayede belirlenmiş olacaktır).

- Karayolu modu hariç bileşenlerin yükleme/boşaltma süreleri konumların ilgili moda ait elleçleme kapasiteleri ve araçların yükleme/boşaltma süreleri kullanılarak hesaplanır. Bunun yanısıra ulaştırma altyapılarının park yeri kısıtları da dikkate alınır.

- Günlük en çok yolculuk süresi taşımada kullanılan her ulaştırma modu için kontrol edilir.

- Araç yol ve araç konum uyumlulukları planlamada dikkate alınır.

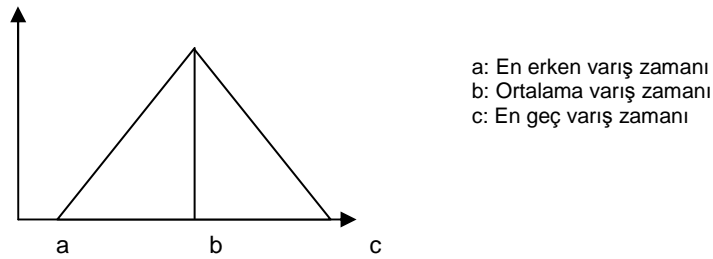
#### **4.5. Önerilen Sezgisel Algoritma**

Önerilen genetik tabanlı sezgisel algoritma Şekil-E8.2'de sunulan kromozom yapısını kullanacaktır. Kromozomlarda bileşenlerin öncelik ve varış zamanlarına ilişkin bilgi bulunmaktadır. Genetik gösterimlerin, ele alınan problemin karar değişkenlerini doğrudan veya dolaylı olarak modellemesi önemlidir. Taşıma planlamasında yer alan kaynak kısıtlarının çözümlenmesi bileşenlerin bu kaynakları kullanmadaki önceliklerinin değerlendirilmesi ile olanaklıdır. Kaynakların farklı öncelik sıralarında bileşenlere sunulması, hangi kaynağın ne zaman hangi bileşen tarafından kullanılacağına belirlenmesini sağlamaktadır. Böylece açıkça ifade edilen en geç varış zamanlarının yanı sıra verilmesi gerekli bir dizi karar öncelik tercihi ile peşinen verilmiş olmaktadır. Bileşenlere öncelik verilmesi, kaynak kullanım problemini bir tür hiyerarşik otomatik karara dönüştürmektedir. Ancak bileşenlerin kendi aralarında belirli kaynakları paylaşmalarını yönlendiren bu yaklaşım, bileşenlerin verilen zaman kısıtları içindeki hareketlerini bir referans hareket zamanına bağlamamaktadır. İkinci önemli karar, bileşenlerin kendilerine verilen en erken ve en geç varış zamanları aralığının içinde gerçekten varmaları en uygun anın belirlenmesidir. Bu da kromozomun ikinci kısmında yer alan ve her bileşenin öngörülen erken ve geç varış zamanlarına uygun, ancak bu iki an arasında, kaynakların daha etkin paylaşımını sağlayacak, varış zamanlarının belirlenmesidir. Böylece farklı varış zamanları belirlenerek, aynı kaynağı kullanma durumunda olan bileşenlere hareket alanı sağlanmaktadır.

| Kromozom No | Bileşen 1  | Bileşen 2 |  | Bileşen N | Var 1                   | Var 2 |  | Var N |
|-------------|------------|-----------|--|-----------|-------------------------|-------|--|-------|
|             | Öncelikler |           |  |           | Bileşen Variş Zamanları |       |  |       |

**Şekil-E8.2:** Kromozom Yapısı

Variş zamanları uygulamada en erken ve en geç varış zamanları aralığında herhangi bir noktada olabilmesine karşın, uç değerler o bileşenin olması muhtemel sapmalar sonucu hızla uygun olmayan çözüm bölgesine çıkmasına sebep olabilirler. Bileşenlerin belirli bir zaman aralığında bir hedef noktaya erişimleri kendileri arasında da bir erişim zamanı uyumunu gerektirebilmektedir. Bu tür koşullarda algortimanın bileşenlerin hareket serbestliğini ençoklayacağı nokta orta noktadır. Ancak bu noktaların de facto varış zamanı olarak belirlenmesi, plan kısıtları belirlenirken öngörülmüş olan bazı bileşen erişim eş zamanlamalarından etkilenecek aynı kaynakları aynı veya benzer zamanlarda talep etmeleri ile sonuçlanabilir. Bu sebeple orta noktayı ön plana çıkaran ancak mutlak olarak bu noktaya bağlanmayan bir yaklaşım olarak üçgen dağılım kullanımı önerilmektedir. Üçgen dağılımın alt ve üst değerleri en erken ve en geç varış zamanları olup, söz konusu problem için eşkenar bir üçgen ile temsil edilir. Böylece bileşenlerin kendilerine ait varış zaman kısıtlarının alt üst değerlerinden uzakta olmaya itilmektedirler. Üçgen dağılımın ortalaması  $\{(a+4b+c)/6\}$  olup, erken ve geç varışlar arasındaki süre simetrik dağıldığından bu ortalama değer  $\{(a+c)/2\}$  olacaktır. Dağılımın bu yaklaşımda kullanılacak simetrik şekliyle görünümü Şekil-E8.3'de verilmiştir. Populasyon büyüklüğü bileşen sayısına bağlı olarak 20 ile 100 arasında kullanıcı tarafından değiştirilebilen sayıda kromozomdan oluşur.



**Şekil-E8.3:** Simetrik Üçgen Dağılımı

#### 4.6. Sezgisel Algoritma Adımları (Şekil-E8.4)

##### 4.6.1. Ön Hesaplamalar

Bu işlemin amacı algoritmanın çalışması sırasında sıkça ihtiyaç duyacağı hesaplamaların bir kez gerçekleştirilip hafızada tutulmasıdır. Bu hesaplamalar şunlardır:

- **Bileşenlerin mevcut konumları ve gidecekleri hedef nokta arasındaki en kısa  $K_1$  eş değer güzergâhın belirlenmesi:** Eş değer güzergâh, taşınacak unsurların görev ve konuş konumları arasındaki farklı ulaştırma modellerinin belirli bir maliyet katsayısına bağlı olarak birbirleri ile karşılaştırılmasına izin verecek şekilde yeniden hesaplanmasıdır. Bu hesap gerçek uzaklığı değil yolların bir diğerine göre tercih edilebilirliğini gösterir. Söz konusu dönüşüm katsayıları kullanıcı tarafından güncellenebilir. Bileşenlerin güzergâh seçimi rutini tarafından kullanılır. Varsayılan katsayılar Tablo-E8.1’de verilmiştir.

**Tablo-E8.1:** Ulaştırma Modları Varsayılan Dönüşüm Katsayıları

| Ulaştırma Modeli    | Denizyolu | Demiryolu | Karayolu | Havayolu |
|---------------------|-----------|-----------|----------|----------|
| Göreceli Maliyetler | 0.006     | 0.03      | 1        | 4.40     |

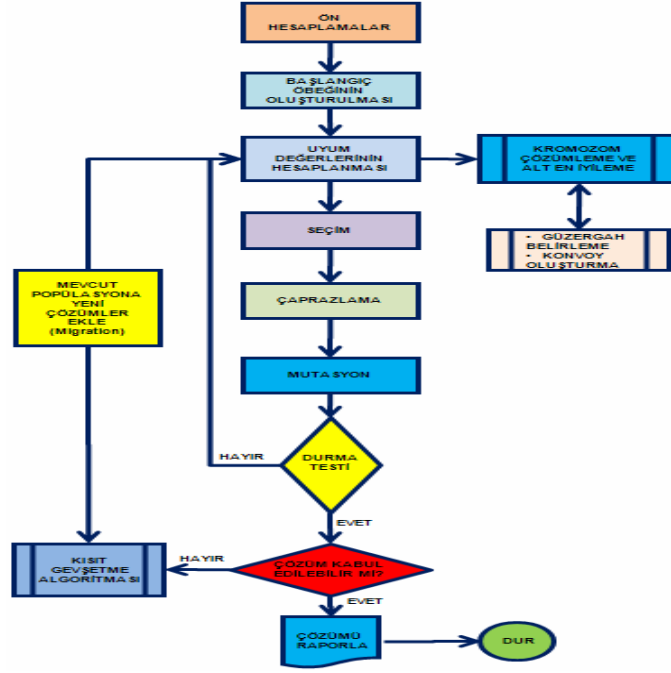
- **Her araç tipi için uyumlu oldukları yollar esas alınarak en kısa  $K_2$  güzergâhın hesaplanması:** Güzergâhlar bileşenlerin mevcut konumları, hedef varış noktaları, araçların plan başlangıç noktaları ve madde (1) ile tespit edilen mod değişim noktaları arasında hesaplanır.

##### 4.6.2. Başlangıç Öbeğinin Oluşturulması

Başlangıç öbeği çözüme katkı sağlaması beklenen bazı basit kurallar ve rassal seçimlerle oluşturulur. Kromozomda yer alan bileşenlerin varış zamanı, en erken varış ile en geç varış zamanları arasında üçgen dağılıma göre oluşturulur. İki bileşenin varışı arasında zaman kısıtı belirlenmiş ise bu durum kromozomların oluşturulmasında göz önüne alınır.

- Bileşenleri en erken ayrılış zamanlarına göre sırala. En erken ayrılış zamanlı bileşen en öncelikli olacak şekilde kromozom oluştur.
- Bileşenleri en erken varış zamanlarına göre sırala. En erken varış zamanlı bileşen en öncelikli olacak şekilde kromozom oluştur.
- Bileşenleri en geç varış zamanlarına göre sırala. En geç varış zamanlı bileşen en öncelikli olacak şekilde kromozom oluştur.
- Bileşenleri en erken ayrılış ve en geç varış zamanları arasındaki farklara göre sırala. Belirtilen farkı en az olan bileşen en öncelikli olacak şekilde kromozom oluştur.
- Bileşenleri kapasite ihtiyaçlarına göre sırala. En çok kapasite ihtiyacı olan bileşen en öncelikli olacak şekilde kromozom oluştur.
- Bileşenleri en kısa ikinci yol ve en kısa birinci yol mesafe farklarına göre sırala. Aradaki farkı en büyük olan bileşen en öncelikli olacak şekilde kromozom oluştur.
- Bileşenleri [(en erken varış zamanı – en erken ayrılış zamanı)x(taşınacak ağırlık)] değerlerine göre sırala. Bu elde edilen değeri en büyük olan bileşen en öncelikli olacak şekilde kromozom oluştur. Aynısını diğer bileşen büyüklük çeşitleri içinde hesaplayarak oluştur.
- En kısa K1 yol içinde mod değişimi en fazla olan öncelikli olacak şekilde bileşenleri önceliklendir.
- En kısa K1 yol içinde en uzun yola sahip olan öncelikli olacak şekilde bileşenleri önceliklendir.
- Kalan kromozomları bileşenlerin rassal önceliklerini sağlayacak şekilde 0-1 aralığında atanacak rassal değerler üzerinden önceliklendir.





Şekil-E8.4: Algoritma İş Akış Şeması

#### 4.6.3. Uyum Değerlerinin Hesaplanması

Genetik algoritma makro düzeyde bileşenlere öncelik verip, görev konumuna varış zamanlarını belirlemektedir. Her iki karar, gerçekte bir dizi alt kararın çözüm uzayını daraltma işlevi taşımaktadır. Söz konusu kararlar, kromozomlardan gelen bilgiler doğrultusunda, bileşenin hangi güzergâhı kullanarak ve o güzergâhta hangi araçlarla taşınırsa bileşenin ilgili kromozomunun belirlediği varış zamanında varılabileceğinin belirlenmesine yöneliktir. Bu alt kararlar yerel eniyileme ile çözüme kavuşturulmakta olup, taşıma maliyetini gözetmektedir. Verilen kararların yarattığı çözümün deterministik simülasyonu ile çözüme ait tüm gösterge ve zamanlar hesaplanabilmektedir. Böylece yerel eniyilemenin gerçek zamanlı uygulanabilir bir çözüm olup olmadığı ortaya konabilmektedir. Uyum değerlerinin hesaplanması iki adımdan oluşmakta olup, simülasyonun güzergâh tahsisi ve araç tahsisi rutinleri ile karşılıklı etkileşimi ve ayrıntıları *Kromozom Çözümleme ve Alt-Eniyileme* bölümünde sunulmuştur:

- Simülasyon: Güzergâh tahsisi ve araç tahsisi rutinlerinden gelen bilgilere uygun olarak deterministik bir sistem simülasyonu gerçekleştirir. Derlediği zaman ve göstergeleri uyum değeri denklemine gönderir.

- Uyum değeri, Kromozom Çözümleme ve Alt-Eniyileme bölümünde Deterministik Simulasyon alt başlığında ele alınıp, belirlenmiş olan sapma ve maliyetleri kullanılarak, Denklem-1'e göre her bir kromozom için ayrı ayrı hesaplanır.

*Uyum Değeri (Kromozom j) =  $k_1(\text{Bileşenlerin toplam gecikmesi})^{P^1} + k_2(\text{Bileşenlerin toplam erken varışı sayısı})^{P^2} + k_3(\text{Bileşenlerin erken yola çıkışı})^{P^3} + k_4(\text{Bileşenlerin taşıma maliyeti+taşıma esnasında bekleme maliyeti})^{P^4} + k_5(\text{Bileşenlerin araç temin sürelerinin plan öncesine sapması})^{P^5} + k_6(\text{İki bileşen arasında olması gereken maksimum süreden toplam sapma})^{P^6}$*

*UD(Kromozom j) =  $k_1(z)^{P^1} + k_2(\text{Bileşenlerin n toplam erken varışı})^{P^2} + k_3(\text{Bileşenlerin erken yola çıkışı})^{P^3} + k_4(\text{MC+WC})^{P^4} + k_5(\text{Bileşenlerin araç temin sürelerinin plan öncesine sapması})^{P^5} + k_6(\text{İki bileşen arasında olması gereken maksimum süreden toplam sapma})^{P^6}$*

#### Denklem-1

**k** ve **p** katsayıları sapmaların yönlendirilmesi ve bunlara yönelik toleransları yansıtmaktadır. Bunlar karar vericinin problem çözümünden beklentilerine göre belirlenmelidir. Ancak eğer karar verici böyle bir tercih yapmayı arzu etmez ise tüm katsayılar 1 olarak atanır. Maliyet değeri her koşulda sıfırdan büyük bir değer olduğundan, bunun çizelge sapmalarından daha ön plana çıkmasını önlemek için ya  $k_4$  0-1 aralığında bir değer olarak ve/veya  $P_4=1/3$  veya daha küçük bir üssel değer kullanılmalıdır.

Denklem-1'de maliyet ifadesi dışındaki ifadeler bileşenlerin uymaları istenen taşıma kısıtlarında ortaya çıkan sapmalar olup sıfır değeri ile sonuçlanmaları beklenir. Sıfır değeri taşıma planında öngörülen kısıtlara tam uyulduğunu gösterir. Denklem-1'de kısıtlar dışında temsil edilen diğer uyum göstergesi maliyetler olup taşıma ve bekleme maliyetlerini içermektedir. Maliyetlere karşı gelen terim her koşulda planın kapsamına bağlı olarak pozitif bir değer alacak olup, taşıma kısıtlarının tümünün sağlanması halinde algoritmanın maliyetlere yoğunlaşmasını ve arama uzayının düşük maliyetli kesimlerini değerlendirmesini hedeflemektedir.

Uyum değerleri, kromozomun içerdiği çözümün kısıtlardan sapmalar ve taşıma maliyetlerinden oluşan ve yine sonuçta çözümün kalitesini maliyet olarak ifade eden bir denklemden hesaplandığından, önerilen yaklaşımda büyük uyum

değerleri artan maliyete, küçük uyum değerleri ise azalan maliyete karşı gelmektedir.

#### **4.6.4. Seçim**

Tüm kromozomların uyum değerleri hesaplandığında en düşük uyum fonksiyonu değeri en kaliteli çözümleri temsil etmektedir. Kromozomlar uyum değerlerine göre küçükten büyüğe sıralandığında en düşük değere sahip olan öbek büyüklüğünün %20 si kadar sayıda kromozom popülasyona aktarılır. Varsayılan “en iyi %20” değeri kullanıcı tarafından arzu edilen seçim baskısını yaratacak şekilde yeniden belirlenebilir. Bu yaklaşım literatürde elitist strateji olarak adlandırılır ve mutasyon ve çaprazlamanın olası yıkıcı etkilerinden belirli bir grubun daha iyi çözümler bulunana kadar korunmasını sağlar. Geriye kalan kromozomlar arasında stokastik turnuva seçim tekniği ile popülasyon tamamlanır.

Stokastik turnuva seçim tekniği verilen bir turnuva büyüklüğü parametresine bağlı olarak kromozomların seçilmelerini öngörür. Turnuva büyüklüğü uyum değerlerine göre karşılaştırılacak kromozom sayısını gösterdiğinden, artan turnuva büyüklüğü öbek üzerindeki seçim baskısını arttırıp, çeşitliliği azaltıcı ve yerel en iyi çözümlere itici etki gösterir. En küçük turnuva büyüklüğü iki olup çeşitliliği koruyucu özelliğe sahiptir. Önerilen yaklaşımda elitizm uygulandığından turnuva büyüklüğünün getireceği seçim baskısına gerek duyulmayacaktır. Verilen parametre sayısında (turnuva büyüklüğü) kromozom öbekten eşit olasılıkla ve rasgele seçilir ve bunlar arasından rulet tekeri yaklaşımına göre herhangi biri seçilerek izleyen öbeğe aktarılır. Turnuva büyüklüğü öbek büyüklüğüne eşit alınırsa yöntem tümüyle rulet tekeri yöntemine dönüşür. Turnuva büyüklüğü en az iki olmakla beraber, kullanıcı tarafından değiştirilebilir.

#### **4.6.5. Çaprazlama**

Çaprazlama operatörü, herhangi bir anda öbekte bazı kromozomların taşıdığı iyi çözüm özelliklerinin (çözüm yapıtaşları) diğer bireylere yayılmasını destekler. Böylece bir dizi farklı kromozomun yakaladığı “alt-iyi” genler (çözüm yapıtaşları) diğer çözümlerde mevcut diğer “alt-iyi” genlerle birleşerek öbekte yer alan yeni çözümlerin oluşturulmasını sağlar.

Seçim sonucu oluşturulan yeni öbekteki seçkin bireylere, verilen çaprazlama oranına uygun olarak DeJong’un parametrik çaprazlama operatörü uygulanır. Bu bileşenler arasında öncelik değişimi veya varış zamanlarının değişimi ile sonuçlanır.

**Tablo-E8.2'de** çaprazlama prosedürü örneklenmiştir. Önerilen çaprazlama operatörünün hangi kromozomlara uygulanacağı Çaprazlama\_Oranı ile belirlenir ve bu oran öbek büyüklüğünün bir yüzdesini gösterir. Çaprazlamaya tabi tutulacak kromozomlar eşit olasılıkla seçilir.

Çaprazlama işlemi, taşıma kısıtlarına uygun olan bireyler arasında uygulanır.

**Tablo-E8.2:** Çaprazlama Operatörünün Kullanımı Örneği

|             |      |      |     |     |
|-------------|------|------|-----|-----|
| Krom1       | 0.45 | 0.78 | 45  | 19  |
| Krom2       | 0.30 | 0.20 | 20  | 16  |
| RassalS     | 0.50 | 0.73 | 0.8 | 0.4 |
| XoverPr=0.6 | <    | >    | >   | <   |
| Çocuk1      | 0.45 | 0.20 | 20  | 19  |
| Çocuk2      | 0.30 | 0.78 | 45  | 16  |

#### 4.6.6. Mutasyon

Mutasyon operatörü, o anki öbekte yer almayan ve çözümünü iyileştirmesi olası özelliklerin kromozomlara eklenmesini ve böylece çeşitliliğin korunmasını ve çözüm uzayının incelenmemiş kesimlerinin, kromozomların performanslarına olası katkılarını yakalayarak, o anki öbekte yer alan çözümleri bütünsel en iyiye getirme işlevini yerine getirir. Önerilen mutasyon operatörü iki farklı şekilde uygulanır:

- Öbekte en kötü uyum değerine sahip olan kromozomlardan (en kötü %5) başlangıç öbeğinde olduğu gibi rassal olarak türetilen bir başka kromozomla yer değiştirmesini sağlar. Buna literatürde göç (migration) denir.
- Klasik anlamda mutasyon, verilen mutasyon oranı ile her kromozomda bileşenlerin öncelikleri veya varış zamanlarının yer değiştirmesini sağlar. Bunun için bir gen rassal seçilir. Söz konusu gen eğer önceliği temsil ediyorsa önceliği rasgele bir genle yer değiştirilir. Böylece iki bileşenin önceliği değişir. Eğer gen varış zamanını temsil ediyorsa varış zamanı en erken varış ve en geç varış arasında düzgün dağılıma uygun olarak değiştirilir.

Mutasyon oranının varsayılan değeri %25 olarak ön görülmüştür. Bu değer kullanıcının istemine bağlı olarak değiştirilebilir. Varsayılan oran, öbekteki kromozomların dörtte birinde değişiklik gerçekleşeceği anlamına gelir.

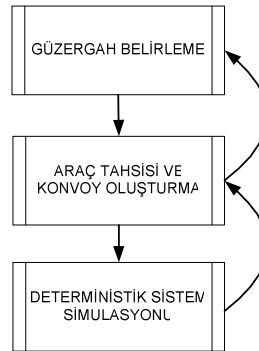
Mutasyon operatörü popülasyondaki tüm bireylere uygulanır.

#### 4.6.7. Durma Testi

Algoritma kullanıcının problemden beklentilerine göre belirlediği bir yineleme (iteration) sayısı veya çalışma süresine ulaşıldığında ya da yinelemeler arasında öbekte bir süre hiç bir iyileşme kaydedilmemesi durumunda kullanıcıya durdurma seçeneği sunarak sona erer.

#### 4.6.8. Kromozom Çözümleme ve Alt-Eniyileme (Şekil-E8.5)

- Kromozomdan bileşenlerin öncelik ve varış bilgilerinin alınması.
- Yukarıdaki madde ile alınan bilgiler bir çözüme esas makro bilgilerdir. Önceliği ve varış zamanı belirlenmiş ancak henüz çizelgelenmemiş her bileşen için (a-b-c) adımları (Güzergâh Belirleme-Araç Tahsisi ve Konvoy Oluşturma-Deterministik Simulasyon) izlenir. Bu bilgilerin öngördüğü çözüm aşağıdaki karar noktaları üzerinden ve geriye doğru çizelgeleme ile oluşturulur ve gerekli bilgiler tutulur.



**Şekil-E8.5:** Alt-Eniyileme Bileşenleri ile Sistem Simulasyonu Etkileşimi

#### 4.6.9. Rotalama

Öncelikle 4.2.7.1.(Ön Hesaplamalar)'de oluşturulan en kısa  $K_1$  eşdeğer güzergâh tablosu esas alınarak kromozomdaki öncelik sırasına göre o an incelenmekte olan bileşen için güzergâh belirlenir. Bu esnada o yolun başka

bileşenler tarafından kullanılıp kullanılmadığı göz ardı edilir. Bileşenin uyumlu olduğu ulaşım modellerinden, konuş ve görev konumları arasındaki en kısa  $K_1$  güzergâh içinden, model değişimi içermiyorsa en kısa olan, içeriyorsa en az model değişimi olanlar içinden en kısa olan seçilir. Seçilen güzergâhtaki farklı tüm ulaştırma modelleri için uyumlu araçlar, kendi kendine hedef noktaya boş olarak taşıma edebilecek ise listelenir ve her model için ihtiyaç kontrolü yapılır. Bu şekilde süzülen araçların taşıma kapasitesi bileşenin ihtiyacını karşılamıyorsa, güzergâh seçimi  $K_1$  en kısa yol listesinden bir önceki güzergâh çıkarılarak tekrarlanır. Bu koşulu sağlayan hiç bir yol olmadığında sistem çözümsüzlük uyarısı verir ve durur. Verilen güzergâhtaki yollar, model değişimi olacak noktalar araç tahsisi rutinine esas teşkil edeceğinden tespit edilir. Araç arama derinliği ve öne çekme sayılarının  $K_1$  yol içinde tüm olası kombinasyonları için algoritmanın sonuç vermemesi Araç Tahsisi ve Konvoy Oluşturma Adım-10 ile sonuç verene kadar sürdürülür. Bu sadece ilgili bileşeni etkiler.

#### 4.6.9.1.Araç Tahsisi ve Grup/Konvoy Oluşturma

- Adım-1: Bileşenin varış zamanı kromozomdan alınır veya ara konumlar için o konuma varış zamanı bir önceki yol için bileşenin yüklenmeye başlanma anı olarak alınır. Bu an Md.4.6.9.2 Adım.4'de tespit edilmiştir.
- Adım-2: Araçlardan, araç tahsisine konu olan yol(lar) ve konum(lar)la uyumlu olanlar süzülür.
- Adım-3: Araçların tercihe esas süreleri (ATES) hesaplanır. Bu süreye uygun olmayan araçlar elenir.

**“ATES=[(Aracın müsait olma anı+Aracın boş olarak gereksinim noktasına ortalama erişim süresi+ Yükleme süresi +Aracın ilgili yolu ortalama aşma süresi)<= Bileşenin varış zamanı-2]”**

koşulunu sağlayan araçlar seçilir (Bileşenin varış zamanından 2 saat düşülmesi gece/gündüz farklarının sebep olabileceği sapmaları ve yolun başka bileşen tarafından kullanılmasının sebep olabileceği beklemleri bertaraf etmek için olup bu varış zamanının doğrudan geciktirileceği anlamına gelmez araç tercihini etkiler). Aracın boş olarak ihtiyaç duyulduğu noktaya erişim süresi ilgili aracın konuş yerinen ihtiyaç duyulduğu noktaya kendi modelinde olan en kısa yol olarak belirlenir.

- Adım-4: Bileşenin taşınacak kapasite türüne göre bazı araçlar listeden çıkarılır. Örneğin yolcu taşıması yok ise sadece buna uygun olan otobüs vb. liste dışı bırakılır.

- Adım-5: Toplam A adet araç kalmış olsun. Bunlar önce ekonomikliklerine [beklenen maliyet(yüksüz katedeceği yolxyüksüz birim maliyet+yüklü katedeceği yolxyüklü birim maliyet)] göre ve sonra meşguliyet saatine göre ayrı ayrı sıralanır. Meşguliyet saati araçların o ana kadar atandıkları görevlerin toplam saatidir. Bir anlamda araç kullanım yoğunluğunu gösterir. Her iki sıralamada da araçlar iki ayrı puan alırlar. Araçların seçilmeleri liste başındaki uygun ve ekonomik araç için A değerinden geriye doğru doğrusal olarak puanlandırılarak yapılır. Birinci araç için örneğin  $A=30$  ve listedeki en az ekonomik araç için  $A=1$  olacaktır. Öte yandan meşguliyet saati daha az olanlar daha yüksek puan alacaklarından, her iki sıralamadaki puanlar toplanarak en yüksekten aşağıya doğru sıralanır. Araçlar, bir araç seçim parametresine bağlı olarak, deterministik bir sıra ile veya rassal olarak seçilir. Buna karşı gelen olasılıklar, belirlenecek rassal sayı aralıkları ile gerçekleştirilir. Böylece ulaştırma modeli, yol ve zaman uyumu olan her aracın kullanımı sıfırdan büyük bir olasılığa sahip kılınır. Her araç seçiminden sonra, birliğin yeni kapasite gereksiniminin hesaplanması ve kalan taşınacak kapasiteye uygun araçların olduğu indirgenmiş bir liste içinden kapasite gereksinimi tümüyle karşılanana kadar seçilmesi devam eder. Bu işlemler öngörülen araç arama derinliği sayısı kadar sürdürülür. Oluşturulan konvoyların ilgili bileşenin kapasite gereksinimi karşısında yetersiz kalması halinde ilgili yol için araç tahsisi ve konvoy oluşturma başarısız olmuştur. Bu, araçların bileşenin istenen hedef varış zamanını sağlayamayacağı anlamına gelir. Adım-6'ya geçilir. Aksi halde oluşturulan konvoy bileşenin kapasite gereksinimine uygun demektir. Adım.7'ye gidilir.

- Adım-6: Verilen zaman penceresi içinde uygun bir konvoy izin verilen arama derinliğinde oluşturulamamıştır. Bu varış zamanının zorlanmasındandır. İlgili yol bileşeni görev konumuna ileten son yol ise, bileşenin varış zamanını mevcut varış zamanı ile en geç varış zamanı arasında üçgen dağılıma uygun yeni bir nokta belirlenir. Öne çekme sayısı bir arttırılır ve Adım.3'e gidilir. Varış zamanı araçların daha müsait oldukları bir başka zaman dilimine çekilmeye çalışılır. Belirtilen varış zamanını öne çekme seçeneği belirli bir deneme sayısından sonra kullanışsız planlara dönüşebilir, çünkü her öne çekme bileşenin ara bir lokasyona erken varması ve beklemesi anlamına gelecektir. Bunu önlemek için tüm olası varış

zamanları aralıklarına bakmak yerine öne çekme sayısı kadar varış zamanını öne çekmek yeterlidir. Öne çekme sayısı izin verilen enbüyük öne çekme sayısından küçük olduğu sürece Adım.3'e dönülür. Dönmeden önce o lokasyona varış zamanı güncellenir. Güncellemede, araç teminine esas süre Bileşen Varış Zamanı yerine, araçlar için sırayla "Bileşen varış zamanı>müsait olma zamanı>0" koşulunu sağlayanlar içinden enbüyük olan alınır. Öne çekmede önerilen yaklaşım kaynak havuzunu değiştiren olayları izleyerek öne alma seçeneğidir. Örneğin birliğin öngörülen varış zamanı 10. saate çizelgelenmiş ve bu saatten geriye giderek yapılan ilk denemede uygun araçlar sağlanamamış olsun. Öngörülen varış zamanı yeni kaynakların serbest kaldığı bölümün önünde yer almalıdır. Öne çekme sayısı> enbüyük öne çekme sayısı ise Adım.8'e gidilir. Aksi halde Adım.7'ye geçilir.

- Adım-6: Uygun araç ataması gerçekleştirilmiş demektir. Bu noktaya kadar yapılan hesaplamalar beklenen değer hesaplaması üzerinden yapıldığından, beklenen sonuç uygun bir çözüm bulunmasıdır. Ancak bunun ayrıntılı değerlendirilmesi izleyen adımda yapılır. Araç konvoyu oluşturulduktan sonra, yol üzerindeki lokasyonların elleçleme+park kapasiteleri kontrol edilir. Eğer sağlanmıyorsa Güzergâh Tahsisi rutini bu lokasyonu dışarıda bırakacak şekilde güncellenir. Böyle bir yol mevcut değil ise bu bileşenin bu lokasyondan geçmesi zorunlu fakat elleçleme kapasitesi sıkıntılı demektir. Araçların elleçleme gereksinimlerini azaltacak şekilde araçlar diğer seçeneklerle değiştirilir. Bunu yaparken elleçleme gereksinimi en çok olan araçtan başlanarak uygun araçlar listesindeki araçlarla ikili değiştirmeler yapılır. Bu yapılırken diğer aracın kapasitesine sahip fakat daha az elleçleme süresi gerektirenler gözönüne alınır. Seçenekler tükendiğinde bu bileşenin taşımaı olanaksız demektir. Raporlanır. Aksi halde araç tahsisi başarılı olmuştur. Deterministik Simulasyon'a geçilir.

- Adım-7: Bileşenin Varış Zamanının Modifikasyonu: Bileşen kromozomun önerdiği varış zamanının öne çekilmesi (öne çekme sayısını aşmadan) ve araç arama derinliğinin öngörülen değeri ile herhangi bir uygun çözüm bulamamış ise, araç seçeneklerini varış zamanını öteleyerek arar. Varış zamanı, bir bileşenin çizelgelenene kadar algoritma tarafından güncellenen, başlangıç değeri kromozomda öngörülen zaman olan değişkendir. Bu öteleme uygun araç tahsisi gerçekleşene kadar varış zamanının ötelenmesi anlamına gelir. Öteleme, öngörülen engeç varış zamanının ötesine de geçebilir. Bu durum gecikme olarak kaydedilir. Öteleme parametrik olarak kromozomda yer alan varış zamanın ertelenmesi anlamına gelir. Gereksinim konumuna kendi kendine taşıma yeteneği olan tüm



araçlar kapasite yeterliliği testinde gözönüne alınacağından, söz konusu öteleme her zaman konvoy oluşturma yeteneği ile sonuçlanır. Her ötelemeden sonra (b.Araç Tahsisi ve Konvoy Oluşturma) rutinini sadece o bileşen için yeni baştan çalıştırma anlamına gelir. Araç arama derinliği yeterli değil ise tüm araçlar serbest kaldığında dahi çözümsüz olabilir. Tüm araçlar serbest kalana kadar öteleme sürebilir. Bu noktada hala araç yetersizliği var ise bu kromozom için araç arama derinliği kullanıcının girdiği değerden adım adım %100'e kadar arttırılır. Bu iki döngü çözümsüzlükle sonuçlanmış ise (arama derinliği ve öteleme) kapasite kontrolü yapıldığı anda kendi kendine taşıma edebilir konumdaki araçlar, bu özelliğe sahip olmayan konumlara gitmişlerdir. Bu durum ancak bu birliğin kullanacağı güzergâh değiştirilerek giderilir.

#### **4.6.9.2.Deterministik Simulasyon**

Bileşenin hangi yolları kullanarak hangi araçlarla öngörülen zamanda görev veya ara konumda olacağı belirlenmiştir. Bileşenin varış zamanından geriye doğru gece-gündüz de izlenerek araç hareketleri, molalar ve elleçleme faaliyetleri simüle edilir. Simülasyon (a), ve (b)'de verilen kararların gerçek zamanlı performansı hakkında gerçeğe en yakın bilgileri sağlarken, uygun olmayan çözümlerle karşılaştığında öteleme veya öne çekme, gerekli durumlarda yeniden araç tahsisi ve konvoy oluşturma rutinine geri gönderme kararlarını verebilmektedir.

- Adım-1: Bileşene varış zamanı ilgili konuma kadar sabitlemiştir.
- Adım-2: Eğer ilgilenilen bileşen varış konumu ise ilgili konumda boşaltma süreleri hesaplanır. Bu yapılırken varış saatinden geriye doğru o gece-gündüz, konum çalışma saatleri vb. gözönünde bulundurulur. Bu işlemlerde herhangi bir karar verilmez sadece süreler hesaplanır ve kaydedilir.
- Adım-3: Bileşenin boşaltma için yolun bitiş konumuna erişiminde karayolu dışındaki ulaştırma modlarında bileşene tahsis edilen en yavaş araç esas alınır. Karayolu modunda molalar gözönüne alınarak hız hesaplanır. Bu hızla kaç saatte ne kadar yol alınacağı, gece, gündüz için değerlendirilir. Böylece yola çıkış zamanının ne olması gerektiği/olacağı belirlenmiş olur. Bu yola çıkış zamanında yol bir başka bileşen tarafından kullanılmak üzere ayrılmış ise o bileşenin hareketi bir saat bekleni, yani yol boşalana kadar bekleme seçeneği kullanılır. Birden fazla bileşen varsa, o zaman yolu dolmadan kullanma seçeneğine başvurulur. Bileşenin yolu kullanmaya başlama zamanı, yolun ilk bir saatlik boşluğuna kadar öne çekilir. Bu da yolun zaman kartından okunabilir.

- Adım-4: Bileşenin yüklenmesi için araçların tamamının ilgili mod değişim konumunda olması gerekir. Bileşenin ilgili konumdan ayrılış zamanına uygun olarak, elleçleme+park kapasitesinin söz konusu konvoy için alacağı süre hesaplanır. (Bu süre konvoy oluşmadan belirlenemez.) Bileşenin ayrılış zamanından çıkarıldığında elleçleme ve park kapasiteleri izin veriyorsa konumun elleçleme ve park kapasiteleri bu bileşene atanır. Değilse o konumda hesaplanmış elleçleme zamanı bulunana kadar konumunun çizelgesi sondan başa doğru taranır. İlk müsait zaman dilimi bu bileşene atanır. Elleçlemeye başlama zamanı hesaplanır. (Bu işlem için bileşenin boşaltılması gerekir. Ancak bu bir önceki yolun bileşen tarafından geçilmesi işleminin yani o noktaya varışın bir parçası olarak değerlendirilmiştir.) Konumun çalışma süreleri ve gece-gündüz çalışma saatleri gözönüne alınarak hesaplanır. Bu zaman bileşene araçların taşıma zamanından (yani engeç gelen boş aracın geliş zamanı veya bileşenin elleçlenmeye hazır olduğu zaman) daha sonra gerçekleşen zamana göre hesaplanır.

- Adım-5: Elleçlemeye başlama zamanı, mevcut konvoydaki bazı araçların bu zamana yetişmelerine izin vermeyecek kadar olması gerekenden erkene alınmış olabilir. Bunun için elleçlemeye başlama zamanı sabitlenerek, araçların bu süre içinde bu konuma erişip erişemeyeceklerine bakılır.

- Tamamı belirtilen sürede erişiyorsa bu yol için işlemler tamamlanmıştır. Konuş konumu istikametinde güzergâh üzerinde yer alan henüz çizelgelenmemiş yol seçilir ve Araç Tahsisi ve Konvoy Oluşturma rutinine (b)'ye dönülür. Bu yapılmadan önce, simulasyon boyunca tespit edilen zamanlar ilgili kaynakların ve konumların zaman-kartlarına işlenir (Gantt diyagramı).

- Araçlardan en az biri elleçleme zamanına vaktinde yetişmiyorsa, elleçlemenin başlama zamanı esas alınarak, araç tahsisi yenilenir. Aynı yol için (b)'ye dönülür.

- Tüm bileşenleri çizelgelenmiş bu kromozomun uyum değerini hesaplamaya esas tüm bilgiler mevcuttur. Uyum değeri hesaplanır (Adım.6).

- Adım-6: Uyum Değerinin Hesaplanması: İncelenen kromozom için aşağıdaki koşullar gözden geçirilir.

- Bileşenlerden engeç varış zamanından geç varan var mı? Evetse toplam gecikmeyi güncelle

▪ Bileşenlerden en erken varış zamanından erken varan var mı? Evetse toplam erken varışı güncelle

▪ Bileşenlerden en erken ayrılış zamanında erken ayrılan var mı? Evetse toplam erken ayrılışı güncelle.

▪ Bileşen çiftleri arasında olması muhtemel varış zamanları arasındaki farkın belirli bir süreden fazla olmaması koşulunu sağlamayan çift var mı? Evetse toplam bileşen varışları arasındaki fark sapmasını güncelle.

▪ Bileşenin konuş konumundan ayrılabilmesi için tahsis edilecek araçların planın başlangıç zamanını ihlal etmesi söz konusu ise toplam araç temin süresi sapmasını güncelle.

▪ Bileşenlerin her biri için taşıma boyunca kullandıkları kaynakların maliyetlerini hesapla toplam maliyete ekle. Boş bekleme için de bir maliyet ekle.

▪ Uyum fonksiyonuna ekle. (6) dışındakiler sıfır ise kromozom uygun bir çözümü temsil ediyor demektir. Değilse sapmalar için rapor isteniyorsa bu noktada raporlanır.

#### **4.6.10. Kısıt Gevşetme Algoritması (Şekil-E8-6)**

Sırayla aşağıdaki gevşetmeleri yaparak algoritmayı Adım.1'den itibaren çalıştırır. Kısıtların tümüyle sağlandığı anda algoritma kalan gevşetmeleri kullanmaz. Kısıt gevşetmeye esas olan algoritma Şekil-E8.6'da sunulmuştur.

- Satınalma Opsiyonu: Bu opsiyon seçilmiş ise araçlar da belirli bir maliyet ile kullanılabilir araçlar listesinde konumları ile birlikte yer almalıdır. Bu durumda uyum fonksiyonundaki maliyet katsayısı satınalma bedellerinin toplam maliyete yansımaları oranında küçültülerek maliyetin planı sürükleyen unsur olması engellenmelidir.

- Yolun Aynı Bileşenler Tarafından Kullanılabilmesi: 4.6.9.2. maddesindeki 1 saatlik bekleme veya erken yola çıkma süresi sıfır olarak ve 4.6.9.1. maddesindeki iki saatlik tolerans bir olarak uygulanır.

- Ulaştırma Terminal Elleçleme Kapasitesi Sonsuz: Bu durumda çizelgeleme yapılırken elleçleme kapasitesi kontrolü yapılmaz. (b-x) ve (c-v)'de değerlendirilir.

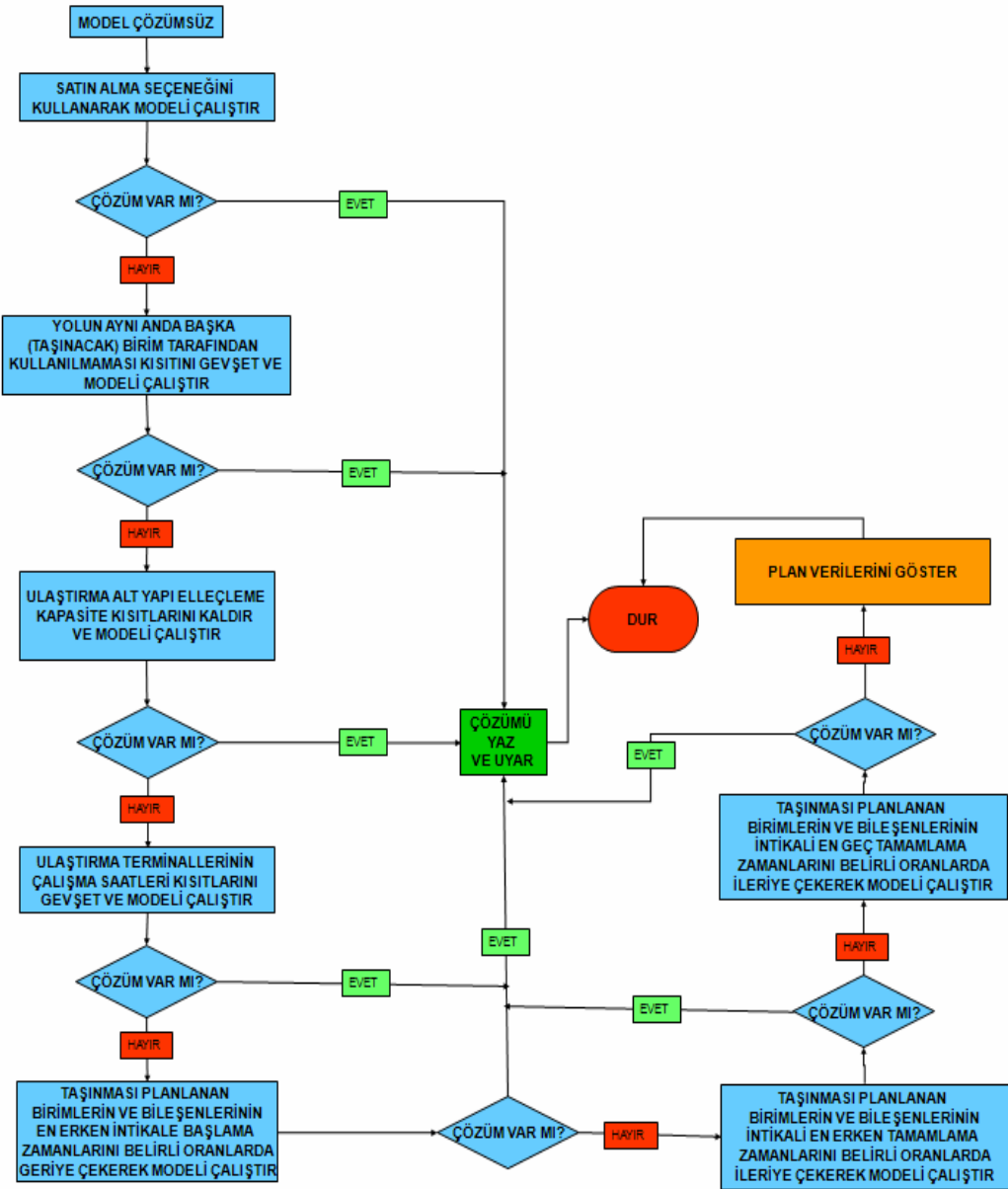
- Ulaştırma Terminallerinin 24 saat çalışması: Bu ilgili arkta çalışma saatlerine bağlı gecikmelerin gözlenmeyeceği anlamına gelir. Bileşenlerin ilgili konumdan ayrılma zamanını kısaltır. Elleçleme süresi hesaplanırken gözönüne alınır. Deterministik Simulasyonda değerlendirilir.
- Bileşenlerin En Erken Taşıma Başlama Zamanlarını Geriye Çekme: Bu bileşenlerin olduğu kadar, taşıma başlama zamanı olan sıfır zamanını da etkileyebilir. Mevcut populasyon söz konusu gevşetme ile yeniden değerlendirilir. Uygunsa raporlanır değilse algoritma çalıştırılır. (6-7) içinde aynı yaklaşım kullanılır.
- Bileşenlerin Taşıma En Erken Tamamlama Zamanlarını Geriye Çekme: “Ulaştırma Terminal Elleçleme Kapasitesi Sonsuz” maddesinde açıklandığı şekilde değerlendirilir.
- Bileşenlerin Taşımayı En Geç Tamamlama Zamanlarını Öteleme: “Ulaştırma Terminal Elleçleme Kapasitesi Sonsuz” maddesinde açıklandığı şekilde değerlendirilir.

#### **4.6.11. Mevcut Öbeğe Yeni Çözümler Eklenmesi**

Algoritmanın kısıt gevşetme rutinini seçmesi verilen bir süre içinde tüm kısıtların sağlanamadığını göstermektedir. Bu durumda yukarıda verilen sıra ile gevşetme seçeneklerinin uygulanması çözüm uzayının genişletilmesi anlamına gelir. Bu genişletme, rutin öncesi uygun olmayan çözümlerin gevşetme sonucu uygun çözümlere dönüşmelerine izin verir. Gevşetme öncesi algoritmanın yakaladığı bir dizi kaliteli “alt-iyi yapıtaşları” gevşetme sonrası algoritmanın performansını arttıracığından, tümüyle yeni bir başlangıç populasyonla aramaya başlanması uygun değildir. Ancak, gevşetme algoritmasının sebep olabileceği veya ihtiyaç duyacağı olası tüm değişikliklerin veya özelliklerin önceki populasyonda yer alması da beklenmez. Belirtilen gerekçeler doğrultusunda, kısıt gevşetme rutini, önceki populasyonun sadece bir kısım çözümlerini yenileri ile değiştirir. Gevşetme rutininin eklediği yeni çözümler için uyum değerleri hesaplanır ve algoritma yeni çözümlerle zenginleştirilmiş önceki populasyonla çalışmaya devam eder.

#### **4.6.12. Çözümün Raporlanması**

Çözümlerin kullanıcıya sunulması, algoritmanın durması üzerine gerçekleşir. Bunun yanında Kullanıcı, algoritma çalışırken ara çözümleri belirli periyotlarla görmeyi seçebilir.



Şekil-E8.6: Kısıt Gevşetme Algoritması.

## KAYNAKLAR

- Altıparmak F., Dengiz B. ve Smith A.E. (2000), "An Evolutionary Approach For Reliability Optimization in Fixed Topology Computer Networks",  
 Braysy O. (2001), Local Search and Variable Neighborhood Search Algorithms for The Vehicle Routing Problem With Time Windows, PH D Thesis.

Deb K., "Genetic Algorithm in Search and Optimization: The Technique and Applications", s. 1-29.

Engin O. (2001), Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Arttırılmasında Parametre Optimizasyonu, İTÜ, Fen Bilimleri Enstitüsü, Yayınlanmamış Doktora Tezi.

Fiğlalı A. ve Engin O. (2002), "Genetik Algoritmalarla Akış Tipi Çizelgelemede Üreme Yöntemi Optimizasyonu", İTÜ Dergisi, s. 1-6.

Fung R.Y.K. , Tang J. ve Wang D. (2001), "Extension Of A Hybrid Genetic Algorithm For Nonlinear Programming Problems With Equality And Inequality Constraints", Computers & Operations Research, Volume: 29, Issue: 3, s. 261-274.

Goldberg D.E. (1989), Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, USA.

Jang J. S. R. (1997), Neuro-Fuzzy and Soft Computing: A Computational Approach To Learning and Machine Intelligence, Chapter 7: Derivative-Free Optimization, Prentice-Hall, USA, s. 173-196.

Sinriech D. ve Samakh E. (1999), "A Genetic Approach to the Pickup/Delivery Station Location Problem in Segmented Flow Based Material Handling Systems", Journal of Manufacturing Systems, Volume: 18, Number: 2, s. 81-99.

Yeniay Ö. (2001), "An Overview of Genetic Algorithms", Anadolu Üniversitesi Bilim ve Teknoloji Dergisi, Cilt: 2, Sayı: 1, s. 37-49.

Yeo M. F. ve Agyel E. O. (1996), "Optimising Engineering Problems Using Genetic Algorithms", Engineering Computations, Volume: 15, Number: 2, s. 268-280.

## EK-9: Tekrarlı Taşıma Planlama Modelleri

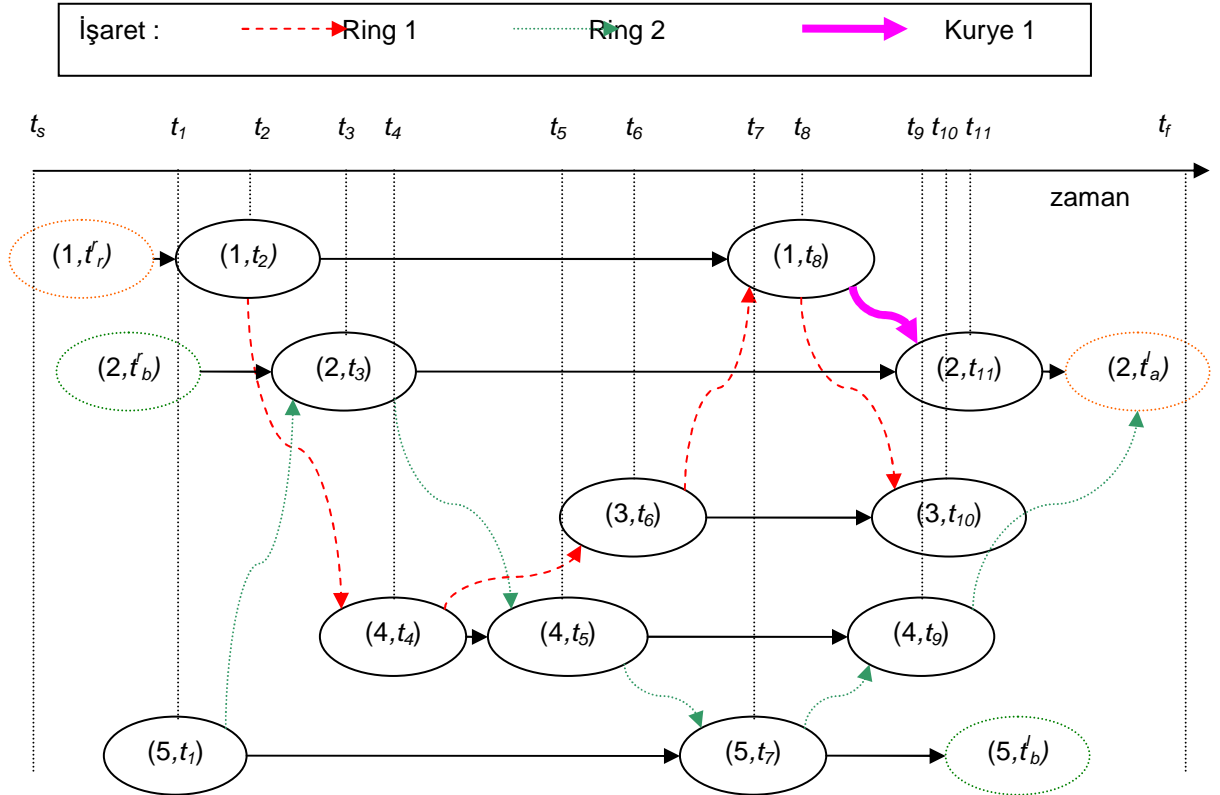
### TEKRARLI TAŞIMA PLANLAMA MODELLERİ

#### 1. TALEP KARŞILAMA TAMSAYISAL PROGRAMLAMA MODELİ

Geliştirilecek modelde, gelmiş olduğu varsayılan bir kurye / ring talepleri kümesinin,  $[t_s, t_f]$  zaman aralığında taşınmasının planlanması problemi ele alınacaktır. Söz konusu taleplerin belirli kuryelere yapılmadığı, sadece başlangıç ve bitiş lokasyonları ve talep iptal zamanı verilerek yapıldığı varsayılacaktır. Her talebin yukarıda belirtilen üç özelliğine ek olarak, yapıldığı an, uyumlu olduğu kurye / ring servisleri alt kümesi, ve  $[1,5]$  aralığında tamsayısal bir taşıma önceliği değeri olacaktır. Taşıma öncelik değeri küçük olan bir talebin önem değeri büyüktür, örneğin önceliği 1 olan bir talebin gerçekleştirilmesi önceliği 2 olan bir talebin gerçekleştirilmesine göre daha önemlidir. Bu nedenle herhangi bir önem derecesindeki bir talebin kabul edilebilmesi için, kendisinden düşük önceliğe sahip olan bütün talepler reddedilebilir ya da ötelenebilir olacağı kabul edilmiştir. Amaç fonksiyonu, taşıma önceliklerini göz önüne alacak şekilde kabul edilen taleplerin sayısını maksimuma çıkarmayı hedeflemektedir. Sistemdeki her kurye/ring seferinin aşılmaması gereken bir istiap haddi, ve gerçekleştirilmesi için her bacağında ulaşılması gereken bir kullanım oranı olacaktır. Taşıma önceliklerine göre taşınabilen taleplerin maksimizasyonunun yanı sıra, bu işlemin minimum maliyetle yapılması göz önünde bulundurulacaktır.

Yukarıda bahsedilen problem özel olarak hazırlanmış bir serim üzerinde "Çok malzemeli serim tasarımı" (multicommodity network design) probleminin bir benzeridir. Modele geçmeden önce söz konusu serimin nasıl oluşturulacağını adım adım anlatılacaktır.  $i$  bir lokasyonu  $t$  ise bir zamanı belirtmek üzere serimin düğümlerine  $(i, t)$  formatında etiketler atanır. Eğer bir kurye / ring'in bir bacağının  $i$  lokasyonundan  $t_i$  anında  $j$  lokasyonuna doğru yola çıkıp  $t_j$  anında varması planlanıyorsa, aracın yüklenme süresi  $l$  ve boşaltılma süresi  $u$  ise, ve hem  $t_i - l$  hem de  $t_j + u$  de  $[t_s, t_f]$  aralığındaysa, o zaman serime  $(i, t_i - l)$  ve  $(j, t_j + u)$  etiketli düğümler eklenir.  $Source(k)$ ,  $destination(k)$ ,  $release\_time(k)$  ve  $latest\_arrival\_time(k)$ ,  $k$  talebinin çıkış lokasyonunu, hedef lokasyonunu, talep anını ve talep iptal zamanını belirtsin. Serime, her  $k$  talebi için talebin çıkış anını ifade eden  $(source(k), \max(release\_time(k), t_s))$  etiketli bir düğüm eklenir. Benzer şekilde, serime her  $k$  talebi için talebin varış anını ifade eden  $(destination(k), \min(latest\_arrival\_time(k), t_f))$

etiketli bir düğüm eklenir. Aynı etikete sahip olan düğümlerden sadece bir tanesi bırakılıp gerisi silinir. Şu anda serimin bütün düğümleri belirlenmiş durumdadır. Müteakiben serimin ayrıtları (arc) eklenir. Eğer bir kurye/ring seferinin bir bacağının  $i$  lokasyonundan  $t_i$  anında  $j$  lokasyonuna doğru yola çıkıp  $t_j$  anında varması planlanıyorsa, aracın yüklenme süresi  $l$  ve boşaltılma süresi  $u$  ise, ve hem  $t_i - l$  hem de  $t_j + u$  de  $[t_s, t_f]$  aralığındaysa, o zaman serime  $(i, t_i - l)$  düğümünden  $(j, t_j + u)$  düğümüne giden yönlü ve karşılık geldiği kurye / ring'in kapasitesine sahip bir ayrıt eklenir. Bu ilk ayrıt kümesi taleplerin taşınmasına karşılık gelmektedir. İkinci olarak, her  $(i, t_p)$  ve  $(i, t_q)$  düğümü için  $(t_q, i$  lokasyonunda  $t_p$ 'den büyük ilk zaman etiketi iken)  $(i, t_p)$  düğümünden  $(i, t_q)$  düğümüne yönlü ve kapasitesiz bir ayrıt eklenir. Bu ikinci ayrıt kümesi taleplerin beklemlerine karşılık gelmektedir. Bahsedilen serimin bir örneği, birincisi lokasyon 1'den lokasyon 2'ye, ikincisi ise lokasyon 2'den lokasyon 5'e olmak üzere iki talebin taşınmasını gösterecek biçimde Şekil-9.1'de verilmiştir. Söz konusu şekilde, düğümlerin zaman etiketlerine verilen isimler, tarihsel sıraları ile ilintilendirilmiştir.



Şekil-E9.1: Serim örneği



## 1.1. Parametreler

|           |   |
|-----------|---|
| $N$       | : Serimin düğümlerinin oluşturduğu küme   |
| $A_1$     | : Bir kurye / ring'in bir bacağına karşılık gelen ayrıtların kümesi                                 |
| $A_2$     | : Beklemelere karşılık gelen ayrıtların kümesi  |
| $A$       | : Bütün ayrıtların kümesi   |
| $K$       | : Kurye / ring talepleri kümesi   |
| $R$       | : $[t_s, t_f]$ aralığında en az bir bacağı planlanmış kurye/ringler kümesi                          |
| $loc(i)$  | : $i$ düğümünün lokasyon etiketi  |
| $time(i)$ | : $i$ düğümünün zaman etiketi   |
| $B(i)$    | : $i$ düğümünün ardıl yıldızı   |
| $F(i)$    | : $i$ düğümünün öncül yıldızı   |
| $rc(i,j)$ | : $(i,j)$ ayrıtlarının ifade ettiği kurye / ring servisi  |
| $u_k$     | : $k$ talebinin amaç fonksiyon katsayısı  |
| $w_k$     | : $k$ talebinin ağırlığı  |
| $v_k$     | : $k$ talebinin hacmi   |
| $p_k$     | : $k$ talebi kapsamında taşınacak personel sayısı   |
| $wu_r$    | : $r$ kurye / ring'inin gerçekleştirilebilmesi için gereken minimum doluluk oranı (ağırlık bazında) |
| $vu_r$    | : $r$ kurye / ring'inin gerçekleştirilebilmesi için gereken minimum doluluk oranı (hacim bazında)   |
| $pu_r$    | : $r$ kurye / ring'inin gerçekleştirilebilmesi için gereken minimum doluluk oranı (yolcu bazında)   |
| $WCap_r$  | : $r$ kurye / ringinin ağırlık taşıma kapasitesi  |
| $VCap_r$  | : $r$ kurye / ringinin hacim taşıma kapasitesi  |
| $PCap$    | : $r$ kurye / ringinin yolcu taşıma kapasitesi  |
| $c_r$     | : $r$ kurye / ringinin maliyeti   |

$s(k)$  : Serimde zaman etiketi  $\max(\text{release\_time}(k), t_s)$  ve lokasyon etiketi  $\text{source}(k)$  olan düğüm

$d(k)$  : Serimde zaman etiketi  $\min(\text{latest\_arrival\_time}(k), t_f)$  ve lokasyon etiketi  $\text{destination}(k)$  olan düğüm

## 1.2. Karar değişkenleri

$x_k$  : 1 eğer  $k$  talebi karşılanabilecekse 0 karşılanamayacaksa

$y_{ijk}$  : 1 eğer  $k$  talebi  $(i,j) \in A$  ayrıtı kullanılarak taşınacaksa  
0 taşınmayacaksa

$z_r$  : 1 eğer  $r$  kurye / ringi gerçekleştirilecekse 0 gerçekleştirilmeyecekse

## 1.3. Model

$$\max \sum_{k \in K} u_k x_k - \sum_{r \in R} c_r z_r \quad (1)$$

s.t.

$$\sum_{a \in B(i)} y_{ak} - \sum_{a \in F(i)} y_{ak} = \begin{cases} x_k & \text{eger } i = s(k) \\ -x_k & \text{eger } i = d(k) \\ 0 & \text{diğer durumlarda} \end{cases}, \forall i \in N, k \in K \quad (2)$$

$$\sum_{k \in K} w_k y_{ak} \leq WCap_{rc(a)} z_{rc(a)}, \forall a \in A_1 \quad (3)$$

$$\sum_{k \in K} v_k y_{ak} \leq VCap_{rc(a)} z_{rc(a)}, \forall a \in A_1 \quad (4)$$

$$\sum_{k \in K} p_k y_{ak} \leq PCap_{rc(a)} z_{rc(a)}, \forall a \in A_1 \quad (5)$$

$$x, y, z \in B \quad (6)$$

## 1.4. Kısıtların anlamları

Kısıt kümesi (2) her bir talebin akış korunumunu sağlamaktadır. Kısıt kümeleri (3), (4), ve (5) kurye / ring seferlerinin kendilerine atanan talepleri taşıyabilecek kapasitede olmasını sağlamaktadır.

### 1.5. Amaç Fonksiyon Katsayılarının Belirlenmesi

Varsayalım ki  $u_{(i)}$   $i$  önceliğine sahip bir talebin amaç fonksiyon katsayısını,  $n_{(i)}$  da  $i$  önceliğine sahip kurye / ring taleplerinin sayısını ifade etsin. Herhangi bir talebin yerine ulaştırılması, maliyetten daha önemli olduğuna göre, bu olayın getirisi bütün kurye / ringlerin çalıştırılmasının maliyetinden daha yüksek olmalıdır. Taşıma önceliğinin yukarıda yapılan tanımı da göz önüne alındığında, amaç fonksiyonu katsayıları aşağıdaki eşitsizlikleri sağlamalıdır:

$$u_{(5)} > \sum_{r \in R} c_r$$

$$u_{(4)} > n_{(5)} u_{(5)}$$

$$u_{(3)} > n_{(4)} u_{(4)} + n_{(5)} u_{(5)}$$

$$u_{(2)} > n_{(3)} u_{(3)} + n_{(4)} u_{(4)} + n_{(5)} u_{(5)}$$

$$u_{(1)} > n_{(2)} u_{(2)} + n_{(3)} u_{(3)} + n_{(4)} u_{(4)} + n_{(5)} u_{(5)}$$

Bu durumda, aşağıdaki değerler istenen koşulları sağlamaktadır:

$$u_{(5)} := \sum_{r \in R} c_r + 1;$$

$$u_{(4)} := n_{(5)} u_{(5)} + 1;$$

$$u_{(3)} := n_{(4)} u_{(4)} + n_{(5)} u_{(5)} + 1;$$

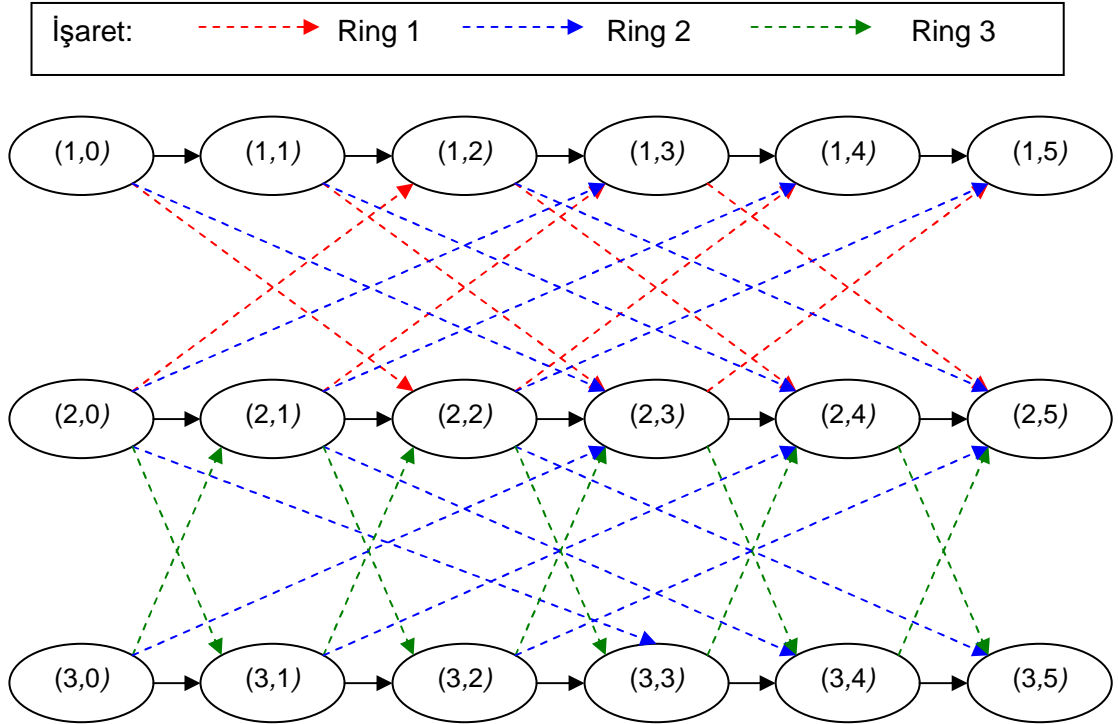
$$u_{(2)} := n_{(3)} u_{(3)} + n_{(4)} u_{(4)} + n_{(5)} u_{(5)} + 1;$$

$$u_{(1)} := n_{(2)} u_{(2)} + n_{(3)} u_{(3)} + n_{(4)} u_{(4)} + n_{(5)} u_{(5)} + 1;$$

## 2. KURPM Kurye / Ring Sistemi Tamsayısal Optimizasyon Modeli

Bu bölümde tekrarlı taşıma sisteminin maliyet etkin bir biçimde planlanabilmesi için tasarlanan bir tamsayısal programlama modeli üzerinde durulacaktır. Söz konusu servisler periyodik olduğu için, kurye / ring olarak tahsis edilmiş olan her aracın planlama süresinin bitiminde başladığı yere dönmesi gerekmektedir. "İstatistiksel bilgi" olarak tanımlanan verinin başlangıç lokasyonu, bitiş lokasyonu, talep anı, ve kıymetten düşme anı belirtilmiş olan taşıma talepleri olduğu varsayılmaktadır.

KURPM rezervasyon alt sistemi için oluşturduğumuz modele benzer şekilde, özel bir serim oluşturacağız. Fakat bu modelde rotalar ve çizelgeler sabit olmadığı için, ilgilendiğimiz zaman noktalarını sınırlı sayıya indirgememiz gerekmektedir (örneğin her saatin başlangıcı). Modele geçmeden önce söz konusu serimin nasıl oluşturulacağını adım adım anlatacağız. Serimin düğümlerine  $(i,t)$  formatında etiketler atayacağız, burada  $i$  bir lokasyonu  $t$  ise bir zamanı belirtmektedir. Her  $i$  lokasyonu ve  $[0, T]$  aralığındaki her  $t$  için, serime  $(i,t)$  düğümü eklenir. Şu anda serimin bütün düğümleri belirlenmiş durumdadır. Şimdi serimin ayrıtlarını (arc) ekleyeceğiz. Eğer  $r$  aracının  $i$  lokasyonundan çıkıp  $j$  lokasyonuna varması  $t_{ijr}$  birim zaman alıyorsa, yüklenme süresi  $l_r$  ve boşaltılma süresi  $u_r$  ise  $(i,t)$  düğümünden  $(j, t + t_{ijr} + l_r + u_r)$  düğümüne her  $t \in [0, T - t_{ijr} - l_r - u_r]$  için bir ayrıtl eklenir. Bu ayrıtlın kapasitesi,  $r$  aracının kapasitesine eşittir. Bu ilk ayrıtl kümesi araçların hareketlerini temsil etmektedir. İkinci olarak, her  $(i,t)$  düğümünden  $(i, t + 1)$  düğümüne bir ayrıtl eklenir. Bu ikinci ayrıtl kümesi araçların beklemlerini temsil etmektedir. Örnek bir serim, Şekil-9.2'de verilmiştir.



Şekil-E9.2: Üç ring için bir serim örneği

Bu model problemin en temel halini anlatmaktadır. İki olası ilgili versiyon aşağıda verilmiştir:

- Bütün talepleri karşılayacak şekilde minimum sayıda araç kullanmak
- Araç sayısını sabit tutup, taşıma önceliklerini göz önünde bulundurarak taşınan talep sayısını maksimize etmek

İki versiyon da yeni karar değişkenleri tanımlanarak kolayca modellenebilir.

## 2.1. Parametreler

|          |   |
|----------|---|
| $N$      | : Serimin düğümlerinin oluşturduğu küme   |
| $A_1$    | : Bir kurye / ring'in hareketine karşılık gelen ayrıtların kümesi   |
| $A_2$    | : Beklemelere karşılık gelen ayrıtların kümesi  |
| $A$      | : Bütün ayrıtların kümesi   |
| $K$      | : Kurye / ring talepleri kümesi   |
| $R$      | : Araç kümesi   |
| $B(i,r)$ | : $r$ aracı için $i$ düğümünün ardıl yıldızı  |
| $F(i,r)$ | : $r$ aracı için $i$ düğümünün öncül yıldızı  |
| $c_{ij}$ | : $(i,j)$ ayrıtlarının maliyeti   |
| $w_k$    | : $k$ talebinin ağırlığı  |
| $v_k$    | : $k$ talebinin hacmi   |
| $p_k$    | : $k$ talebi kapsamında taşınacak personel sayısı   |
| $WCap_r$ | : $r$ kurye / ringinin ağırlık taşıma kapasitesi  |
| $VCap_r$ | : $r$ kurye / ringinin hacim taşıma kapasitesi  |
| $PCap_r$ | : $r$ kurye / ringinin personel taşıma kapasitesi   |
| $s(k)$   | : Serimde zaman etiketi $\max(\text{release\_time}(k), t_s)$ ve lokasyon etiketi $\text{source}(k)$ olan düğüm              |
| $d(k)$   | : Serimde zaman etiketi $\min(\text{latest\_arrival\_time}(k), t_f)$ ve lokasyon etiketi $\text{destination}(k)$ olan düğüm |
| $s(r)$   | : Serimde zaman etiketi 0 ve lokasyon etiketi $\text{home\_base}(r)$ olan düğüm   |

$d(r)$  : Serimde zaman etiketi  $T$  ve lokasyon etiketi  $home\_base(r)$  olan düğüm

## 2.2. Karar değişkenleri

$y_{ak}$  :1 eğer  $k$  talebi  $a \in A$  ayrıtı kullanılarak taşınacaksa.  
0 taşınmayacaksa

$z_a^1$  :1 eğer kurye / ring  $r$  kendisine ait olan  $a \in A_1$  ayrıtını kullanacaksa 0 kullanmayacaksa

$z_{ar}^2$  :1 eğer kurye / ring  $r$   $a \in A_2$  ayrıtını kullanacaksa  
0 kullanmayacaksa

## 2.3. Model

$$\min \sum_{\substack{a \in A_1 \\ r \in R}} c_{ar} z_{ar}^2 \quad (1) \text{ s.t.}$$

$$\sum_{a \in F(i)} y_{ak} - \sum_{a \in B(i)} y_{ak} = \begin{cases} 1 & \text{eger } i = s(k) \\ -1 & \text{eger } i = d(k) \\ 0 & \text{diger durumlarda} \end{cases}, \forall i \in N, k \in K \quad (2)$$

$$\sum_{\substack{a \in F(i,r) \\ a \in A_1}} z_a^1 + \sum_{\substack{a \in F(i) \\ a \in A_2}} z_{ar}^2 - \sum_{\substack{a \in B(i,r) \\ a \in A_1}} z_a^1 - \sum_{\substack{a \in B(i) \\ a \in A_2}} z_{ar}^2 = \begin{cases} 1 & \text{eger } i = s(r) \\ -1 & \text{eger } i = d(r) \\ 0 & \text{diger durumlarda} \end{cases} \quad (3)$$

$$, \forall i \in N, r \in R$$

$$\sum_{k \in K} w_k y_{ak} \leq WCap_{ring(a)} z_a^1, \forall a \in A_1 \quad (4)$$

$$\sum_{k \in K} v_k y_{ak} \leq VCap_{ring(a)} z_a^1, \forall a \in A_1 \quad (5)$$

$$\sum_{k \in K} p_k y_{ak} \leq PCap_{ring(a)} z_a^1, \forall a \in A_1 \quad (6)$$

$$y, z^1, z^2 \in B \quad (7)$$

#### **2.4. Kısıtların Anlamları**

Kısıt kümesi (2) her bir talebin akış korunumunu sağlamaktadır. Kısıt kümesi (3) her bir aracın akış korunumunu sağlamaktadır. Kısıt kümeleri (4), (5), ve (6) kurye / ring seferlerinin kendilerine atanan talepleri taşıyabilecek kapasitede olmasını sağlamaktadır. Kısıt kümesi (7) ise değişkenleri tamsayı değerleri almaya zorlamaktadır.

**EK-10: Kavramsal ve Fiziksel İlişkisel Veri Modeli**

# **Kavramsal ve Fiziksel İlişkisel Veri Modeli**

**Versiyon .....**



## 1. AMAÇ ve KAPSAM

Sistem, Kullanıcının her durumda ortaya çıkabilecek kara yolu, demir yolu, deniz yolu, hava yolu ve iç su yolu ulaştırma ihtiyaçlarının en etkin, süratli ve ekonomik olarak karşılanmasına yönelik ulaştırma planlamalarının mevcut ulaştırma kaynaklarının, araç boş kapasitelerinin ve faaliyetlerinin tüm ulaştırma planlamacıları tarafından karşılıklı olarak zamanında görülerek, paralel ve entegre olarak yapılabilmesi, ve karar verici makamlara karar desteği sağlanması ana amacını taşımaktadır.

Proje yönetim metodolojisi olarak, sistem geliştirilmesi bakımından bu alanda dünyanın en gelişmiş metodoloji/yaklaşımlarından biri olan MSF/SDD metodolojisini uygulanacaktır. Uygulanacak MSF/SDD metodolojisi, aşama yaklaşım (phased approach) ve aşamalı-döngüsel (iterative-spiral) bir geliştirme süreci öngörmektedir. Tüm proje aşamaları, proje yaşam döngüsü boyunca sürekli olarak tekrar eden ve devam eden aktivitelerdir. Son/nihai ürün geliştirilene ve devreye alınana kadar tüm tanımlı proje aşamaları döngüsel olarak devam etmektedir. Bu yaklaşımın temel amacı; yazılımı aşamalı/döngüsel olarak geliştirmek, ve erken sürümlerle birlikte müşteri ve dolayısı ile son kullanıcının görüşlerini alarak, eski sürümlerin geliştirilmesi aşamasındaki alınan ders ve tecrübelerden de yararlanarak, yazılıma yönelik yeni sürümler gerçekleştirebilmektir. Bu yaklaşımda kullanıcıdan gelen geri bildirim ve görüşler yeni sürümlere yansıtılır. Bu döngüsel ve aşamalı geliştirme süreci, nihai ürün teslim edilene kadar bu şekilde devam eder.

Sistem Kavramsal/Fiziksel ilişkisel Veri Modelleri diyagramları, uygulanacak proje yönetim metodolojisi içinde gerçekleştirilecek (ve proje yönetim anlayışı gereğince döngüsel olarak proje süresince devam edecek) ihtiyaç tespit ve mimari tasarım çalışmaları sonucunda oluşmuş, sistemin veri modellemesine yönelik kavramsal ilişkileri özetleyen çalışmalardır. Bu doküman, ekinde sunulan veri modellemesine yönelik diyagramların, proje ekibi tarafından yorumlanmasını ve takip edilebilmesini sağlamak amacıyla bir kaynak kılavuz olarak hazırlanmıştır.

## 2. VERİ MODELLERİ DİYAGRAMI

Bu dokümanın 2.6 ve 2.7 maddelerinde sunulan, Sistem Veri Modelleri Diyagramlarına yönelik tanımlayıcı ve açıklayıcı bilgiler aşağıdaki paragraflarda sunulmuştur.

Diyagramlarda takip ve anlayış kolaylığı sağlamak üzere renk kodları da kullanılmıştır. Bu çerçevede, Yeşil Renk Kodu VTYM yapısını, Pembe Renk Kodu ETARİTM yapısını, Mavi Renk Kodu TPM yapısını ve Sarı Renk Kodu KURPM yapısını örneklemiştir.

Aşağıdaki paragraflarda her bir alt-sistem için veri modeli bileşenlerine yönelik tanım ve açıklamalar sunulmuştur.

## **2.1. VTYM (Yeşil Renk Kodu)**

VTYM alt sistemi için veri modellemesine yönelik, EK diyagramlarda da gösterilen, 58 adet veri modeli bileşeni tanımlanmıştır. Aşağıdaki paragraflarda bu bileşenlerin listesi ve tanımlamaları sunulmuştur.

### **2.1.1. Ulaşım Modeli**

Bir ulaşım modu bir veya birden fazla ulaşım ağından oluşur.

### **2.1.2. Ulaşım Ağı**

Bir ulaşım ağı bir veya birden fazla ulaşım hattından oluşur.

### **2.1.3. Ulaşım Hatları**

Bir ulaşım hattı bir başlangıç ve bir bitiş lokasyonu ile ifade edilmektedir. Bir ulaşım hattı birden fazla kurye/ring bacağına ait olabilir. Ayrıca bir ulaşım hattı birden fazla taşıma rotasına da ait olabilir.

### **2.1.4. Ulaşım Hattı Kısıtları**

Ulaşım hattına ait kısıtları ifade eder. Bir ulaşım hattının birden fazla kısıtı bulunabilir.

### **2.1.5. Harita Projeleri**

Bir harita projesi birden fazla haritadan oluşabilir. Bir harita projesi birden fazla lokasyon bilgisi içermektedir.

### **2.1.6. Harita**

Bir veya birden fazla harita bir harita projesini meydana getirmektedir.

### **2.1.7. Lokasyonlar**

Lokasyon, bir terminal, istasyon, havaalanı veya limanın konumunu, bir barış konuş yerini, bir kurye talebinin nereden nereye yapılacağını, taşıma planı

görev yerini, bir ulaşım hattının başlangıç veya bitiş noktasını belirtmekte kullanılmaktadır.

#### **2.1.8. Lokasyon Türleri**

Bir lokasyonun türünü belirtmektedir. Bir lokasyon birden fazla lokasyon türüne sahip olabilir.

#### **2.1.9. Lokasyon Yakıt Tipi**

Bir lokasyonda bulunan yakıt tiplerini ifade etmektedir. Bir lokasyonda birden fazla yakıt tipi bulunabilir.

#### **2.1.10. Yakıt Tipi**

Yakıt tiplerini ifade etmektedir. Bir lokasyonda birden fazla yakıt tipi bulunabilir.

#### **2.1.11. Ülke**

Bir ülke, kendine ait olan lokasyonlarla ifade edilmektedir. Bir ülke bir veya daha fazla lokasyon bilgisine sahiptir.

#### **2.1.12. Terminal**

Bir lokasyonda birden fazla terminal bulunabilir. Bir terminale ait birden fazla elleçleme kapasitesi bilgisi bulunabilir. Her terminalin bir terminal tipi vardır.

#### **2.1.13. Terminal Tipi**

Terminallerin tipini ifade eder. Her terminalin bir terminal tipi vardır.

#### **2.1.14. Terminal Elleçleme Kapasitesi**

Her terminal elleçleme kapasitesine ait bir lokasyon elleçleme tipi bilgisi mevcuttur.

#### **2.1.15. İstasyon**

Her istasyona ait bir lokasyon bilgisi mevcuttur. Bir lokasyonda birken fazla istasyon bulunabilir. Bir istasyona ait birden fazla elleçleme kapasitesi bilgisi bulunabilir. Her istasyonun bir tipi vardır.

#### **2.1.16. İstasyon Tipi**

İstasyonun tipini ifade eder. Her istasyonun bir tipi vardır.

### **2.1.17. İstasyon Elleçleme Kapasitesi**

Her istasyon elleçleme kapasitesine ait bir lokasyon elleçleme tipi bilgisi mevcuttur.

### **2.1.18. Liman**

Sistemde her limana ait bir lokasyon bilgisi mevcuttur. Bir lokasyonda birden fazla liman bulunabilir. Bir limana ait birden fazla elleçleme kapasitesi bilgisi bulunabilir. Her limanın bir tipi vardır.

### **2.1.19. Liman Tipi**

Limanın tipini ifade eder. Her limanın bir tipi vardır.

### **2.1.20. Liman Elleçleme Kapasitesi**

Her liman elleçleme kapasitesine ait bir lokasyon elleçleme tipi bilgisi mevcuttur.

### **2.1.21. Liman Rıhtımı**

Her limana ait en az bir liman rıhtımı bilgisi mevcuttur.

### **2.1.22. Liman Rıhtım Tipi**

Her liman rıhtımının bir tipi vardır.

### **2.1.23. Havaalanı**

Sistemde her havaalanına ait bir lokasyon bilgisi mevcuttur. Bir lokasyonda birden fazla havaalanı bulunabilir. Bir havaalanına ait birden fazla elleçleme kapasitesi bilgisi bulunabilir. Her havaalanının bir tipi vardır.

### **2.1.24. Havaalanı Tipi**

Havaalanının tipini ifade eder. Her havaalanının bir tipi vardır.

### **2.1.25. Havaalanı Elleçleme Kapasitesi**

Her havaalanı elleçleme kapasitesine ait bir lokasyon elleçleme tipi bilgisi mevcuttur.

### **2.1.26. Havaalanı Pisti**

Her havaalanının bir veya daha fazla pisti bulunabilmektedir.

### **2.1.27. Havaalanı Parkı**

Her havaalanının bir veya daha fazla parkı bulunabilmektedir.

### **2.1.28. Lokasyon Elleçleme Tipi**

Her terminal, istasyon, liman ve havaalanı elleçleme kapasitesine ait bir lokasyon elleçleme tipi bilgisi mevcuttur.

### **2.1.29. Taşınacak Birim**

Her taşınacak birimin bir konuş lokasyonu mevcuttur.

### **2.1.30. Taşınacak Birim Profili**

Her taşınacak birim için bir profil tanımlıdır. Her taşınacak birim profiline ait taşınacak birim envanteri mevcuttur.

### **2.1.31. Profil**

Her taşınacak birim hiyerarşisi bir profile aittir. Bir profilde birden fazla taşınacak birim hiyerarşisi bulunabilir.

### **2.1.32. Gizlilik**

Her profilin bir gizliliği söz konusudur. Aynı gizliliğe sahip birden fazla profil bulunabilir. Aynı şekilde her planın bir gizliliği vardır ve aynı gizliliğe sahip birden fazla plan olabilir.

### **2.1.33. Malzeme**

Her malzemenin malzeme renk, envanter tipi, hareket kabiliyeti, ülke bilgisi, taşınacak birim kategorisi ve REK kodu söz konusudur. Bir tip malzeme birden fazla talep manifestosunda, ve taşınacak birim envanterinde yer alabilir.

### **2.1.34. Taşınacak Birim Envanteri**

Taşınacak Birim Envanteri, malzemelerden oluşur. Bir veya birden fazla malzemeyi içerir. Bir taşınacak birim envanteri birden fazla araç manifestosunda yer alabilir.

### **2.1.35. Malzeme Renk**

Aynı renkte birden fazla malzeme bulunabilir.

### **2.1.36. Envanter Tipi**

Aynı envanter tipinde birden fazla malzeme bulunabilir.

### **2.1.37. Hareket Kabiliyeti**

Aynı hareket kabiliyetine sahip birden fazla malzeme bulunabilir.

### **2.1.38. Taşınacak Birim Kategorisi**

Aynı taşınacak birim kategorisine sahip birden fazla envanter bulunabilir.

### **2.1.39. Raporlanabilir Envanter Kodu (REK)**

Her malzemenin bir REK kodu mevcuttur. Aynı REK koduna sahip birden fazla malzeme olabilir. Her REK kodunun bir üst REK kodu ve REK indikatörü vardır. Her REK koduna ait bir yükleme önceliği söz konusudur.

### **2.1.40. REK Ayırıcısı (Indicator)**

Her REK kodunun bir REK ayırıcısı mevcuttur.

### **2.1.41. Yükleme Önceliği**

Her REK koduna ve araca ait bir yükleme önceliği mevcuttur.

### **2.1.42. Yükleme Öncelik Tanımı**

Yükleme öncelikleri, yükleme öncelik tanımını oluşturmaktadır. Her yükleme önceliğinin ait olduğu bir yükleme öncelik tanımı vardır.

### **2.1.43. Araç Bilgisi**

Her araç bilgisine ait bir lokasyon, ülke, araç kaynak tipi, araç tipi bilgisi söz konusudur. Ayrıca araç bilgisine ait yükleme önceliği ve iz bilgisi mevcuttur. Bir araç bilgisinin birden fazla iz bilgisi ve yükleme önceliği olabilir.

### **2.1.44. Araç Kaynak Tipi**

Her aracın bir araç kaynak tipi bilgisi mevcuttur. Aynı araç kaynak tipine sahip birden fazla araç olabilir.

### **2.1.45. Araç Tipi Bilgisi**

Her araç tipinin bir ulaştırma modu bilgisi söz konusudur. Bir araç tipi birden fazla araca ait olabilir. Bir araç tipinde birden fazla kargo bölümü bulunabilir.

### **2.1.46. Kargo Bölümü**

Her kargo bölümüne sahip olan en az bir araç mevcuttur. Bir araçta birden fazla kargo bölümü bulunabilir. Her kargo bölümünün kargo bölüm tipi ve kargo depo tipi bilgileri mevcuttur.

### **2.1.47. Kargo Bölüm Tipi**

Bir kargo bölüm tipi bir veya birden fazla kargo bölümüne ait olabilir.

#### **2.1.48. Kargo Depo Tipi**

Bir kargo depo tipi bir veya birden fazla kargo bölümüne ait olabilir.

#### **2.1.49. Denizyolu Aracı**

Her denizyolu aracının bir araç tipi bilgisi mevcuttur. Ayrıca deniz araç tipi, deniz araç rampa tipi, deniz araç kategorisi ve deniz araç seyir sınıfı bilgileri mevcuttur.

#### **2.1.50. Deniz Araç Tipi**

Her denizyolu aracının bir deniz araç tipi bilgisi mevcuttur.

#### **2.1.51. Deniz Araç Rampa Tipi**

Her denizyolu aracının bir deniz araç rampa tipi bilgisi mevcuttur.

#### **2.1.52. Deniz Araç Kategorisi**

Her denizyolu aracının bir deniz araç kategorisi bilgisi mevcuttur.

#### **2.1.53. Deniz Araç Seyir Sınıfı**

Her denizyolu aracının bir deniz araç seyir sınıfı bilgisi mevcuttur.

#### **2.1.54. Karayolu Aracı**

Her karayolu aracının bir araç tipi bilgisi mevcuttur. Ayrıca her karayolu aracına ait kara araç kullanım tipi ve kara araç fonksiyon bilgileri mevcuttur.

#### **2.1.55. Kara Araç Kullanım Tipi**

Her kara araç kullanım tipi en az bir karayolu aracına aittir.

#### **2.1.56. Kara Araç Fonksiyon**

Her kara araç fonksiyon bilgisi en az bir karayolu aracına aittir.

#### **2.1.57. Demiryolu Aracı**

Her demiryolu aracının bir araç tipi bilgisi mevcuttur.

#### **2.1.58. Demiryolu Vagon Tipi**

Her demiryolu aracının, demiryolu vagon tipi bilgisi mevcuttur.

## **2.2. ETARİTM (Pembe Renk Kodu)**

ETARİTM alt sistemi için veri modellemesine yönelik, 2.6. ve 2.7.'de sunulan diyagramlarda da pembe renk kodu ile gösterilen İz Bilgisi veri modeli bileşeni tanımlanmıştır.

### **2.2.1. İz Bilgisi**

Her iz bilgisi bir araca aittir. Bir aracın birden fazla iz bilgisi bulunabilir.

## **2.3. TPM (Mavi Renk Kodu)**

TPM alt sistemi için veri modellemesine yönelik, EK diyagramlarda da gösterilen, 7 adet veri modeli bileşeni tanımlanmıştır. Aşağıdaki paragraflarda bu bileşenlerin listesi ve tanımlamaları sunulmuştur.

### **2.3.1. Taşınacak Birim Yer Değiştirme**

Taşınacak birimlerin birden fazla taşınacak birim yer değiştirme bilgisi bulunabilir. Ayrıca her ihtiyaç bilgisinin birden fazla taşınacak birim yer değiştirme bilgisi bulunabilir.

### **2.3.2. Taşıma Rota**

Her taşınacak birim yer değiştirmesine ait en az bir taşıma rota bilgisi mevcuttur. Her taşıma rotası bir ulaşım hattına aittir.

### **2.3.3. Araç Manifesto**

Her araç manifestosuna ait en az bir taşınacak birim envanteri ve araç bilgisi mevcuttur.

### **2.3.4. Plan**

Her planın bir üst plan bilgisi mevcuttur. Her planın bir plan tipi bilgisi vardır. Her planın en az bir ihtiyaç bilgisi söz konusudur. Ayrıca her planın bir gizlilik bilgisi mevcuttur.

### **2.3.5. Plan Tipi**

Bir plan tipine sahip birden fazla plan olabilir.

### **2.3.6. İhtiyaç**

Her plana ait en az bir taşınma ihtiyacı söz konusudur. Her ihtiyaç bilgisine ait en az bir taşınacak birim yer değiştirme bilgisi vardır. Her ihtiyaca ilişkin bir lokasyon bilgisi mevcuttur.



## **2.4. KURPM (Sarı Renk Kodu)**

KURPM alt sistemi için veri modellemesine yönelik, EK diyagramlarda da gösterilen, 12 adet veri modeli bileşeni tanımlanmıştır. Aşağıdaki paragraflarda bu bileşenlerin listesi ve tanımlamaları sunulmuştur.

### **2.4.1. Talep**

Her talebin bir talep tipi, talep nedeni, yükleme öncelik tanımı, taşınacak birim ve lokasyon bilgileri mevcuttur. Her talep manifestosu en az bir talebe aittir.

### **2.4.2. Talep Nedeni**

Her talep nedeni en az bir talebe aittir.

### **2.4.3. Talep Tipi**

Her talep tipi en az bir talebe aittir.

### **2.4.4. Talep Manifesto**

Her talep manifestosuna ait bir talep, yolcu bilgisi ve malzeme bilgisi mevcuttur. Her talep manifestosunun en az bir rezervasyon bilgisi vardır.

### **2.4.5. Yolcu**

Her yolcu bilgisi en az bir talep manifestosuna aittir.

### **2.4.6. Rezervasyon**

Her rezervasyonun bir talep manifesto ve sefer bilgisi mevcuttur. Bir sefere ve talep manifestosuna ait birden fazla rezervasyon olabilir.

### **2.4.7. Kurye - Ring Bacak**

Her kurye – ring bacağı bir ulaşım hattına aittir. Bir kurye – ring bacak bilgisine ait birden fazla sefer bilgisi olabilir. Bir kurye – ringe ait birden fazla bacak bilgisi de söz konusudur.

### **2.4.8. Sefer**

Bir sefere ait birden fazla rezervasyon bilgisi bulunabilir. Her sefer bir kurye/ring bacağına aittir.

### **2.4.9. Kurye – Ring**

Bir kurye/ringde birden fazla bacak bilgisi bulunabilir. Her kurye/ringin bir kurye/ring tipi bilgisi ve en az bir kurye/ring sefer kodu bilgisi mevcuttur.

#### **2.4.10. Kurye – Ring Tipi**

Aynı kurye/ ring tipine sahip birden fazla kurye/ring bulunabilir.

#### **2.4.11. Kurye – Ring Sefer Kodu**

Her kurye/ringin en az bir kurye/ring sefer kodu bilgisi mevcuttur. Her sefer koduna ait en az bir sefer ve sefer aracı bilgisi vardır.

#### **2.4.12. Sefer Aracı**

Her sefer aracına ait bir araç bilgisi söz konusudur.

### **2.5. Dinamik Veri Yapılandırma (Beyaz Renk Kodu)**

LAHİKA-1 ve LAHİKA-2'de verilen diyagramlarda Beyaz Renk Kodu ile gösterilen 5 adet veri modeli bileşeni, sistemde dinamik değişkenler tanımlaması yapılabilmeye yönelik öngörülen bileşenleri listelemiştir. Aşağıdaki paragraflarda bu bileşenlerin listesi ve tanımlamaları sunulmuştur.

#### **2.5.1. Özellik Tanım**

Her varlığın ve sözlük tipinin en az bir özellik tanımı mevcuttur.

#### **2.5.2. Varlık Tipi**

Sistemdeki her varlığın bir üst varlık tipi vardır. Her varlığın en az bir özellik tanımı mevcuttur.

#### **2.5.3. Özellik Değeri**

Söz konusu özelliğe ilişkin değeri ifade eder. Bir özellik tanımında birden fazla özellik değeri bulunabilir.

#### **2.5.4. Sözlük Tipi**

Her sözlük tipine ait en az bir özellik tanımı mevcuttur. Her sözlük tipine ait en az bir sözlük değeri vardır.

#### **2.5.5. Sözlük Değeri**

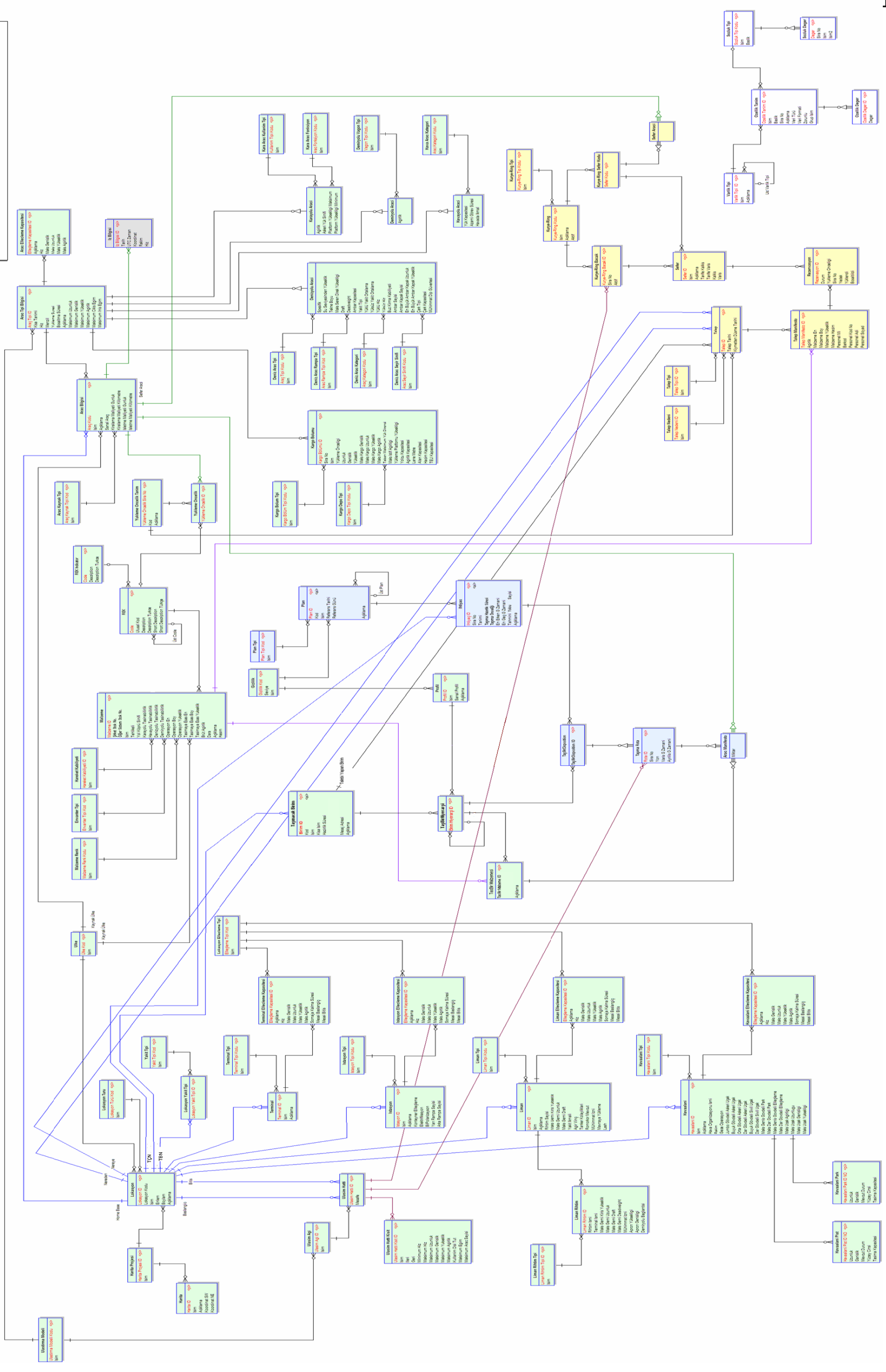
Her sözlük tipine ait en az bir sözlük değeri vardır.

**LAHİKA-1: SİSTEM FİZİKSEL İLİŞKİSEL VERİ MODELİ**

**LAHİKA-2: SİSTEM KAVRAMSAL İLİŞKİSEL VERİ MODELİ**



# SİSTEM KAVRAMSAL İLİŞKİSEL VERİ MODELİ



EK-11: Sistem Veri Değişimi

| ENTITY                     | Adı                              | Kodu                             | Veri Tipi                 | Zorunlu | ŞİRKET BİLGİ SİSTEM VERİ TABANI | ŞİRKET LOJİSTİK BİLGİ SİSTEMİ | ŞİRKET OPERASYONEL BİLGİ SİSTEMİ | ŞİRKETİN BAĞLI OLDUĞU ULUSLARARASI SİSTEMLER |
|----------------------------|----------------------------------|----------------------------------|---------------------------|---------|---------------------------------|-------------------------------|----------------------------------|--|
| Araç Bilgisi               | Araç Kodu                        | ARAC_KODU                        | Variable characters (30)  | Evet    | -                               | -                             | +                                | -  |
|                            | İsim                             | ISIM                             | Variable characters (100) | Evet    | -                               | -                             | +                                | -  |
|                            | Açıklama                         | ACIKLAMA                         | Variable characters (500) | Hayır   | -                               | -                             | +                                | -  |
|                            | Sanal Araç                       | SANAL_ARAC                       | Boolean                   | Hayır   | -                               | -                             | -                                | -  |
|                            | Kiralama Maliyeti Gunluk         | KIRALAMA_MALİYETI_GUNLUK         | Integer                   | Hayır   | -                               | -                             | +                                | -  |
|                            | Kiralama Maliyeti Kilometre      | KIRALAMA_MALİYETI_KILOMETRE      | Integer                   | Hayır   | -                               | -                             | -                                | -  |
|                            | İsletme Maliyeti Gunluk          | ISLETME_MALİYETI_GUNLUK          | Integer                   | Hayır   | -                               | -                             | +                                | -  |
|                            | İsletme Maliyeti Kilometre       | ISLETME_MALİYETI_KILOMETRE       | Integer                   | Hayır   | -                               | -                             | +                                | -  |
|                            | NSN Kodu                         | NSN_KODU                         | Characters (13)           | Evet    | -                               | -                             | +                                | -  |
| Araç Elleçleme Kapasitesi  | Elleçleme Kapasitesi ID          | ELLECLEME_KAPASITESI_ID          | Characters (36)           | Evet    | -                               | -                             | -                                | -  |
|                            | Açıklama                         | ACIKLAMA                         | Variable characters (500) | Hayır   | -                               | -                             | -                                | -  |
|                            | Hız                              | HIZ                              | Integer                   | Hayır   | -                               | -                             | -                                | -  |
|                            | Maks Genislik                    | MAKS_GENISLIK                    | Integer                   | Hayır   | -                               | -                             | -                                | -  |
|                            | Maks Uzunluk                     | MAKS_UZUNLUK                     | Integer                   | Hayır   | -                               | -                             | -                                | -  |
|                            | Maks Yükseklik                   | MAKS_YUKSEKLIK                   | Integer                   | Hayır   | -                               | -                             | -                                | -  |
|                            | Maks Agirlik                     | MAKS_AGIRLIK                     | Integer                   | Hayır   | -                               | -                             | -                                | -  |
| Araç Kaynak Tipi           | Araç Kaynak Tipi Kod             | ARAC_KAYNAK_TIPI_KOD             | Variable characters (30)  | Evet    | -                               | +                             | -                                | -  |
|                            | İsim                             | ISIM                             | Variable characters (100) | Evet    | -                               | +                             | +                                | -  |
| Araç Manifesto             | Miktar                           | MIKTAR                           | Integer                   | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Araç Tipi Bilgisi          | Araç Tipi ID                     | ARAC_TIPI_ID                     | Characters (36)           | Evet    | ?                               | ŞÖK                           | +                                | -  |
|                            | Kısa Tanımı                      | KISA_TANIMI                      | Variable characters (100) | Evet    | ?                               | ŞÖK                           | +                                | -  |
|                            | Hız                              | HIZ                              | Integer                   | Hayır   | ?                               | ŞÖK                           | +                                | -  |
|                            | Menzil                           | MENZIL                           | Integer                   | Hayır   | ?                               | ŞÖK                           | +                                | -  |
|                            | Yukleme Suresi                   | YUKLEME_SURESI                   | Integer                   | Evet    | ?                               | ŞÖK                           | -                                | -  |
|                            | Bosaltma Süresi                  | BOSALTIMA_SURESI                 | Integer                   | Evet    | ?                               | ŞÖK                           | -                                | -  |
|                            | Açıklama                         | ACIKLAMA                         | Variable characters (500) | Hayır   | ?                               | ŞÖK                           | +                                | -  |
|                            | Maksimum Uzunluk                 | MAKSIMUM_UZUNLUK                 | Integer                   | Hayır   | ?                               | ŞÖK                           | +                                | -  |
|                            | Maksimum Genislik                | MAKSIMUM_GENISLIK                | Integer                   | Hayır   | ?                               | ŞÖK                           | +                                | -  |
|                            | Maksimum Yükseklik               | MAKSIMUM_YUKSEKLIK               | Integer                   | Hayır   | ?                               | ŞÖK                           | +                                | -  |
|                            | Maksimum Agirlik                 | MAKSIMUM_AGIRLIK                 | Integer                   | Hayır   | ?                               | ŞÖK                           | +                                | -  |
|                            | Maksimum Cikis Egim              | MAKSIMUM_CIKIS_EGIM              | Integer                   | Hayır   | ?                               | ŞÖK                           | -                                | -  |
|                            | Maksimum Inis Egim               | MAKSIMUM_INIS_EGIM               | Integer                   | Hayır   | ?                               | ŞÖK                           | -                                | -  |
| Araç Tasinabilirlik        | Araç Tasinabilirlik ID           | ARAC_TASINABILIRLIK_ID           | Characters (36)           | Evet    | -                               | -                             | +                                | -  |
|                            | Tasinan Araç ID                  | TASINAN_ARAC_ID                  | Characters (36)           | Evet    | -                               | -                             | -                                | -  |
|                            | Tasiyan Araç ID                  | TASIYAN_ARAC_ID                  | Characters (36)           | Evet    | -                               | -                             | -                                | -  |
| Tasinacak Birim            | Tasinacak Birim ID               | TASINACAK_BIRIM_ID               | Characters (36)           | Evet    | +                               | +                             | +                                | -  |
|                            | Kod                              | KOD                              | Variable characters (30)  | Evet    | +                               | +                             | +                                | -  |
|                            | İsim                             | ISIM                             | Variable characters (100) | Evet    | +                               | +                             | +                                | -  |
|                            | Kısa İsim                        | KISA_ISIM                        | Variable characters (20)  | Evet    | +                               | +                             | +                                | -  |
|                            | Hazırlık Süresi                  | HAZIRLIK_SURESI                  | Integer                   | Evet    | -                               | -                             | +                                | -  |
|                            | Malzeme Hazırlık Oranı           | MALZEME_HAZIRLIK_ORANI           | Byte                      | Hayır   | -                               | -                             | -                                | -  |
|                            | Mesaj Adresi                     | MESAJ_ADRESI                     | Variable characters (100) | Evet    | +                               | -                             | +                                | -  |
|                            | Açıklama                         | ACIKLAMA                         | Variable characters (500) | Hayır   | +                               | -                             | +                                | -  |
| Tasinacak Birim Hiyerarşi  | Tasinacak Birim Hiyerarşi ID     | TASINACAK_BIRIM_HIYERARSI_ID     | Characters (36)           | Evet    | +                               | +                             | +                                | -  |
| Tasinacak Birim Kategorisi | Kategori Kod                     | KATEGORI_KOD                     | Characters (5)            | Evet    | ?                               | -                             | +                                | -  |
|                            | İsim                             | ISIM                             | Variable characters (100) | Evet    | ?                               | -                             | +                                | -  |
| Tasinacak BirimKonularına  | Taşınacak Birim Yerdeğiştirme ID | TASINACAK_BIRIM_YERDEGISTIRME_ID | Characters (36)           | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Tasinacak Birim Envanteri  | Taşınacak Birim Malzeme ID       | TASINACAK_BIRIM_MALZEME_ID       | Characters (36)           | Evet    | ?                               | +                             | +                                | -  |

EK-11: Sistem Veri Değişimi

| ENTITY                  | Adı                            | Kodu                           | Veri Tipi                 | Zorunlu | ŞİRKET BİLGİ SİSTEM VERİ TABANI | ŞİRKET CALS | ŞİRKET LOJİSTİK BİLGİ SİSTEMİ | ŞİRKET OPERASYONEL BİLGİ SİSTEMİ | ŞİRKETİN BAĞLI OLDUĞU ULUSLARARASI SİSTEMLER |
|-------------------------|--------------------------------|--------------------------------|---------------------------|---------|---------------------------------|-------------|-------------------------------|----------------------------------|--|
|                         | Kadro Miktarı                  | KADRO_MIKTARI                  | Integer                   | Evet    | ?                               | -           | +                             | +                                | -  |
|                         | Eldeki Miktar                  | ELDEKI_MIKTAR                  | Integer                   | Evet    | ?                               | -           | +                             | +                                | -  |
|                         | Açıklama                       | ACIKLAMA                       | Variable characters (500) | Hayır   | ?                               | -           | +                             | +                                | -  |
| Tasinacak Birim Seviye  | Seviye Kod                     | SEVIYE_KOD                     | Characters (5)            | Evet    | +                               | -           | +                             | +                                | -  |
|                         | İsim                           | ISIM                           | Variable characters (100) | Evet    | +                               | -           | +                             | +                                | -  |
| Tasinacak Birim Statüsü | Tasinacak Birim Statüsü Kodu   | TASINACAK_BIRIM_STATUSU_KOD    | Characters (5)            | Evet    | ?                               | -           | +                             | +                                | -  |
|                         | İsim                           | ISIM                           | Variable characters (100) | Evet    | ?                               | -           | +                             | +                                | -  |
| Demiryolu Aracı         | Ağırlık                        | AGIRLIK                        | Integer                   | No      | -                               | -           | -                             | -                                | -  |
| Demiryolu Vagon Tipi    | Vagon Tipi Kodu                | VAGON_TIPI_KODU                | Variable characters (30)  | Evet    | -                               | -           | -                             | -                                | -  |
|                         | İsim                           | ISIM                           | Variable characters (100) | Evet    | -                               | -           | -                             | -                                | -  |
| Deniz Arac Kategorisi   | Araç Kategorisi Kodu           | ARAC_KATEGORI_KODU             | Variable characters (30)  | Evet    | -                               | -           | -                             | -                                | -  |
|                         | İsim                           | ISIM                           | Variable characters (100) | Evet    | -                               | -           | -                             | -                                | -  |
| Deniz Arac Rampa Tipi   | Araç Rampa Tipi Kod            | ARAC_RAMPA_TIPI_KOD            | Characters (5)            | Evet    | -                               | -           | -                             | -                                | -  |
|                         | İsim                           | ISIM                           | Variable characters (100) | Evet    | -                               | -           | -                             | -                                | -  |
| Deniz Arac Seyir Sinifi | Araç Seyir Sinifi Kodu         | ARAC_SEYIR_SINIFI_KODU         | Variable characters (30)  | Evet    | -                               | -           | -                             | -                                | -  |
|                         | İsim                           | ISIM                           | Variable characters (100) | Evet    | -                               | -           | -                             | -                                | -  |
| Deniz Arac Tipi         | Araç Tipi Kodu                 | ARAC_TIPI_KODU                 | Variable characters (30)  | Evet    | -                               | -           | -                             | -                                | -  |
|                         | İsim                           | ISIM                           | Variable characters (100) | Evet    | -                               | -           | -                             | -                                | -  |
| Denizyolu Aracı         | Spesifik                       | SPEŞİFİK                       | Boolean                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Su Seviyesinden Yükseklik      | SU_SEVIYESINDEN_YUKSEKLIK      | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Tekne Boyu                     | TEKNE_BOYU                     | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Maks Seren Direk Yüksekliği    | MAKS_SEREN_DIREK_YUKSEKLIĞI    | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Draft                          | DRAFT                          | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Deadweight                     | DEADWEIGHT                     | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Ambar Kapasitesi               | AMBAR_KAPASITESI               | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Yakit Tipi                     | YAKIT_TIPI                     | Variable characters (30)  | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Yüklü Yakıt Ortalama           | YUKLU_YAKIT_ORTALAMA           | Decimal (6,1)             | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Yüksüz Yakıt Ortalama          | YUKSUZ_YAKIT_ORTALAMA          | Decimal (6,1)             | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Yüklü Hiz                      | YUKLU_HIZ                      | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Yüksüz Hiz                     | YUKSUZ_HIZ                     | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Buz Kırma Kabiliyeti           | BUZ_KIRMA_KABILIYETI           | Boolean                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Ambar Sayısı                   | AMBAR_SAYISI                   | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Ambar Kapak Sayısı             | AMBAR_KAPAK_SAYISI             | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | En Büyük Ambar Kapak Uzunluk   | EN_BUYUK_AMBAR_KAPAK_UZUNLUK   | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | En Büyük Ambar Kapak Yükseklik | EN_BUYUK_AMBAR_KAPAK_YUKSEKLIK | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Çark Tipi                      | CARK_TIPI                      | Variable characters (30)  | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Çark Kapasitesi                | CARK_KAPASITESI                | Decimal (6,1)             | Hayır   | -                               | -           | -                             | -                                | -  |
|                         | Mühimmat Dip Güvertesi         | MUHIMMAT_DIP_GUVERTESI         | Boolean                   | Hayır   | -                               | -           | -                             | -                                | -  |
| Envanter Tipi           | Envanter Tipi Kod              | ENVANTER_TIPI_KOD              | Characters (5)            | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|                         | İsim                           | ISIM                           | Variable characters (100) | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
| Gizlilik                | Gizlilik Kod                   | GIZLILIK_KOD                   | Characters (5)            | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|                         | Seviye                         | SEVIYE                         | Integer                   | Hayır   | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|                         | İsim                           | ISIM                           | Variable characters (100) | Hayır   | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
| Hareket Kabiliyeti      | Hareket Kabiliyeti ID          | HAREKET_KABILIYETI_ID          | Characters (5)            | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|                         | İsim                           | ISIM                           | Variable characters (100) | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
| Harita                  | Harita ID                      | HARITA_ID                      | Characters (36)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                         | İsim                           | ISIM                           | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |

EK-11: Sistem Veri Değişimi

| ENTITY                         | Adı                          | Kodu                         | Veri Tipi                 | Zorunlu | ŞİRKET BİLGİ SİSTEM VERİ TABANI | ŞİRKET CALS | ŞİRKET LOJİSTİK BİLGİ SİSTEMİ | ŞİRKET OPERASYONEL BİLGİ SİSTEMİ | ŞİRKETİN BAĞLI OLDUĞU ULUSLARARASI SİSTEMLER |
|--------------------------------|------------------------------|------------------------------|---------------------------|---------|---------------------------------|-------------|-------------------------------|----------------------------------|--|
|                                | Acıklama                     | ACIKLAMA                     | Variable characters (500) | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                                | Koordinat SW                 | KOORDINAT_SW                 | Characters (15)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                                | Koordinat NE                 | KOORDINAT_NE                 | Characters (15)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Harita Projesi                 | Harita Projesi ID            | HARITA_PROJESI_ID            | Characters (36)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                                | İsim                         | ISIM                         | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Hava Arac Kategorisi           | Arac Kategorisi Kodu         | ARAC_KATEGORI_KODU           | Variable characters (30)  | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|                                | İsim                         | ISIM                         | Variable characters (100) | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
| Havaalanı                      | Havaalanı ID                 | HAVAALANI_ID                 | Characters (36)           | Evet    | ?                               | -           | -                             | -                                | -  |
|                                | İsim                         | ISIM                         | Variable characters (100) | Evet    | ?                               | -           | -                             | +                                | -  |
|                                | Acıklama                     | ACIKLAMA                     | Variable characters (500) | Hayır   | ?                               | -           | -                             | -                                | -  |
|                                | Hava Organizasyonu İsmi      | HAVA_ORGANIZASYONU_ISMI      | Variable characters (100) | Hayır   | ?                               | -           | -                             | -                                | -  |
|                                | Rakım                        | RAKIM                        | Integer                   | Hayır   | ?                               | -           | -                             | +                                | -  |
|                                | Gece Operasyon               | GECE_OPERASYON               | Boolean                   | Hayır   | ?                               | -           | -                             | +                                | -  |
|                                | Jumbo Gövdeli Askeri Uçak    | JUMBO_GOVDELI_ASKERI_UCAK    | Boolean                   | Hayır   | ?                               | -           | -                             | ?                                | -  |
|                                | Buyuk Gövdeli Askeri Uçak    | BUYUK_GOVDELI_ASKERI_UCAK    | Boolean                   | Hayır   | ?                               | -           | -                             | ?                                | -  |
|                                | Orta Gövdeli Askeri Uçak     | ORTA_GOVDELI_ASKERI_UCAK     | Boolean                   | Hayır   | ?                               | -           | -                             | ?                                | -  |
|                                | Dar Gövdeli Askeri Uçak      | DAR_GOVDELI_ASKERI_UCAK      | Boolean                   | Hayır   | ?                               | -           | -                             | ?                                | -  |
|                                | Buyuk Gövdeli Sivil Uçak     | BUYUK_GOVDELI_SIVIL_UCAK     | Boolean                   | Hayır   | ?                               | -           | -                             | ?                                | -  |
|                                | Dar Gövdeli Sivil Uçak       | DAR_GOVDELI_SIVIL_UCAK       | Boolean                   | Hayır   | ?                               | -           | -                             | ?                                | -  |
|                                | Maks Genis Govdeli Park      | MAKS_GENIS_GOVDELI_PARK      | Integer                   | Hayır   | ?                               | -           | -                             | ?                                | -  |
|                                | Maks Dar Govdeli Park        | MAKS_DAR_GOVDELI_PARK        | Integer                   | Hayır   | ?                               | -           | -                             | ?                                | -  |
|                                | Maks Genis Govdeli Elleçleme | MAKS_GENIS_GOVDELI_ELLECLEME | Integer                   | Hayır   | ?                               | -           | -                             | ?                                | -  |
|                                | Maks Dar Govdeli Elleçleme   | MAKS_DAR_GOVDELI_ELLECLEME   | Integer                   | Hayır   | ?                               | -           | -                             | ?                                | -  |
|                                | Maks Uçak Ağırliği           | MAKS_UCAK_AGIRLIGI           | Integer                   | Hayır   | ?                               | -           | -                             | ?                                | -  |
|                                | Maks Uçak Uzunlugu           | MAKS_UCAK_UZUNLUGU           | Integer                   | Hayır   | ?                               | -           | -                             | +                                | -  |
|                                | Maks Uçak Genisliği          | MAKS_UCAK_GENISLIGI          | Integer                   | Hayır   | ?                               | -           | -                             | +                                | -  |
|                                | Maks Uçak Yüksekliği         | MAKS_UCAK_YUKSEKLIGI         | Integer                   | Hayır   | ?                               | -           | -                             | ?                                | -  |
| Havaalanı Elleçleme Kapasitesi | Elleçleme Kapasitesi ID      | ELLECLEME_KAPASITESI_ID      | Characters (36)           | Evet    | -                               | -           | -                             | -                                | -  |
|                                | Acıklama                     | ACIKLAMA                     | Variable characters (500) | Hayır   | -                               | -           | -                             | -                                | -  |
|                                | Hız                          | HIZ                          | Integer                   | Hayır   | -                               | -           | -                             | +                                | -  |
|                                | Maks Genislik                | MAKS_GENISLIK                | Integer                   | Hayır   | -                               | -           | -                             | +                                | -  |
|                                | Maks Uzunluk                 | MAKS_UZUNLUK                 | Integer                   | Hayır   | -                               | -           | -                             | +                                | -  |
|                                | Maks Yükseklik               | MAKS_YUKSEKLIK               | Integer                   | Hayır   | -                               | -           | -                             | +                                | -  |
|                                | Maks Ağırlik                 | MAKS_AGIRLIK                 | Integer                   | Hayır   | -                               | -           | -                             | +                                | -  |
|                                | Somaja Kalma Süresi          | SOMAJA_KALMA_SURESI          | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|                                | Mesai Baslangıç              | MESAI_BASLANGIC              | Time                      | Hayır   | -                               | -           | -                             | +                                | -  |
|                                | Mesai Bitis                  | MESAI_BITIS                  | Time                      | Hayır   | -                               | -           | -                             | +                                | -  |
| Havaalanı Park                 | Havaalanı Park ID No         | HAVAALANI_PARK_ID_NO         | Variable characters (30)  | Evet    | -                               | -           | -                             | -                                | -  |
|                                | Uzunluk                      | UZUNLUK                      | Integer                   | Hayır   | -                               | -           | -                             | +                                | -  |
|                                | Genislik                     | GENISLIK                     | Integer                   | Hayır   | -                               | -           | -                             | +                                | -  |
|                                | Mevcut Durum                 | MEVCUT_DURUM                 | Variable characters (50)  | Hayır   | -                               | -           | -                             | +                                | -  |
|                                | Yüzey Cinsi                  | YUZEY_CINSI                  | Variable characters (50)  | Hayır   | -                               | -           | -                             | +                                | -  |
|                                | Tasima Kapasitesi            | TASIMA_KAPASITESI            | Integer                   | Hayır   | -                               | -           | -                             | +                                | -  |
| Havaalanı Pist                 | Havaalanı Pist ID No.        | HAVAALANI_PIST_ID_NO         | Variable characters (30)  | Evet    | -                               | -           | -                             | -                                | -  |
|                                | Uzunluk                      | UZUNLUK                      | Integer                   | Hayır   | -                               | -           | -                             | +                                | -  |
|                                | Genislik                     | GENISLIK                     | Integer                   | Hayır   | -                               | -           | -                             | +                                | -  |
|                                | Mevcut Durum                 | MEVCUT_DURUM                 | Variable characters (50)  | Hayır   | -                               | -           | -                             | +                                | -  |
|                                | Yüzey Cinsi                  | YUZEY_CINSI                  | Variable characters (50)  | Hayır   | -                               | -           | -                             | +                                | -  |

EK-11: Sistem Veri Değişimi

| ENTITY                        | Adı                    | Kodu                    | Veri Tipi                 | Zorunlu | ŞİRKET BİLGİ SİSTEM VERİ TABANI | ŞİRKET CALS | ŞİRKET LOJİSTİK BİLGİ SİSTEMİ | ŞİRKET OPERASYONEL BİLGİ SİSTEMİ | ŞİRKETİN BAĞLI OLDUĞU ULUSLARARASI SİSTEMLER |
|-------------------------------|------------------------|-------------------------|---------------------------|---------|---------------------------------|-------------|-------------------------------|----------------------------------|--|
|                               | Tasima Kapasitesi      | TASIMA_KAPASITESI       | Integer                   | Hayır   | -                               | -           | -                             | +                                | -  |
| Havaalani Tipi                | Havaalani Tipi Kodu    | HAVAALANI_TIPI_KODU     | Characters (36)           | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|                               | İsim                   | ISIM                    | Variable characters (100) | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
| Havayolu Araci                | Yük Kapasitesi         | YUK_KAPASITESI          | Integer                   | Hayır   | -                               | -           | -                             | +                                | -  |
|                               | Azami Görev Süresi     | AZAMI_GOREV_SURESI      | Integer                   | Evet    | -                               | -           | -                             | +                                | -  |
|                               | Havada İkmal           | HAVADA_IKMAL            | Boolean                   | Hayır   | -                               | -           | -                             | +                                | -  |
| İhtiyac                       | İhtiyaç ID             | IHTIYAC_ID              | Characters (36)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Sıra No                | SIRA_NO                 | Integer                   | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Tanımı                 | TANIMI                  | Variable characters (50)  | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Taşıma Hazırlık Süresi | TASIMA_HAZIRLIK_SURESI  | Integer                   | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Taşıma Önceliği        | TASIMA_ONCELIGI         | Integer                   | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | En Erken G Zamani      | EN_ERKEN_G_ZAMANI       | Integer                   | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | En Geç G Zamani        | EN_GEC_G_ZAMANI         | Integer                   | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Tahmini Yolcu Sayisi   | TAHMINI_YOLCU_SAYISI    | Integer                   | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Açıklama               | ACIKLAMA                | Variable characters (500) | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Tasima Rota                   | Rota ID                | ROTA_ID                 | Characters (36)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Sıra No                | SIRA_NO                 | Integer                   | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Yon                    | YON                     | Integer                   | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Varis G Zamani         | VARIS_G_ZAMANI          | Integer                   | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Ayrılis G Zamani       | AYRILIS_G_ZAMANI        | Integer                   | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| İstasyon                      | İstasyon ID            | ISTASYON_ID             | Characters (36)           | Evet    | ?                               | -           | -                             | -                                | -  |
|                               | İsim                   | ISIM                    | Variable characters (100) | Evet    | ?                               | -           | -                             | -                                | -  |
|                               | Açıklama               | ACIKLAMA                | Variable characters (500) | Hayır   | ?                               | -           | -                             | -                                | -  |
|                               | Konteyner Eleçleme     | KONTEYNER_ELLECLEME     | Boolean                   | Hayır   | ?                               | -           | -                             | -                                | -  |
|                               | Elektrifikasyon        | ELEKTRIFIKASYON         | Boolean                   | Hayır   | ?                               | -           | -                             | -                                | -  |
|                               | Bipolarizasyon         | BIPOLARIZAYSON          | Boolean                   | Hayır   | ?                               | -           | -                             | -                                | -  |
|                               | Yan Rampa Sayisi       | YAN_RAMPA_SAYISI        | Integer                   | Hayır   | ?                               | -           | -                             | -                                | -  |
|                               | Arka Rampa Sayisi      | ARKA_RAMPA_SAYISI       | Integer                   | Hayır   | ?                               | -           | -                             | -                                | -  |
| İstasyon Elleçleme Kapasitesi | Eleçleme Kapasitesi ID | ELLECLEME_KAPASITESI_ID | Characters (36)           | Evet    | ?                               | -           | -                             | -                                | -  |
|                               | Açıklama               | ACIKLAMA                | Variable characters (500) | Hayır   | ?                               | -           | -                             | -                                | -  |
|                               | Hiz                    | HIZ                     | Integer                   | Hayır   | ?                               | -           | -                             | -                                | -  |
|                               | Maks Genislik          | MAKS_GENISLIK           | Integer                   | Hayır   | ?                               | -           | -                             | -                                | -  |
|                               | Maks Uzunluk           | MAKS_UZUNLUK            | Integer                   | Hayır   | ?                               | -           | -                             | -                                | -  |
|                               | Maks Yükseklik         | MAKS_YUKSEKLIK          | Integer                   | Hayır   | ?                               | -           | -                             | -                                | -  |
|                               | Maks Agirlik           | MAKS_AGIRLIK            | Integer                   | Hayır   | ?                               | -           | -                             | -                                | -  |
|                               | Somaja Kalma Süresi    | SOMAJA_KALMA_SURESI     | Integer                   | Hayır   | ?                               | -           | -                             | -                                | -  |
|                               | Mesai Baslangiç        | MESAI_BASLANGIC         | Time                      | Hayır   | ?                               | -           | -                             | -                                | -  |
|                               | Mesai Bitis            | MESAI_BITIS             | Time                      | Hayır   | ?                               | -           | -                             | -                                | -  |
| İstasyon Tipi                 | İstasyon Tipi Kodu     | ISTASYON_TIPI_KODU      | Characters (36)           | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|                               | İsim                   | ISIM                    | Variable characters (100) | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
| İz Bilgisi                    | İz Bilgisi ID          | IZ_BILGISI_ID           | Characters (36)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Tarih                  | TARIH                   | Date & Time               | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | UTC Zaman              | UTC_ZAMAN               | Date & Time               | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Koordinat              | KOORDINAT               | Characters (15)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Rakim                  | RAKIM                   | Integer                   | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Hiz                    | HIZ                     | Integer                   | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Kara Arac Fonksiyon           | Arac Fonksiyon Kodu    | ARAC_FONKSIYON_KODU     | Variable characters (30)  | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|                               | İsim                   | ISIM                    | Variable characters (100) | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |



EK-11: Sistem Veri Değişimi

| ENTITY                  | Adı                          | Kodu                         | Veri Tipi                 | Zorunlu | ŞİRKET BİLGİ SİSTEM VERİ TABANI | ŞİRKET LOJİSTİK BİLGİ SİSTEMİ | ŞİRKET OPERASYONEL BİLGİ SİSTEMİ | ŞİRKETİN BAĞLI OLDUĞU ULUSLARARASI SİSTEMLER |
|-------------------------|------------------------------|------------------------------|---------------------------|---------|---------------------------------|-------------------------------|----------------------------------|--|
| Kara Arac Kullanım Tipi | Kullanım Tipi Kodu           | KULLANIM_TIPI_KODU           | Variable characters (30)  | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
| Karayolu Aracı          | İsim                         | ISIM                         | Variable characters (100) | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
|                         | Ağırlık                      | AGIRLIK                      | Integer                   | Hayır   | ŞÖK                             | ŞÖK                           | ŞÖK                              | ŞÖK  |
|                         | Yük Sınıfı                   | YUK_SINIFI                   | Characters (5)            | Hayır   | ŞÖK                             | ŞÖK                           | ŞÖK                              | ŞÖK  |
|                         | Platform Yüksekliği Maksimum | PLATFORM_YUKSEKLIGI_MAKSIMUM | Integer                   | Hayır   | ŞÖK                             | ŞÖK                           | ŞÖK                              | ŞÖK  |
|                         | Platform Yüksekliği Minimum  | PLATFORM_YUKSEKLIGI_MINIMUM  | Integer                   | Hayır   | ŞÖK                             | ŞÖK                           | ŞÖK                              | ŞÖK  |
| Kargo Bölüm Tipi        | Kargo Bölüm Tipi Kodu        | KARGO_BOLUM_TIPI_KODU        | Variable characters (30)  | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
|                         | İsim                         | ISIM                         | Variable characters (100) | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
| Kargo Bölümü            | Kargo Bölümü ID              | KARGO_BOLUMU_ID              | Characters (36)           | Evet    | -                               | -                             | +                                | -  |
|                         | Sıra No                      | SIRA_NO                      | Integer                   | Evet    | -                               | -                             | +                                | -  |
|                         | İsim                         | ISIM                         | Variable characters (100) | Hayır   | -                               | -                             | +                                | -  |
|                         | Yükleme Önceligi             | YUKLEME_ONCELIGI             | Integer                   | Evet    | -                               | -                             | -                                | -  |
|                         | Uzunluk                      | UZUNLUK                      | Integer                   | Hayır   | -                               | -                             | -                                | -  |
|                         | Genislik                     | GENISLIK                     | Integer                   | Hayır   | -                               | -                             | +                                | -  |
|                         | Yükseklik                    | YUKSEKLIK                    | Integer                   | Hayır   | -                               | -                             | +                                | -  |
|                         | Maks Kargo Genislik          | MAKS_KARGO_GENISLIK          | Integer                   | Hayır   | -                               | -                             | +                                | -  |
|                         | Maks Kargo Uzunluk           | MAKS_KARGO_UZUNLUK           | Integer                   | Hayır   | -                               | -                             | +                                | -  |
|                         | Maks Kargo Yükseklik         | MAKS_KARGO_YUKSEKLIK         | Integer                   | Hayır   | -                               | -                             | +                                | -  |
|                         | Maks Kargo Ağırlık           | MAKS_KARGO_AGIRLIK           | Integer                   | Hayır   | -                               | -                             | +                                | -  |
|                         | Taban Maksimum Yük Direnci   | TABAN_MAKSIMUM_YUK_DIRENCI   | Decimal (7.2)             | Hayır   | -                               | -                             | ?                                | -  |
|                         | Maks İstif Ağırlığı          | MAKS_ISTIF_AGIRLIGI          | Integer                   | Hayır   | -                               | -                             | +                                | -  |
|                         | Yükleme Platformu Yüksekliği | YUKLEME_PLATFORMU_YUKSEKLIGI | Integer                   | Hayır   | -                               | -                             | ?                                | -  |
|                         | Yolcu Kapasitesi             | YOLCU_KAPASITESI             | Integer                   | Hayır   | -                               | -                             | ?                                | -  |
|                         | Ağırlık Kapasitesi           | AGIRLIK_KAPASITESI           | Integer                   | Hayır   | -                               | -                             | +                                | -  |
|                         | Lane Metre                   | LANE_METRE                   | Decimal (7.2)             | Hayır   | -                               | -                             | -                                | -  |
|                         | Alan Kapasitesi              | ALAN_KAPASITESI              | Decimal (7.2)             | Hayır   | -                               | -                             | -                                | -  |
|                         | Hacim Kapasitesi             | HACIM_KAPASITESI             | Decimal (8.2)             | Hayır   | -                               | -                             | ?                                | -  |
|                         | TEU Kapasitesi               | TEU_KAPASITESI               | Integer                   | Hayır   | -                               | -                             | +                                | -  |
| Kargo Depo Tipi         | Kargo Depo Tipi Kodu         | KARGO_DEPO_TIPI_KODU         | Variable characters (30)  | Hayır   | VDL                             | VDL                           | VDL                              | VDL  |
|                         | İsim                         | ISIM                         | Variable characters (100) | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
| Kullanıcı               | Kullanıcı ID                 | KULLANICI_ID                 | Characters (36)           | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                         | Kullanıcı Adı                | KULLANICI_ADI                | Variable characters (20)  | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                         | Adı Soyadı                   | ADI_SOYADI                   | Variable characters (20)  | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                         | Sifre                        | SIFRE                        | Variable characters (50)  | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Kullanıcı Yetki Grubu   | Yetki ID                     | YETKI_ID                     | Characters (36)           | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Kurye-Ring              | Kurye-Ring Kodu              | KURYE_RING_KODU              | Variable characters (30)  | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                         | İsim                         | ISIM                         | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                         | Acıklama                     | ACIKLAMA                     | Variable characters (500) | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                         | Aktif                        | AKTIF                        | Boolean                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Kurye-Ring Bacak        | Kurye-Ring Bacak ID          | KURYE_RING_BACAK_ID          | Characters (36)           | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                         | Sıra No                      | SIRA_NO                      | Integer                   | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                         | Aktif                        | AKTIF                        | Boolean                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Kurye-Ring Sefer Kodu   | Sefer Kodu                   | SEFER_KODU                   | Variable characters (30)  | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Kurye-Ring Tipi         | Kurye-Ring Tip Kodu          | KURYE_RING_TIP_KODU          | Variable characters (30)  | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                         | İsim                         | ISIM                         | Variable characters (100) | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
|                         | Liman ID                     | LIMAN_ID                     | Characters (36)           | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
|                         | İsim                         | ISIM                         | Variable characters (100) | Evet    | -                               | -                             | -                                | -  |
|                         | Acıklama                     | ACIKLAMA                     | Variable characters (500) | Hayır   | +                               | +                             | -                                | -  |

EK-11: Sistem Veri Değişimi

| ENTITY | Adı                       | Kodu                      | Veri Tipi                 | Zorunlu | ŞİRKET BİLGİ SİSTEM VERİ TABANI | ŞİRKET CALS | ŞİRKET LOJİSTİK BİLGİ SİSTEMİ | ŞİRKET OPERASYONEL BİLGİ SİSTEMİ | ŞİRKETİN BAĞLI OLDUĞU ULUSLARARASI SİSTEMLER |
|--------|---------------------------|---------------------------|---------------------------|---------|---------------------------------|-------------|-------------------------------|----------------------------------|--|
|        | Rihtim Sayısı             | RIHTIM_SAYISI             | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Maks Gemi Kiris Yukseklik | MAKS_GEMI_KIRIS_YUKSEKLIK | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Maks Gemi Uzunluk         | MAKS_GEMI_UZUNLUK         | Integer                   | Hayır   | ?                               | -           | -                             | -                                | -  |
|        | Maks Gemi Draft           | MAKS_GEMI_DRAFT           | Integer                   | Hayır   | ?                               | -           | -                             | -                                | -  |
|        | Yakit İkmali              | YAKIT_IKMALI              | Boolean                   | Hayır   | +                               | -           | -                             | -                                | -  |
|        | Agir Vinç                 | AGIR_VINC                 | Boolean                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Tanker Kolayliklari       | TANKER_KOLAYLIKлари       | Boolean                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Romorkor Mevcut           | ROMORKOR_MEVCUT           | Boolean                   | Hayır   | +                               | -           | -                             | -                                | -  |
|        | Tehlikeli Madde Izni      | TEHLIKELI_MADDE_IZNI      | Boolean                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Mavnaya Yukleme           | MAVNAYA_YUKLEME           | Boolean                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Lash                      | LASH                      | Boolean                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Ellecleme Kapasitesi ID   | ELLECLEME_KAPASITESI_ID   | Characters (36)           | Evet    | -                               | -           | -                             | -                                | -  |
|        | Açıklama                  | ACIKLAMA                  | Variable characters (500) | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Hiz                       | HIZ                       | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Maks Genislik             | MAKS_GENISLIK             | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Maks Uzunluk              | MAKS_UZUNLUK              | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Maks Yukseklik            | MAKS_YUKSEKLIK            | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Maks Agirlik              | MAKS_AGIRLIK              | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | SomaJa Kalma Süresi       | SOMAJA_KALMA_SURESI       | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Mesai Baslangic           | MESAI_BASLANGIC           | Time                      | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Mesai Bitis               | MESAI_BITIS               | Time                      | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Liman Rihtim ID           | LIMAN_RIHTIM_ID           | Characters (36)           | Evet    | -                               | -           | -                             | -                                | -  |
|        | Rihtim Ismi               | RIHTIM_ISMI               | Variable characters (100) | Evet    | -                               | -           | -                             | -                                | -  |
|        | Terminal Ismi             | TERMINAL_ISMI             | Variable characters (100) | Evet    | -                               | -           | -                             | -                                | -  |
|        | Maks Gemi Kiris Yukseklik | MAKS_GEMI_KIRIS_YUKSEKLIK | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Maks Gemi Uzunluk         | MAKS_GEMI_UZUNLUK         | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Maks Gemi Draft           | MAKS_GEMI_DRAFT           | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Maks Gemi Deadweight      | MAKS_GEMI_DEADWEIGHT      | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Tehlikeli Madde Izni      | TEHLIKELI_MADDE_IZNI      | Boolean                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Apron Yuksekligi          | APRON_YUKSEKligi          | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Apron Genisligi           | APRON_GENISLIGI           | Integer                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Demiryolu Baglantisi      | DEMIRYOLU_BAGLANTISI      | Boolean                   | Hayır   | -                               | -           | -                             | -                                | -  |
|        | Liman Rihtim Tipi ID      | LIMAN_RIHTIM_TIPI_ID      | Characters (36)           | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|        | Isim                      | ISIM                      | Variable characters (100) | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|        | Liman Tipi Kodu           | LIMAN_TIPI_KODU           | Characters (36)           | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|        | Isim                      | ISIM                      | Variable characters (100) | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|        | Lokasyon ID               | LOKASYON_ID               | Characters (36)           | Evet    | TBS CBS                         | TBS CBS     | TBS CBS                       | TBS CBS                          | TBS CBS                                      |
|        | Lokasyon Kodu             | LOKASYON_KODU             | Variable characters (30)  | Evet    | TBS CBS                         | TBS CBS     | TBS CBS                       | TBS CBS                          | TBS CBS                                      |
|        | Isim                      | ISIM                      | Variable characters (100) | Evet    | TBS CBS                         | TBS CBS     | TBS CBS                       | TBS CBS                          | TBS CBS                                      |
|        | Enlem                     | ENLEM                     | Decimal                   | Evet    | TBS CBS                         | TBS CBS     | TBS CBS                       | TBS CBS                          | TBS CBS                                      |
|        | Boylam                    | BOYLAM                    | Decimal                   | Hayır   | TBS CBS                         | TBS CBS     | TBS CBS                       | TBS CBS                          | TBS CBS                                      |
|        | Açıklama                  | ACIKLAMA                  | Variable characters (500) | Hayır   | TBS CBS                         | TBS CBS     | TBS CBS                       | TBS CBS                          | TBS CBS                                      |
|        | Ellecleme Tipi Kod        | ELLECLEME_TIPI_KOD        | Variable characters (30)  | Hayır   | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|        | Isim                      | ISIM                      | Variable characters (100) | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|        | Lokasyon Türü Kod         | LOKASYON_TURU_KOD         | Variable characters (30)  | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|        | Isim                      | ISIM                      | Variable characters (100) | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|        | Lokasyon Yakit Tipi ID    | LOKASYON_YAKIT_TIPI_ID    | Characters (36)           | Evet    | VDL                             | VDL         | VDL                           | VDL                              | VDL  |
|        | Malzeme ID                | MALZEME_ID                | Characters (36)           | Evet    | ?                               | -           | +                             | +                                | -  |

EK-11: Sistem Veri Değişimi

| ENTITY        | Adı                      | Kodu                     | Veri Tipi                 | Zorunlu | ŞİRKET BİLGİ SİSTEM VERİ TABANI | ŞİRKET LOJİSTİK BİLGİ SİSTEMİ | ŞİRKET OPERASYONEL BİLGİ SİSTEMİ | ŞİRKETİN BAĞLI OLDUĞU ULUSLARARASI SİSTEMLER |
|---------------|--------------------------|--------------------------|---------------------------|---------|---------------------------------|-------------------------------|----------------------------------|--|
|               | Stok No                  | MILLI_STOK_NO            | Variable characters (30)  | Evet    | ?                               | +                             | +                                | -  |
|               | İsim                     | ISIM                     | Variable characters (100) | Evet    | ?                               | +                             | +                                | -  |
|               | Tehlikeli                | TEHLIKELI                | Boolean                   | Evet    | ?                               | -                             | +                                | -  |
|               | Yol Köprü Sınıfı         | YOL_KOPRU_SINIFI         | Characters (5)            | Hayır   | ?                               | -                             | -                                | -  |
|               | Karayolu Tasinabilirlik  | KARAYOLU_TASINABILIRLIK  | Boolean                   | Hayır   | ?                               | -                             | +                                | -  |
|               | Havayolu Tasinabilirlik  | HAVAYOLU_TASINABILIRLIK  | Boolean                   | Hayır   | ?                               | -                             | +                                | -  |
|               | Denizyolu Tasinabilirlik | DENIZYOLU_TASINABILIRLIK | Boolean                   | Hayır   | ?                               | -                             | +                                | -  |
|               | Demiryolu Tasinabilirlik | DEMIRYOLU_TASINABILIRLIK | Boolean                   | Hayır   | ?                               | -                             | +                                | -  |
|               | Operasyon En             | OPERASYON_EN             | Integer                   | Hayır   | ?                               | -                             | -                                | -  |
|               | Operasyon Boy            | OPERASYON_BOY            | Integer                   | Hayır   | ?                               | -                             | -                                | -  |
|               | Operasyon Yükseklik      | OPERASYON_YUKSEKLIK      | Integer                   | Hayır   | ?                               | -                             | -                                | -  |
|               | Tasimaya Esas En         | TASIMAYA_ESAS_EN         | Integer                   | Hayır   | ?                               | -                             | +                                | -  |
|               | Tasimaya Esas Boy        | TASIMAYA_ESAS_BOY        | Integer                   | Hayır   | ?                               | -                             | +                                | -  |
|               | Tasimaya Esas Yükseklik  | TASIMAYA_ESAS_YUKSEKLIK  | Integer                   | Hayır   | ?                               | -                             | +                                | -  |
|               | Brüt Ağırlık             | BRUT_AGIRLIK             | Integer                   | Hayır   | ?                               | +                             | +                                | -  |
|               | Dara                     | DARA                     | Integer                   | Hayır   | ?                               | -                             | +                                | -  |
|               | Açıklama                 | ACIKLAMA                 | Variable characters (500) | Hayır   | ?                               | -                             | +                                | -  |
|               | Resim                    | RESIM                    | Bitmap                    | Hayır   | ?                               | -                             | +                                | -  |
| Malzeme Renk  | Malzeme Renk Kodu        | MALZEME_RENK_KODU        | Variable characters (30)  | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
|               | İsim                     | ISIM                     | Variable characters (100) | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
| Ondeger       | İsim                     | ISIM                     | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Deger                    | DEGER                    | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Ozellik Deger | Ozellik Deger ID         | OZELLIK_DEGER_ID         | Characters (36)           | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Deger                    | DEGER                    | Long variable characters  | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Ozellik Tanim | Ozellik Tanim ID         | OZELLIK_TANIM_ID         | Characters (36)           | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | İsim                     | ISIM                     | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Baslık                   | BASLIK                   | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Sıra No                  | SIRA_NO                  | Integer                   | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Açıklama                 | ACIKLAMA                 | Variable characters (500) | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Veri Türü                | VERI_TURU                | Characters (1)            | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Veri Formatı             | VERI_FORMATI             | Variable characters (50)  | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Zorunlu                  | ZORUNLU                  | Boolean                   | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Grup İsim                | GRUP_ISIM                | Variable characters (20)  | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | İsim                     | ISIM                     | Variable characters (100) | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
|               | Kısaltma                 | KISALTMA                 | Variable characters (5)   | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
| Plan          | Plan ID                  | PLAN_ID                  | Characters (36)           | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Kod                      | KOD                      | Variable characters (30)  | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | İsim                     | ISIM                     | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Referans Tarihi          | REFERANS_TARIHI          | Date                      | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Referans Günü            | REFERANS_GUNU            | Integer                   | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Açıklama                 | ACIKLAMA                 | Variable characters (500) | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Plan Tipi     | Plan Tipi Kod            | PLAN_TIPI_KOD            | Characters (5)            | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
|               | İsim                     | ISIM                     | Variable characters (100) | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
| Profil        | Profil ID                | PROFIL_ID                | Characters (36)           | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | İsim                     | ISIM                     | Variable characters (100) | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Sanal Profil             | SANAL_PROFIL             | Boolean                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|               | Açıklama                 | ACIKLAMA                 | Variable characters (500) | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| REK           | Kod                      | KOD                      | Characters (6)            | Evet    | -                               | -                             | -                                | +  |

EK-11: Sistem Veri Değişimi

| ENTITY          | Adı                    | Kodu                   | Veri Tipi                 | Zorunlu | ŞİRKET BİLGİ SİSTEM VERİ TABANI | ŞİRKET CALS | ŞİRKET LOJİSTİK BİLGİ SİSTEMİ | ŞİRKET OPERASYONEL BİLGİ SİSTEMİ | ŞİRKETİN BAĞLI OLDUĞU ULUSLARARASI SİSTEMLER |
|-----------------|------------------------|------------------------|---------------------------|---------|---------------------------------|-------------|-------------------------------|----------------------------------|--|
|                 | Ulusal Kod             | ULUSAL_KOD             | Variable characters (30)  | Hayır   | -                               | -           | -                             | -                                | +  |
|                 | Tanımı                 | TANIMI                 | Characters (50)           | Evet    | -                               | -           | -                             | -                                | +  |
|                 | Tanım Türkçe           | TANIM_TURKCE           | Characters (50)           | Hayır   | -                               | -           | -                             | -                                | +  |
|                 | Kısa Tanım             | KISA_TANIMI            | Characters (50)           | Hayır   | -                               | -           | -                             | -                                | +  |
|                 | Kısa Tanım Türkçe      | KISA_TANIM_TURKCE      | Characters (50)           | Hayır   | -                               | -           | -                             | -                                | +  |
| REK Gösterge    | Kod                    | KOD                    | Characters (1)            | Evet    | -                               | -           | -                             | -                                | +  |
|                 | Tanımı                 | TANIMI                 | Characters (50)           | Evet    | -                               | -           | -                             | -                                | +  |
|                 | Tanım Türkçe           | TANIM_TURKCE           | Characters (50)           | Hayır   | -                               | -           | -                             | -                                | +  |
| Rezervasyon     | Rezervasyon ID         | REZERVASYON_ID         | Characters (36)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Durum                  | DURUM                  | Byte                      | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Yükleme Önceliği       | YUKLEME_ONCELIGI       | Integer                   | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Sıra No                | SIRA_NO                | Integer                   | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Yedek                  | YEDEK                  | Boolean                   | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Yüklemlendi            | YUKLENDI               | Boolean                   | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Bosaltıldı             | BOSALTILDI             | Boolean                   | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Sefer           | Sefer ID               | SEFER_ID               | Characters (36)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | İsim                   | ISIM                   | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Acıklama               | ACIKLAMA               | Variable characters (500) | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Tarife Kalkis          | TARIFE_KALKIS          | Date & Time               | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Tarife Varis           | TARIFE_VARIS           | Date & Time               | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Kalkis                 | KALKIS                 | Date & Time               | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Varis                  | VARIS                  | Date & Time               | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Sozluk Deger    | Deger                  | DEGER                  | Variable characters (30)  | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Sıra No                | SIRA_NO                | Integer                   | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | İsim                   | ISIM                   | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | İsim2                  | ISIM2                  | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Sozluk Tipi     | Sozluk Tip Kodu        | SOZLUK_TIP_KODU        | Variable characters (30)  | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | İsim                   | ISIM                   | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Baslik                 | BASLIK                 | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Talep           | Talep ID               | TALEP_ID               | Characters (36)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Talep Tarihi           | TALEP_TARIHI           | Date                      | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Kiymetten Düsme Tarihi | KIYMETTEN_DUSME_TARIHI | Date                      | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Talep Manifesto | Talep Manifesto ID     | TALEP_MANIFESTO_ID     | Characters (36)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Agirlik                | AGIRLIK                | Integer                   | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Malzeme En             | MALZEME_EN             | Integer                   | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Malzeme Boy            | MALZEME_BOY            | Integer                   | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Malzeme Yükseklik      | MALZEME_YUKSEKLIK      | Integer                   | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Malzeme Hacim          | MALZEME_HACIM          | Decimal (8,2)             | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Yolcu Miktarı          | YOLCU_MI               | Boolean                   | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Barkod                 | BARKOD                 | Variable characters (50)  | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Yolcu Kod No           | YOLCU_KOD_NO           | Variable characters (30)  | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Yolcu Adı              | YOLCU_ADI              | Variable characters (20)  | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | Yolcu Soyad            | YOLCU_SOYAD            | Variable characters (20)  | Hayır   | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Talep Nedeni    | Talep Nedeni ID        | TALEP_NEDENI_ID        | Characters (36)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | İsim                   | ISIM                   | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Talep Tipi      | Talep Tipi ID          | TALEP_TIPI_ID          | Characters (36)           | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                 | İsim                   | ISIM                   | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL    | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Terminal        | Terminal ID            | TERMINAL_ID            | Characters (36)           | Evet    | -                               | -           | -                             | -                                | -  |

EK-11: Sistem Veri Değişimi

| ENTITY                        | Adı                     | Kodu                    | Veri Tipi                 | Zorunlu | ŞİRKET BİLGİ SİSTEM VERİ TABANI | ŞİRKET LOJİSTİK BİLGİ SİSTEMİ | ŞİRKET OPERASYONEL BİLGİ SİSTEMİ | ŞİRKETİN BAĞLI OLDUĞU ULUSLARARASI SİSTEMLER |
|-------------------------------|-------------------------|-------------------------|---------------------------|---------|---------------------------------|-------------------------------|----------------------------------|--|
|                               | İsim                    | ISIM                    | Variable characters (100) | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
|                               | Acıklama                | ACIKLAMA                | Variable characters (500) | Hayır   | –                               | –                             | –                                | –  |
| Terminal Ellekleme Kapasitesi | Ellekleme Kapasitesi ID | ELLECLEME_KAPASITESI_ID | Characters (36)           | Evet    | –                               | –                             | –                                | –  |
|                               | Açıklama                | ACIKLAMA                | Variable characters (500) | Hayır   | –                               | –                             | –                                | –  |
|                               | Hiz                     | HIZ                     | Integer                   | Hayır   | –                               | –                             | –                                | –  |
|                               | Maks Genislik           | MAKS_GENISLIK           | Integer                   | Hayır   | –                               | –                             | –                                | –  |
|                               | Maks Uzunluk            | MAKS_UZUNLUK            | Integer                   | Hayır   | –                               | –                             | –                                | –  |
|                               | Maks Yükseklik          | MAKS_YUKSEKLIK          | Integer                   | Hayır   | –                               | –                             | –                                | –  |
|                               | Maks Ağırlık            | MAKS_AGIRLIK            | Integer                   | Hayır   | –                               | –                             | –                                | –  |
|                               | Somaja Kalma Süresi     | SOMAJA_KALMA_SURESI     | Integer                   | Hayır   | –                               | –                             | –                                | –  |
|                               | Mesai Başlangıç         | MESAI_BASLANGIC         | Time                      | Hayır   | –                               | –                             | –                                | –  |
|                               | Mesai Bitis             | MESAI_BITIS             | Time                      | Hayır   | –                               | –                             | –                                | –  |
| Terminal Tipi                 | Terminal Tipi Kodu      | TERMINAL_TIPI_KODU      | Characters (36)           | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
|                               | İsim                    | ISIM                    | Variable characters (100) | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
| Ulasim Hatti                  | Ulasim Hatti ID         | ULASIM_HATTI_ID         | Characters (36)           | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Mesafe                  | MESAFE                  | Integer                   | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Ulasim Hatti Kisit            | Ulasim Hatti Kisit ID   | ULASIM_HATTI_KISIT_ID   | Characters (36)           | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | İsim                    | ISIM                    | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | İleri                   | ILERI                   | Boolean                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Geri                    | GERI                    | Boolean                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Minimum Hiz             | MINIMUM_HIZ             | Integer                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Maksimum Hiz            | MAKSIMUM_HIZ            | Integer                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Maksimum Uzunluk        | MAKSIMUM_UZUNLUK        | Integer                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Maksimum Genislik       | MAKSIMUM_GENISLIK       | Integer                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Maksimum Yükseklik      | MAKSIMUM_YUKSEKLIK      | Integer                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Maksimum Ağırlık        | MAKSIMUM_AGIRLIK        | Integer                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Kullanım Disi Tut       | KULLANIM_DISI_TUT       | Boolean                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Maksimum Egim           | MAKSIMUM_EGIM           | Integer                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Maksimum Arac Sayisi    | MAKSIMUM_ARAC_SAYISI    | Integer                   | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Ulastirma Modu                | Ulastirma Modu Kodu     | ULASTIRMA_MODU_KODU     | Characters (5)            | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
|                               | İsim                    | ISIM                    | Variable characters (100) | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
| Ulke                          | Ulke Kod                | ULKE_KOD                | Characters (5)            | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
|                               | İsim                    | ISIM                    | Variable characters (100) | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
| Varlik Tipi                   | Varlik Tipi ID          | VARLIK_TIPI_ID          | Characters (36)           | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | İsim                    | ISIM                    | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | Acıklama                | ACIKLAMA                | Variable characters (500) | Hayır   | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Yakit Tipi                    | Yakit Tipi Kod          | YAKIT_TIPI_KOD          | Characters (5)            | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
|                               | İsim                    | ISIM                    | Variable characters (100) | Evet    | VDL                             | VDL                           | VDL                              | VDL  |
| Yetki Grubu                   | Yetki Grup ID           | YETKI_GRUP_ID           | Characters (36)           | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
|                               | İsim                    | ISIM                    | Variable characters (100) | Evet    | INTERNAL                        | INTERNAL                      | INTERNAL                         | INTERNAL                                     |
| Yukleme Oncelik               | Yukleme Öncelik ID      | YUKLEME_ONCELIK_ID      | Characters (36)           | Evet    | –                               | –                             | –                                | –  |
| Yukleme Oncelik Tanim         | Yukleme Öncelik Sira No | YUKLEME_ONCELIK_SIRA_NO | Integer                   | Evet    | –                               | –                             | –                                | –  |
|                               | Kod                     | KOD                     | Characters (5)            | Evet    | –                               | –                             | –                                | –  |
|                               | Acıklama                | ACIKLAMA                | Variable characters (500) | Hayır   | –                               | –                             | –                                | –  |

**EK-12: Yazılım Geliştirme Planı**

# **Yazılım Geliştirme Planı**

**Versiyon .....**

## **1. GİRİŞ**

### **1.1. Amaç ve Kapsam**

Sistem, ulařtırma ihtiyaçlarının en etkin, sűratli ve ekonomik olarak karřılanmasına yűnelik ulařtırma planlamalarının; mevcut ulařtırma kaynaklarının, araç boş kapasitelerinin ve faaliyetlerinin tűm/yetkili ulařtırma planlamacıları tarafından karřılıklı olarak zamanında gűrűlmesi, paralel ve entegre olarak yapılabilmesi ve karar verici makamlara karar desteęi saęlanması ana amacını tařımaktadır.

Proje Yazılım Geliřtirme Planı, bu proje kapsamındaki yazılımların geliřtirilmesinde uygulanacak genel proje yűnetim ve yazılım geliřtirme aktiviteleri ile birlikte, plan, sűreçler ve proje kaynakları ve dokűmantasyona yűnelik, IEEE ve PMI standartlarında belirlenen çerçevede bilgiler iermektedir. Yazılım geliřtirme planının ana amacı, proje kapsamında uygulanacak yazılım geliřtirme sűreçlerini űzetlemektir. Bu kapsamda, proje yűnetim sűreçleri ierisinde yer alan/alacak tűm proje ekibine ve proje paydařlarına, bu proje kapsamında uygulanacak proje yűnetim yaklařımı ile uygulama yazılımı geliřtirme sűreçlerinde uygulanacak standart aktiviteler sunulmaktadır.

Yazılım geliřtirme ekibi tarafından uygulanacak yazılım geliřtirme faaliyetleri, MSF/SDD, PMI ve IEEE 12207 standartlarına uygun olarak gerekleřtirilecektir. Dięer proje yűnetimi/uygulama yazılımı geliřtirme yaklařımlarından farklı olarak, uygulanan yazılım geliřtirme metodolojisi Rational, Prince 2, UML gibi farklı metodolojilerden de faydalanabilmeyi műmkűn kılmakta ve onlarla bűtűnleřerek CMM, SPICE, ISO 9001:2000 vb. kalite yűnetim sistemlerini uygulamayı etkinleřtirmektedir. Bu dokűman ile, genel olarak, projenin izlenebilmesi, kontrolű ve planlamasının yapılabilmesine yűnelik yűnetimsel bir ara saęlanması hedeflenmiř ve dokűman tűműyle, uygulanan yazılım geliřtirme metodolojilerine de uygun olarak, uluslararası kabul gűrműř PMI ve IEEE 12207 sűre ve standartlarına uygun olarak hazırlanmıřtır.

### **1.2. Proje űzeti**

Sistem; kullanıcının gűnűműzdeki ve gelecekteki ihtiyaçlarına cevap verecek űekilde, kurye/ring seferleri planlaması, műnferit veya toplu tařımaların sűratle ve maliyet-etkin olarak planlanması, icrası ve icrasının gerek ve gereęe yakın zamanlı olarak takibi, mevcut ulařtırma kaynaklarının deęerlendirilmesi ve bu

planlama ve değerlendirme faaliyetleri konusunda her seviyedeki karar verici makama karar desteği sağlanması ve ulaştırma planlamacılarının taşımaya esas bilgilere eş zamanlı olarak ulaşması ile paralel ve entegre planlama yapılması imkanı sağlayan çok modlu taşıma planlama ve koordinasyon sistemi olarak tanımlanabilir. Sistemin, genel olarak, planlamalar ve icra sürecinin izlenmesi ile ilgili çalışan unsur, kurum ve kuruluşları kapsamı hedeflenmiştir.

Sistem, taşımaya esas malzeme bilgileri ve ulaştırma alt yapısının planlanmasına yönelik bütünleşik ortak bir veri tabanı (VTYM) ve bu veri tabanını kullanarak, taşınacak birim taşımalarının ve görevlendirilen kuryelerin en maliyet etkin bir şekilde planlanması, programlanması ve koordinasyonuna yönelik ulaştırma ihtiyaçlarının ortaya konması (TPM ve KURPM) ve yapılan planlamaları müteakip işlerliğin ve olası problem alanlarının belirlenebilmesi için gerçek ve gerçeğe yakın zamanlı etkin araç/malzeme iz takibinin yapılabilmesi (ETARİTM) amacıyla geliştirilecek dört ayrı modülden oluşmaktadır;

#### **1.2.1. Veri Tabanı Yönetim Modülü (VTYM)**

VTYM, sistemin diğer modülleri (KURPM, TPM, ETARİTMS) tarafından kullanılacak, kullanıcının diğer mevcut ve geliştirilmekte olan lojistik ve komuta kontrol bilgi sistemleri ile veri değişimi yapabilecek, veri tabanını yönetecek, işleyecek, değiştirebilecek, güncelleştirebilecek fonksiyonlar içeren bir sistem olarak tasarlanmıştır.

VTYM, sistem kapsamındaki diğer modüllerin ihtiyaç duyabileceği taşınacak birim, yolcu, malzeme bilgileri, ulaştırma araç bilgileri, altyapı bilgileri, iz bilgileri, planlı kurye/ring seferleri bilgileri, plansız ulaştırma aracı tahsis bilgileri, planlı/plansız seferlere planlanmış yolcu/yük bilgileri, KURPM'ta kullanılan ulaştırma araç/ altyapı bilgileri, taşıma plan bilgilerini içermesi hedeflenen genel bir veritabanıdır.

#### **1.2.2. Taşıma Planlama Modülü (TPM)**

TPM, genel olarak, VTYM veri tabanını kullanarak, taşımaların maliyet etkin bir şekilde planlanabilmesi, programlanabilmesi ve koordinasyonuna yönelik ulaştırma ihtiyaçlarının ortaya konması için geliştirilen bir karar destek sistemi olarak tanımlanabilir. Bu çerçevede, TPM aşağıda listelenen ana hedefleri kapsamaktadır;

- Taşıma yeteneğinin, maliyet etkin, hızlı taşıma planlarıyla artırılması,



- Planlama kapsamındaki taşınacak birimlerin istenilen zamanda istenilen yerde olmalarının sağlanması,
- Taşıma esnasında, hangi taşınacak birimin, ne kadar sürede, ne kadar uzağa, hangi ulaştırma araçları ile ve hangi ulaştırma altyapısını kullanarak uygun bir planlama süresi içinde taşınmasına yönelik planlamaların sağlanması.

### **1.2.3. Kurye/Ring Yolcu/Yük Planlama Modülü (KURPM)**

KURPM, kullanıcının yurt içi ve yurt dışında taşınacak birimlerin taşınması için, maliyet etkin kurye/ring seferleri yapısı oluşturabilmek amacıyla, yeni kurye/ring seferleri ihdas edilmesinde ve mevcut seferlerinin birleştirilmesi veya uygulamadan kaldırılmasında karar desteği sağlayacak ve ihdas edilmiş kurye/ring seferlerine yönelik taleplerin maliyet-etkin olarak karşılanmasının planlanmasını sağlayacak alt sistem olarak tanımlanabilir.

### **1.2.4. Etkin Araç İz Takip Modülü (ETARİTM)**

ETARİTM, sistem kapsamında gerçekleştirilecek planlamaların işlerliğini ve olası problem alanlarının belirlenebilmesi için gerçek ve gerçeğe yakın zamanlı araç/malzeme iz takibinin yapılabilmesi sağlayacak alt sistem olarak tanımlanabilir.

Uluslararası ortamda kabul görmüş bütünsel aktif ve pasif iz takip yeteneklerini içerecek şekilde geliştirilecek olan bu modül MAP-INFO yazılımı ve güncel/istendiği zaman güncellenebilir verilerini kullanacaktır. Aktif iz takip kitleri ile iletişim/haberleşme altyapısı kullanılarak, ETARİTM Sunucusuna otomatik olarak gönderilecek enlem, boylam, yükseklik, UTC zamanı, aracın hızı ve yönü gibi konum verileri sisteme otomatik olarak aktarılacaktır.

Projenin ihtiyaçları kullanıcı ve geliştirici arasında imzalanan ..... tarih ve ..... Nu.lı Sözleşme eklerinde yer alan idari ve teknik şartnamelerde belirtilmiştir. İdari ve teknik şartnamelerde sistemin, mevcut taşıma yeteneklerinin ve kuryelerin en maliyet etkin şekilde planlanması ve programlanması, anılan optimizasyon sürecinin gerçekleştirilmesine ve karşılaşılan taşıma taleplerinin yerine getirilmesine yönelik ön değerlendirmelerin yapılabilmesi için gerekli veri setlerini içeren sistem veri tabanı alt yapısının ve veri tabanının geliştirilmesi ve müteakiben planın işlerliğinin ve olası problem alanlarının belirlenebilmesi için etkin araç iz takip planlama sistemlerinin geliştirilmesine yönelik ihtiyaç duyulan sistem, endüstri ve lojistik alanlarındaki mühendislik hizmetleri, danışmanlık, yazılım, donanım, eğitim

ve işletim desteği ile ilgili mal ve hizmetlerin tedarikine ilişkin hususlar detaylı bir biçimde açıklanmıştır.

#### **1.2.5. Referanslar**

Kullanıcı ve geliştirici arasında imzalanan ..... tarih ve ..... No'lu Sözleşme Sistem Teknik ve İdari Şartnameleri.

IEEE Yazılım Geliştirme Standartları; "Institute of Electrical and Electronics Engineers. IEEE Standard for Software Development Plan, ANSI/IEEE Standard"

IEEE 12207 Yazılım Yaşam Döngüsü

PMI Rehber Dokümanı; "A Guide to the Project Management Body of Knowledge, Project Management Institute"

#### **2. YAZILIM GELİŞTİRME FAALİYETLERİ**

Bu bölüm Proje Uygulama Planı dokümanında açıklanmıştır.

#### **3. KALİTE GÜVENCE ve RİSK YÖNETİMİ**

Bu bölüm Proje Uygulama Planı dokümanında açıklanmıştır.

#### **4. SONUÇ**

Bu bölüm Proje Uygulama Planı dokümanında açıklanmıştır.

**EK-13: Proje Uygulama Planı**

# **Proje Uygulama Planı**

**Versiyon .....**

## 1. GİRİŞ

### 1.1. Amaç ve Kapsam

Sistem, ulaştırma ihtiyaçlarının en etkin, süratli ve ekonomik olarak karşılanmasına yönelik ulaştırma planlamalarının; mevcut ulaştırma kaynaklarının, araç boş kapasitelerinin ve faaliyetlerinin tüm/yetkili ulaştırma planlamacıları tarafından karşılıklı olarak zamanında görülmesi, paralel ve bütünlük olarak yapılabilmesi ve karar verici makamlara karar desteği sağlanması ana amacını taşımaktadır.

Proje Uygulama Planı dokümanı ile; bahse konu temel amacın sağlanabilmesi maksadıyla, uygulanması planlanan metodolojiler ve uygulama esaslarının ortaya konması hedeflenmiştir. Proje Uygulama Planı;

- Proje kapsamında uygulanacak genel proje yönetim ve uygulama faaliyetlerine yönelik IEEE ve PMI standartlarında belirlenen çerçevede bilgiler içerecektir.

- Proje kapsamında uygulanacak proje yönetim metodolojisi genel çerçevesi ile proje süreçlerini özetleyecektir.

- Proje yönetim süreçleri içerisinde yer alan/alacak tüm proje ekibine ve proje paydaşlarına, proje kapsamında uygulanacak proje yönetim yaklaşımı ile proje süreçlerinde uygulanacak standart faaliyetler hakkında bilgiler sunacaktır.

- Genel proje ihtiyaçları kapsamındaki tüm süreç ve faaliyetler için standart ve tanımlı çerçeveleri kullanacaktır.

- Genel proje ihtiyaçlarına paralel olarak ortaya çıkan her projeye özel değişiklikler de dikkate alındığında, geliştirme süreçlerinin yaşayan/dinamik bir niteliğe sahip şekilde gelişen/güncellenen tüm proje ihtiyaçlarının karşılanmasında esneklik sağlayacaktır.

- Uygulanacak proje yönetim faaliyetlerinin,

- MSF/SDD, PMI ve IEEE 12207 standartlarına uygun olarak gerçekleştirilmesi,

- Rational, Prince 2, UML gibi farklı metodolojilerden de faydalanmasının mümkün kılması

- Onlarla bütünleşerek CMM, SPICE, ISO 9001:2000 vb. kalite yönetim sistemlerinin uygulamasını etkinleştirecektir.

Sistem günümüzdeki ve gelecekteki ihtiyaçlara cevap verecek şekilde;

- Kurye/ring seferleri de dâhil her türlü taşıma faaliyetinin;

- Süratle ve maliyet-etkin olarak planlanmasını,

▫ İcrasının pasif ve aktif yöntemlerle gerçek zamanlı olarak takibini,

- Mevcut ulaştırma kaynaklarının değerlendirilmesini,
- Her seviyedeki yöneticiye karar desteği sağlanmasını,
- Taşımaya esas bilgilere eş zamanlı erişim imkanı vererek paralel ve entegre planlama yapılması imkanı sağlayacaktır. Sistem, her seviyede yapılacak planlamalar ve icra sürecinin izlenmesi ile ilgili çalışan tüm birimleri kapsayacaktır.

## **1.2. Sistemin Yapısı**

Sistem; taşımaya esas malzeme bilgileri ve ulaştırma alt yapısının planlanmasına yönelik bütünlük ortak bir Veri Tabanı (VTYM), bu veri tabanını kullanarak her türlü ulaştırma faaliyetinin maksimum maliyet etkinliği sağlayacak şekilde planlama, programlama ve koordinasyonuna yönelik ulaştırma ihtiyaçlarının ortaya konması (TPM ve KURPM) ve yapılan planlamaları müteakip işlerliğin ve olası problem alanlarının belirlenebilmesi için gerçek zamanlı araç/malzeme iz takibinin yapılabilmesi (ETARİTM) amacıyla geliştirilecek modüllerden oluşacaktır.

### **1.2.1. Veri Tabanı Yönetim Modülü (VTYM)**

VTYM, sistemin diğer modülleri (KURPM, TPM, ETARİTM) tarafından kullanılacak, kullanıcının diğer mevcut ve geliştirilmekte olan lojistik ve komuta kontrol bilgi sistemleri ile veri değişimi yapabilecek, veri tabanını yönetecek, işleyecek, değiştirebilecek, güncelleştirebilecek fonksiyonlar içeren bir sistem olarak tasarlanmıştır.

VTYM, sistem kapsamındaki diğer modüllerin ihtiyaç duyabileceği taşınacak birim, yolcu, malzeme bilgileri, ulaştırma araç bilgileri, altyapı bilgileri, iz bilgileri, planlı kurye/ring seferleri bilgileri, plansız ulaştırma aracı tahsis bilgileri, planlı/plansız seferlere planlanmış yolcu/yük bilgileri, KURPM'de kullanılan ulaştırma araç/altyapı bilgileri, taşıma plan bilgilerini içerecektir.

### **1.2.2. Kurye/Ring Yolcu/Yük Planlama Modülü (KURPM)**

KURPM sayesinde; belli güzergâhlar dâhilinde sisteme gelecek sürekli taşıma istekleri değerlendirilerek rutin seferler düzenlenmesi, mevcut seferlerin geçmişteki istatistikî veriler ışığında belli periyotlar arasında değerlendirilerek kurye sisteminin optimizasyonunun sağlanması (mevcut seferlerin bacaklarının güncellenmesi, bir kuryenin iptali veya yeni bir kurye ihdas edilmesi) amaçlanmıştır.

### **1.2.3. Taşıma Planlama Modülü (TPM)**

TPM, genel olarak, veri tabanını kullanarak, taşımaların maliyet etkin bir şekilde planlanabilmesi, programlanabilmesi ve koordinasyonuna yönelik ulaştırma

ihtiyaçlarının ortaya konması için geliştirilen bir karar destek sistemi olarak tanımlanabilir. Bu çerçevede, TPM ile;

- Taşıma yeteneğinin maliyet etkin ve hızlı taşıma planlarıyla artırılması,
- Maliyet etkinliği,
- Taşıma modu/modları (ulaştırma çeşidi) etkinliği ve karar desteği,
- Kullanılacak ulaştırma aracı/araçları etkinliği ve karar desteği,
- Kullanılacak güzergâh ve ulaştırma altyapısı etkinliği ve karar desteği,
- Müşterinin taşınacak birimin son noktada olmasını arzu ettiği en son zaman (CRD- Customer's Required Date) bakımından taşıma zamanlama etkinliği,
- Planların icra esnasında dahi güncellenebilmesi,
- Her seviyede yapılan planlamaların maliyet etkinliği bakımından gerektiğinde birleştirilmesi, birleştirilecek planlamalarda ortaya çıkabilecek ulaştırma çatışmalarının giderilmesi,
- Yapılan planların simülasyonu ve animasyonunun sağlanması.

#### **1.2.4. Etkin Araç İz Takip Modülü (ETARİTM)**

ETARİTM, sistem kapsamında gerçekleştirilecek planlamaların işlerliğini ve olası problem alanlarının belirlenebilmesi için gerçek ve gerçeğe yakın zamanlı araç/malzeme iz takibinin yapılabilmesi sağlayacak alt sistem olarak tanımlanabilir.

Uluslar arası ortamda kabul görmüş bütünsel aktif ve pasif iz takip yeteneklerini içerecek şekilde geliştirilecek olan bu modül MAP-INFO yazılımı ve güncel/istendiği zaman güncellenebilir verilerini kullanacaktır. Aktif iz takip kitleri ile iletişim/haberleşme altyapısı kullanılarak, ETARİTM Sunucusuna otomatik olarak gönderilecek enlem, boylam, yükseklik, UTC zamanı, aracın hızı ve yönü gibi konum verileri sisteme otomatik olarak aktarılacaktır.

Sistem ayrıca; taşınacak malzemelerin cins ve adet bazında takibini sağlayacak, hareket halindeki araçların taşıdıkları malzemelere ilişkin bilgiler de pasif iz takip sistemi (indirme noktalarında kurulu barkod veya RFID sistemleri kullanılarak) kullanılarak otomatik güncellenecektir. Bu sayede hangi yük aktarma noktasına/depoya hangi malzemenin geldiği, hangi malzemenin ilgili yerden yüklenerek belirlenen diğer bir noktaya ulaştığı bilgilerini sunacaktır.

#### **1.3. Tanımlar, Kısa Adlar ve Kısaltmalar**

Bu bölüm EK-1'de sunulmuştur.

#### 1.4. Referanslar

- .... ile ..... arasında imzalanan ..... tarih ve ..... No'lu Sözleşme.
- Sistem Teknik ve İdari Şartnameleri (Bu çalışma kapsamında hazırlanmamış, ancak bu şartnamelerde yer alabileceği değerlendirilen tüm bilgiler ve özellikler "ihtiyaç özellikleri belirleme (SRS)" dokümanında (EK-....) sunulmuştur)
- IEEE Yazılım Geliştirme Standartları; "Institute of Electrical and Electronics Engineers. IEEE Standard for Software Development Plan, ANSI/IEEE Standard"
- IEEE 12207 Yazılım Yaşam Döngüsü
- PMI Rehber Dokümanı; "A Guide to the Project Management Body of Knowledge, Project Management Institute"

## 2. UYGULANACAK PROJE YÖNETİM FAALİYETLERİ

### 2.1. Proje Yönetim Yaklaşımı

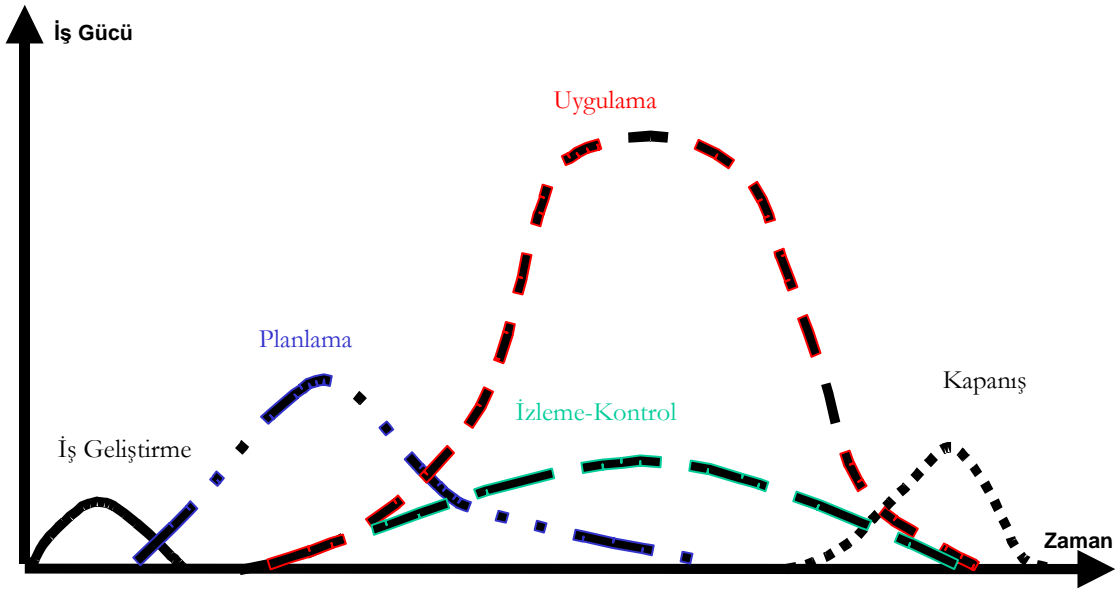
Bilişim projelerinde, standart ve başarısı test edilmiş PMI'nın "Proje Yönetimi Bilgi Ünitesi Rehberi (PMBOK- A Guide to the Project Management Body of Knowledge)" proje yönetim metodolojisi uygulanacaktır.

1969 yılında Amerika da kurulmuş olan PMI bugün 100.000 den fazla üyesi olan ve Proje Yönetimi eğitimi ve sertifikası veren bir kuruluştur. PMI tarafından geliştirilen PMBOK, bugün bilgi sistemleri proje yönetimi konusunda ANSI ve IEEE tarafından uluslararası bir standart olarak kabul edilmekte ve birçok sektörde proje yönetimi referans başvuru kitaplarından biri olarak kullanılmaktadır.

Zaman ve verimliliğin rekabet avantajı olduğu günümüzde, bilişim sektöründe gerçekleştirilen projelerin, doğru zamanda ve arzu edilen özellik ve niteliklerle teslim edilmesinde proje yönetiminin sağladığı sinerjinin yadsınamayacağı açıktır. Maalesef bilişim projelerindeki genel başarısızlık nedenlerinin başını proje yönetimi yetersizliği çekmektedir. Yazılım projeleri profesyonel bir şekilde planlanıp yönetilememekte ve her sürece verilen önem, o dönemdeki iş yoğunlukları ve eldeki kaynaklarla orantılı olmaktadır. Bu da projelerde, özellikle planlama, kontrol ve izleme gibi çok önemli süreçlere yeterli kaynak ve zaman ayıramamayı ve dolayısı ile projelerin kapanış süreçlerinde çok yoğun bir karmaşa ve kaynak problemi yaşanmasına neden olmaktadır. Bu dengesiz/kontrolsüz kaynak ve süreç dağılımı, hem ilgili projeyi hem de kaynağın verimsiz kullanımı nedeniyle devam eden ve/veya iş geliştirme aşamasındaki diğer projeleri etkilemektedir.

Şekil-E13.1’de, PMI tarafından belirlenen bu süreçlerin zamana ve harcanan iş gücüne yönelik bir grafiği sunulmuştur. PMI, projelerde kaynak/zaman planlamasının olabildiğince bu grafikteki dağılım doğrultusunda gerçekleşmesini, başarılı bir proje için ön koşul olarak belirlemektedir. Proje yaşam döngüsü boyunca sağlanacak tüm hizmetlerin proje hedeflerine uygun olarak gerçekleşmesinin en etkili ve verimli yolu;

- Proje ihtiyaçlarına cevap verebilecek, her biri kendi uzmanlık alanlarında güçlü bir deneyime sahip bir proje ekibi oluşturulması,
- Firma içerisinde her birimin proje kapsamındaki rol ve işbirliklerinin net olarak ortaya çıkarılması,
- Belirlenen rol ve sorumluluklar çerçevesinde her birimin proje yönetimine doğrudan katılımının sağlanabilmesidir.



**Şekil-E13.1:** Zaman ve harcanan iş gücüne göre PMI proje yönetim süreçleri (PMI 2000).

PMI standart ve yöntemleri tüm sektörlere olduğu gibi IT projelerine de uygulanabilecek bir çerçeve çizmekte ve etkili proje yönetimi için nelerin yapılması gerektiğini ortaya koymaktadır.

Proje kapsamında PMI’ın proje yönetim metodolojileri yanında standart ve başarısı test edilmiş MSF/SDD, IEEE 12207 metodolojileri de uygulanacaktır. Bu çerçevede, proje kapsamında tanımlanan hizmetlerin tatmin edici bir şekilde sağlanabilmesi ve proje ihtiyaçlarına etkinlik ve verimlilikle en üst düzeyde cevap

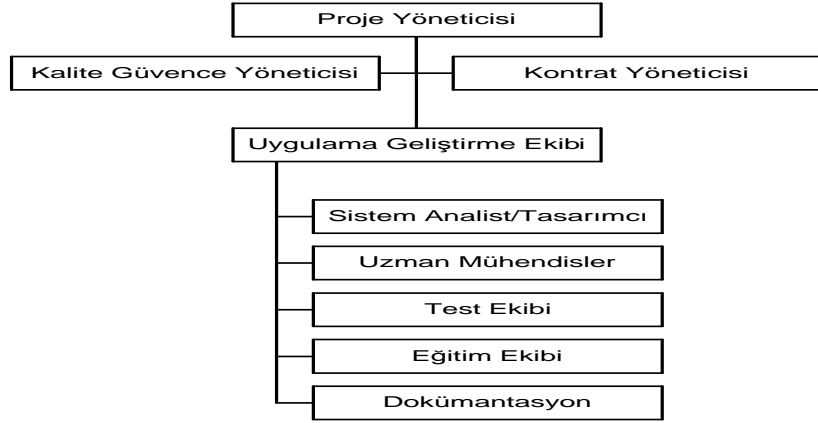


verilebilmesi maksadıyla, her biri kendi uzmanlık alanlarında güçlü bir deneyime sahip profesyonellerden yararlanılması gerekmektedir. Buna kapsamda uygulanacak proje yönetim metodolojisinin ana hedefleri aşağıdaki maddelerde özetlenmiştir:

- Proje ihtiyaçları ve iş tanımlarına uygun nitelik ve nicelikte bir proje ekibi oluşturulması ve bu ekipten optimum verimin sağlanabilmesi için uygun bir teşkilat yapısı oluşturulması,
- Projenin değişik aşamalarında farklı insanların yetenek ve tecrübelerini kullanmak için yöntemler geliştirilmesi,
- Projenin her aşamasında gerekli olan etkin proje yönetiminin ve operasyonel desteğin sağlanması,
- Erken problem saptama ve çözüm üretmek için yöntemler geliştirilmesi,
- Proje faaliyetlerinin zamanında uygulanmasının garantilenmesi,
- Proje kapsamındaki iş tanımlarının yönetilebilir ve kolay kontrol edilebilir alt bileşenlere ayrılarak, hangi aşamada hangi faaliyetlerin icra edileceğini net ve açık bir şekilde belirlenmesi,
- Sözleşme kapsamındaki hizmetler çerçevesinde gerçekleştirilen tüm çalışmaların kontrol edilmesi,
- Proje ekibi isteklendirmesinin ve takım ruhunun sürekli belirli bir seviyenin üzerinde olmasının sağlanması,
- Önceden onay yetkisinin belirlenmesi,
- Proje hedeflerine ulaşabilmek için proje yönetiminin en uygun teknolojik ürün ve yöntemler ile bütünleştirilmesi ve bu ürün ve yöntemlerin proje yönetimi süreci boyunca kullanılması.

Uygulanacak proje yönetim metodolojisinin de bir gereği olarak, projenin sözleşmede tanımlı amacına ulaşmasının temini, projeye dâhil olan her proje paydaşı için ana sorumluluktur. Bu amaçla, proje kapsamındaki yönetim faaliyetlerinin yürütülmesi ve desteklenmesi için gerekli olan her türlü kaynak ve insan gücünün temin edilmesi, ayrıca projede görevlendirilmiş personelin devamlılığının sağlanması, projenin başarısı için çok önemli ölçütlerdir.

MSF/SDD, PMI ve IEEE standartlarına uygun olarak gerçekleştirilen tüm projelerde olduğu gibi, projelerin gereklerine ve uygulanan yazılım geliştirme metodolojilerine uygun yapıda proje ekibi yapılması oluşturulacaktır. Örnek bir yapılanma Şekil-E13.2'deki gibidir.



**Şekil-E13.2:** MSF/SDD, PMI ve IEEE standartlarına uygun proje ekibi yapılanması.

Uygulanacak proje yönetim yapılanmasının bir gereği olarak, proje ekibinin tüm unsurları yapılan işin ve idareye/kullanıcıya teslim edilecek mal ve hizmetlerin ayrıntılı teknik ve profesyonel niteliğinden müştereken ve ayrı ayrı sorumlu tutulacaktır.

Proje kapsamında talep edilen hizmetlerle ilgili ekip niteliklerinin gücü yanında başarılı bir proje için gerekli olduğu değerlendirilen bir etken olan ekip üyeleri arasındaki takım ruhu düzeyinin sağlanması için şu temel özellikler sağlanacaktır:

- Proje ekibi buldukları görev yanında projenin ihtiyaç duyduğu iş tanımları ve hizmetler için geniş bir uygulama/gerçekleştirme deneyimine sahip uzmanlardan seçilecektir.
- Proje ekibinin her bireyi proje kapsamında tanımlı rol ve sorumluluklarıyla ilintili tecrübe, eğitim ve sertifikasyona sahip olacaktır.
- Proje ekibi sektöre yönelik projeler konusunda deneyimli ve sektörün ihtiyaç duyacağı diğer sertifikasyonu da karşılayabilecek niteliklere (güvenlik, standartlar, mevzuat, belge/form/raporlama, işleyiş, terminoloji, entegrasyon/birlikte çalışabilirlik) sahip olacaktır.
- Proje ekibi proje öncesinde ve/veya proje süresince aynı ekipte çalışmış/çalışacak personelden oluşturulmaya çalışılacaktır.

## 2.2. Proje Yönetim ve Uygulama Süreçleri

Proje yönetim faaliyetleri, MSF/SDD, PMI ve IEEE 12207 standardına uygun olarak “Döngüsel (Spiral) Model” ve “Şelale (Waterfall) Modeli” standart modellerini

birleştirecek ve gerektiğinde bütünleşik proje timi (IPT-Integrated Project Team) yaklaşımıyla hareket edecek bir yapı içinde gerçekleştirilecektir.

Uygulanacak proje yönetim süreçlerinin, PMI, MSF/SDD ve IEEE 12207 standartlarına da uygun olarak listelenmiş temel adımları aşağıda listelenmiştir;

- Sistem İhtiyaçları Analizi
- Sistem Tasarımı
- Uygulama Geliştirme
- Test, Entegrasyon ve Kurulum
- Eğitim
- Teknik Destek

Proje yaşam döngüsü süresince MSF/SDD, PMI ve IEEE 12207 metodolojilerinin uygulanabilmesini sağlayan ve özel olarak bu proje için geliştirilecek bir bilgi yönetim sistemi aracından yararlanılacaktır. Bu sayede proje ve konfigürasyon yönetimini bir bütün olarak uygulandığı saha ve her bir yazılım unsuru/bileşeni (komponent, ekran, rapor, süreç, rol, script vb) bazında sağlayabilen üstün bir yapı kurgulanmış olacaktır.

Uygulanabilecek proje yönetim ve uygulama metodolojilerinin benimsediği ve proje kapsamında kullanılacak temel anlayış, prensip ve esaslar aşağıda özetlenmiştir;

- **Kullanıcıya Özel Tasarım**

- Ancak kullanıcının ve/veya kullanım ortamların özel ihtiyaçlarına cevap verebilmek üzere tasarlanan çözümler ile gerçek bir performans artışına ve başarıya ulaşılabilir.

- Proje yönetim sürecinin önemli bir parçası konumundaki ihtiyaç analiz çalışmaları ve bu çalışmalar neticesinde ortaya çıkacak gerçek müşteri ve sistem ihtiyaçları tespitinin başarılı bir proje için ilk ve en önemli adımlardan biridir.

- Müşteri'nin her sürece katılımının sağlanması en üst düzeyde müşteri memnuniyetini elde etmek için çok önemli bir ön şarttır.

- **Müşteri Odaklı Yaklaşım**

- "Başarılı bilgi sistemi uygulamaları, ürün tedarikçilerinden daha çok kendini tam olarak projeye adanmış iş ortakları ile gerçekleştirilebilir.

- Müşteriler Çözüm Ortakları olarak yorumlanmalı ve proje ekibinin bir parçası konumuna getirilmelidir.

▫ Müşteri odaklı yaklaşım projenin tüm safhalarında sağlanmalı ve korunmalıdır.

- **Açık Sistem**

Müşterinin hem güncel ve hem de ileriye dönük ihtiyacını karşılayabilecek, en yüksek müşteri memnuniyetini sağlayan, bütünleştirmeye (integration), genişlemeye (extension) ve birlikte çalışmaya (interoperability) hazır nitelikte çözümler sunulmalıdır.

- **Uygulamada Esneklik**

Çözümler müşterilerin hayatlarını kolaylaştırmalı, onlara, sahip oldukları yetki ve rollere paralel olarak, gerektiğinde sistemi geliştirebilme, düzeltme yapabilme, çeşitli raporlar üretebilme, sorgulama yapabilme esnekliğini sağlamalıdır.

- **Ürünü Bilmek**

▫ Hizmet verilen müşteri ortamının ve gerçekleştirilen servis ve ürünün en ince detayına kadar bilinmesi gerekliliğinin projelerin başarısı ve müşteri memnuniyeti için çok önemli bir etkidir,

▫ Tüm proje yaşam döngüsü boyunca gerçekleştirilen servis/ürünlerin kalitesinin takip edilmesi gerekmektedir. Bu çerçevede aşağıda listelenen aktiviteler sürekli olarak uygulanmalıdır;

- ◆ Yüksek kaliteli ihtiyaçlar geliştirmek ve ihtiyaç gözden geçirmeleri gerçekleştirmek,

- ◆ Kodu erken ve sık sık test etmek,

- ◆ Müşteriden/Son kullanıcıdan gelen geri bildirimleri takip etmek, sürece dahil etmek,

- ◆ Hataların kökenindeki nedenleri ve yeni sorunlara karşı korunma analizi yapmak

- **Bilgilendirici Olmak**

▫ Uygulanacak yöntem, projenin tüm aşamalarında proje ortakları arasında doğru bilgi paylaşımını sağlamalıdır.

▫ Müşteri de proje takımında değerlendirilmeli ve bütün proje süreçlerine yönelik doğru bilgi akışı tüm proje paydaşları arasında sağlanmalıdır. Bu sayede, projelerin her aşamasında gerek duyulabilecek etkin ve zamanlı kararların, doğru bilgilere dayanarak, alınabilmesi sağlanmış olmaktadır.

▫ Projelerin her aşamasında, her bir proje ekibi, kendi sorumlulukları çerçevesinde, projenin gelişim süreçlerini takip etmelidir. Böylelikle

her proje adımının, planlandığı gibi gerçekleştirilmesi ve gerektiğinde uygun doğrultucu önlemlerin anında alınabilmesi sağlanabilir.

- **İşbirliği Yapmak**

Proje yönetimi tüm seviyelerde Müşterilerin (uzmanlardan, idareci, üst düzey yetkililere kadar tüm kullanıcılar) işbirliğini ve desteğini projenin başarısı için hayati bir önemde değerlendirmeli ve bu işbirliği ortamını projenin tüm yaşam döngüsü boyunca temin etmelidir.

### **2.3. Yazılım Geliştirme Metodolojisi**

Yazılım geliştirme ve uygulama yönetimi, ihtiyaçların, uygulanan teknolojilere bağlı olarak oldukça basit ve anlaşılabilir olduğu zamanlarda dahi başarılması çok güç ve değişken faktörlere bağlı bir sorumluluktur. Günümüz yazılım projelerindeki durumun ikilemini “The Standish Group” yetkilisi Jim Johnson “Bir proje çok nadiren teknik nedenlerle başarısız olur” sözüyle çok net özetlemektedir: Yüksek teknoloji, iyi eğitilmiş mühendisler ve kaynak gibi ihtiyacımız olan her şey elimizde olmasına rağmen yazılım projeleri halen başarısız olabilmektedir. Yine “The Standish Group” yazılım projelerinin ortalama % 189 fazla maliyetle bittiğini, projelere %94 oranında yeniden başlanıldığını, %222 zamanaşımına uğranıldığını, ancak tüm bunlara rağmen elde edilen sonucun, projenin başında saptanan hedefin ancak %61’i olabildiğini ifade etmektedir. Bu durum kabul edilebilir bir sonuç değildir. Projelerdeki başarısızlıkların temel sebepleri aşağıda listelenmiştir;

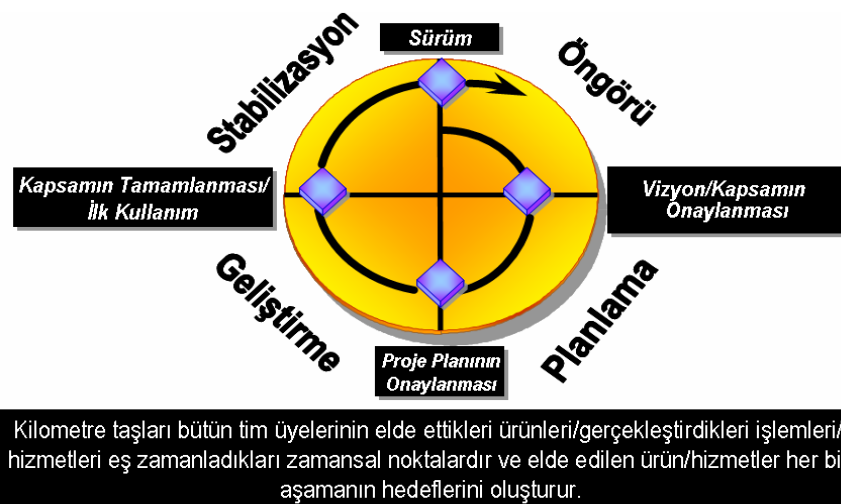
- Hedeflenen sonuç ile elde edilen fonksiyonun birbirinden ayrılması
- Teknoloji seçimindeki uyumsuzluk
- Zaman ve risk yönetiminin yapılamaması
- İletişim eksikliği ve takım oyunu olmayışı
- Adapte edilemeyen süreçler

Projelerin başarısı için çözüm, iyi ve test edilmiş, metodolojik yaklaşımların desteklendiği bir yazılım geliştirme altyapısının, uygulanabilir en iyi araçlar ve uygun en iyi çözümler ile birlikte sunulmasıdır. Bu özellikleri sağlayan Microsoft Solutions Framework/Software Development Discipline (MSF/SDD) başarılı yazılım projeleri gerçekleştirebilmek için oldukça etkin bir altyapı sunmaktadır. MSF/SDD’nin temel felsefesi “bir ürünü tamamlamak herkesin görevidir” olarak özetlenebilir. MSF/SDD tümüyle IEEE 12207 standartlarında tanımlı yazılım geliştirme süreçleri ile uyumlu, PMI proje yönetme yöntemlerini kullanmayı teşvik eden bir geliştirme disiplini.

MSF/SDD tarafından öngörülen uygulama modeli servis bazlı ve gelişmiş 3/n katmanlı sistem mimarisini öngören bir modeldir. Kullanıcı, iş ve veri servislerinin

yanında yazılımın kalıcı ve yaşayan bir sistem olmasını sağlayan iş akışı servisleri ve dinamik yapılar bu modelde desteklenmektedir. Uygulanan MSF/SDD metodolojisi, aşamalı yaklaşımlı (phased approach) ve iterative-spiral (tekrarlı-döngüsel) bir geliştirme süreci öngörür. Microsoft'un standartlaştırdığı bu yazılım geliştirme/proje yönetim metodolojisindeki ana amaç, hedeflenen yazılımı en kısa sürede, ilk etapta, temel ve çekirdek fonksiyonları ile oluşturabilmek ve kullanıma almak, gerçek ortamda, gerçek kullanıcılardan gelecek değerli geri bildirimlerden de yararlanarak, projenin diğer aşamalarında yeni sürümler ile tüm işlevsellikte ve tüm modüllerle hedeflenen uygulamayı tamamlayabilmektir. Bu yapısı ile uygulanan metodoloji, standart şelale modelinden ayrılır. Aşağıda uygulanabilecek iterative (tekrarlı) geliştirme yönteminin genel bir özeti verilmiştir:

- İlk etapta kullanıcı için en kritik ve öncelikli fonksiyonları içeren bir başlangıç sürümü hazırlanır (ilk sürüm). Böylelikle hem müşterinin öncelikli ve kritik ihtiyaçları temel fonksiyonlar ile karşılanır hem de ürünün diğer işlevselliği için gerçek kullanıcıdan gerçek ortamda geri bildirimler alınır,
- Alınan geri bildirimleri ve devam eden analiz/tasarım çalışmalarını temel alarak kaynak ve zaman planlaması güncellenir/gerçekleştirilir,
- Bir sonraki sürümler için tasarım ve geliştirme süreçleri planlanır,
- Geliştirmeler/uyarlamalar gerçekleştirilir,
- Sistem ve bütünleştirme testleri gerçekleştirilir,
- Yeni sürüm, yeni işlevler ve modüller ile hayata geçirilir.



**Şekil-E13.3:** Uygulanabilecek yazılım geliştirme metodolojisi.

Uygulanabilecek yazılım geliştirme metodolojisi, yukarıdaki şekilde de ifade edildiği şekilde “analiz”, “tasarım”, “geliştirme”, “test” ve “uygulama” gibi bir takım standart proje aşamalarından oluşmaktadır. Fakat MSF/SDD aşama yaklaşım (phased approach) ve iterative-spiral (tekrarlı-döngüsel) bir geliştirme süreci öngörmektedir, bunun anlamı, tüm proje aşamaları, proje yaşam döngüsü boyunca sürekli olarak tekrar eden ve devam eden faaliyetlerdir. Son/nihai ürün geliştirilene ve devreye alınana kadar analiz dâhil tüm tanımlı proje aşamaları döngüsel olarak devam etmektedir. Bu yaklaşımın temel amacı; yazılımı aşamalı/döngüsel olarak geliştirmek ve erken sürümlerle birlikte müşteri ve dolayısı ile son kullanıcının görüşlerini alarak, eski sürümlerin geliştirilmesi aşamasındaki alınan ders ve tecrübelerden de yararlanarak, yazılıma yönelik yeni sürümler gerçekleştirebilmektir. Bu yaklaşımda teslim edilen ara sürümler için bir onay mekanizması/onay aşaması yoktur, sadece kullanıcıdan gelen geri bildirim ve görüşler yeni sürümlere yansıtılır. Bu döngüsel ve aşamalı geliştirme süreci, nihai ürün teslim edilene kadar bu şekilde devam eder.

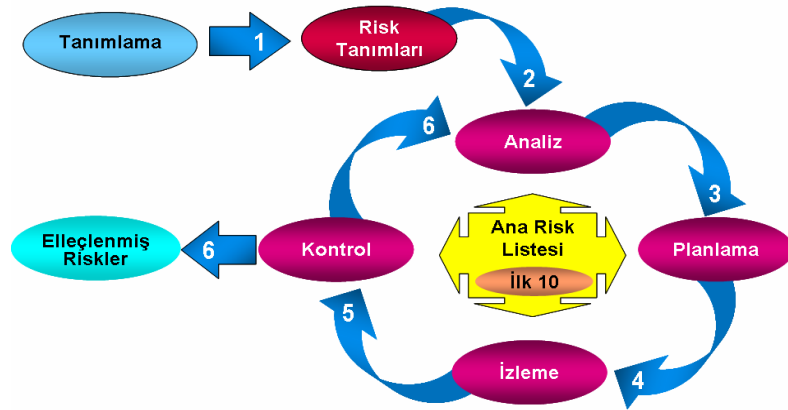
MSF/SDD yazılım geliştirme için yeni bir süreç, uygulama, ekip ve risk yönetimi modelini de beraberinde getirmektedir. Süreç modeli; Yourson, SSADM veya SDLC gibi klasik metodolojilerin aksine şelale değil spiral/döngülü bir model önermektedir. Süreç aşamalara bölünmüştür ve kritik kabul edilen noktalarda kilometre taşı esaslı çalışmaktadır. Kullanıcı, iş ve veri servislerinin yanında yazılımın kalıcı ve yaşayan bir sistem olmasını sağlayan iş akışı servisleri ve dinamik yapılar bu modelde desteklenmektedir. Ekip modeli (Şekil-E13.4), küçük ancak değişik disiplinlerden oluşan, birbirine bağımlı rolleri ve paylaşımlı sorumlulukları olan bir ekibi öngörmektedir. Ekibin derin teknik bilgisinin yanında iş sezgisinin olması, işin yapılmasına ve ürünün tamamlanmasına odaklanması, net hedef ve amacın varlığı ve kullanıcı/müşteri katkısının sağlanması ekip oluşumu ve işleyişinde önem arz etmektedir.



Ekibin modelinde bir kuruluş şeması yoktur. Timlerin görevleri birbirlerine bağımlılık gösterir ve tim liderleri proje yönetim sorumluluğunu paylaşırlar.

**Şekil-E13.4:** Ekip Modeli

Jim McCarthy "Yazılım Geliştirme Dinamikleri" adlı kitabında "En başarılı projelerin bile neredeyse her aşamasında, çok sayıda ve çok önemli bilinmeyen hususlar söz konusudur." demektedir. Bu noktadan hareketle, MSF/SDD, tepkili (reactive) veya geçişli bir risk yönetim modeli yerine önaktif (proactive) bir risk yönetimini öngörmektedir (Bkz. Şekil-E13.5). Risk yönetilmeden önce çok net olarak tanımlanır, tanımlanan risk ise karar vermeyi sağlayacak olan zamanlama, olasılık, etki matrisine dönüştürülür ve nihayetinde toplam risk tahminlenebilir/bilinebilir hale getirilir.



Risk tanımlama ve yönetimi projenin tüm aşamalarında gerçekleştirilir.

**Şekil-E13.5.:** Önaktif Risk Yönetim Süreci.



MSF/SDD yazılım geliştirme metodolojisi, tümüyle PMI proje yönetim süreçlerini kullanmayı tetikleyen ve ISO 12207 süreçleri ile uyumlu bir metodolojidir ve Rational Firması Birleştirilmiş/Tekleştirilmiş Süreci (RUP-Rational Unified Process), Prince2, Birleştirilmiş/Tekleştirilmiş Modelleme Dili (UML-Unified Modelling Language) gibi metodoloji ve araçların yanında CMM, SPICE, ISO 9000:2000 gibi çok sayıda kalite yönetimi sisteminin de kullanımını teşvik etmekte ve önermektedir.

#### **2.4. Genel Proje Süreçleri**

Proje aşamaları aşağıda açıklandığı şekilde tanımlanabilir:

- Kullanıcı İhtiyaçlarının Analizi ve Bilgi Toplama Faaliyetleri,
- Detaylı Çalışma Planı ve Takviminin Hazırlanması,
- Sistem Mühendisliği Çalışmaları (Sistem Analizi ve Tasarım Çalışmaları),
- Ana Mimarının (yazılım, donanım, veri tabanı, iletişim altyapısı vb.) oluşturulması,
- Veri Tabanı Kavramsal, Fiziksel, İlişkisel Veri Modellerinin Geliştirilmesi,
- Veri Sözlüğünün Geliştirilmesi,
- Taslak Modül Kullanıcı Ara Yüzlerinin Geliştirilmesi,
- Uygulama Yazılımlarının Geliştirilmesi,
- Sisteme Örnek ve Gerçek Verilerin Girilmesi,
- Test, Değerlendirme, Geçerleme ve Doğrulama Faaliyetleri,
- Dokümantasyonun Nihai Hale Getirilmesi,
- Sistem Kullanıcı Eğitimlerinin Verilmesi,
- Projenin tamamlanması ve müteakiben işletim ve idamesinin belirlenecek garanti süreleri içerisinde gerçekleştirilmesidir.

Uygulanacak yazılım geliştirme metodolojisi (Şekil-E13.6), standart “waterfall” (şelale) metodolojisinde olduğu gibi, aşağıda genel özellikleri sunulan “analiz”, “tasarım”, “geliştirme”, “test” ve “uygulama” gibi standart proje aşamalarından oluşacaktır. Tüm bu süreçlerde ve ilgili dokümantasyonda proje ihtiyaçlarının öncelikli olarak karşılanması esas olacaktır.



kullanıcı ihtiyaçları, güvenlik, bütünleştirme ve birlikte çalışabilirlik (interoperability) ihtiyaçları, ara yüz, işleyiş ve bakım ihtiyaçları, eğitim, destek, dokümantasyon ve kalite ihtiyaçları),

- Yeni sistemin birbirleri ile ve mevcut sistemler ile ilişkilerinin, mevcut bilgi akışı ve ilgili başka kurumlar arasındaki bilgi akışlarının belirlenmesi,
- Gerekli faaliyet, süreç ve veri modellemelerin yapılması, mevcut veri sözlüklerine ve işleyen sistem ve süreçlere uygun olarak bilgi akışı, bilgi formatı ve veri sözlüklerinin hazırlanması;
- Proje kapsamındaki uygulamalara yönelik mevcut işleyişler ve iş akışları, manuel yapının ayrıntılı olarak analiz edilmesi, mevcut tüm süreçlerin gözden geçirilmesi ve anlaşılması,
- Mevcut uygulamalarını projede tanımlanan özelliklere ulaştırmak için ihtiyaç duyulacak düzenlemeler, ilave fonksiyonlar, yapı ve mekanizmaların tanımlanması,
- Mevcut sistemlerden tasarlanan sistemlere geçiş planının hazırlanması.

#### **2.4.2. Sistem Tasarımı Aşaması**

Sistem Tasarımı aşamasında sistem ve kullanıcı ihtiyaçları doğrultusunda gerçekleştirilen sistem ihtiyaçları analizi çalışması çıktılarından yararlanılarak, Sistem Mimari Tasarımı oluşturulması hedeflenir. Bu çerçevede projelerin ihtiyaç ve kapsamına göre çeşitlilik gösterse de, genel olarak aşağıdaki faaliyetler gerçekleştirilir;

- Öncelikle sistemin üst seviye (high-level) mimarisi tasarlanır. Bu süreçte tasarlanan sistem ile ilgili donanım, yazılım ve işletim elemanları ile ilgili konfigürasyon ve performans hususlarının belirlenmesi,
- Tasarlanan sistem elemanları ile sistem ihtiyaçları arasında izlenebilirlik sağlamaya yönelik bir izleme matrisi oluşturulması,
- Sistem ihtiyaçları ile tutarlılığın takip edilmesi,
- Sistemin güvenlik özellikleri yanında, sisteme yönelik tehditlere (virüs, solucan, veri madenciliği girişimleri, iz takibi) karşı icra edilecek faaliyetlerin (replication) ve yedekleme özelliklerinin tanımlanması.
- Tasarım standartlarının ve metodlarının uygunluğunun değerlendirilmesi,
- Donanım ve yazılım kalemlerinin sistem ihtiyaçlarına uyumluluğunun tanımlanması.

- Yazılımla ilgili menü, ekran ve rapor yapıların hazırlanması ve kullanıcı ara yüz tasarımının gerçekleştirilmesi,
- Her bir modül için detay tasarım çalışması yapılması ve modüller arası, fonksiyonlar ve birimler arası veri ilişkilerinin belirlenmesi,
- Veri yapıları ve veri tabanı detay tasarımının gerçekleştirilmesi,
- Bütünleşik mimarinin tasarlanması,
- Veri tekleştirme, dönüştürme, eşleştirme ve birleştirme çalışmalarına yönelik yaklaşımların belirlenmesi,
- Test senaryolarının ve Test Planı ilk sürümünün hazırlanması,
- Kullanıcı dokümanlarının hazırlanması,
- Uygulanan yazılım geliştirme standartlarına uygun rutin gözden geçirme süreçlerinin IEEE standartları rehber alınarak gerçekleştirilmesi,
- Tüm sistem tasarım sürecinin müşteri ve son kullanıcı ile tam bir işbirliği çerçevesinde gerçekleştirilmesi.

#### **2.4.3. Uygulama Geliştirme ve Test Aşaması**

Uygulama geliştirme aşamasında sistem analiz ve sistem tasarım aşamaları sonuçlarından faydalanılarak uygulama yazılımının geliştirilmesi sağlanacaktır. Geliştirme aşaması; kodlama, derleme, hata ayıklama, ara yüzlerin oluşturulması, veri tabanlarının tanımlanması gibi temel faaliyetleri kapsayacaktır.

Kodlama sırasında her yazılım birimine yönelik konfigürasyon yönetimi gerçekleştirilecektir. Bu çerçevede projelerin ihtiyaç ve kapsamına göre çeşitlilikler gösterse de genel olarak aşağıdaki faaliyetler gerçekleştirilecektir:

- Her yazılım birimi belirlenen ayrıntılı tasarım çalışması uyarınca geliştirilecektir.
- Her yazılım birimine yönelik test işlemleri ve test verileri geliştirilecektir.
- Her yazılım birimi önceden belirlenen ihtiyaçları karşılayabilmeleri bakımından teste tabi tutulacaktır.
- Yazılım bütünleştirme plan ve takvimi belirlenecektir.
- Yazılım kodları ve test sonuçları gözden geçirilecek ve belirlenen kalite standartları ve işlevsel ihtiyaçlar paralelinde IEEE standartları rehber alınarak değerlendirme yapılacaktır.
- Bütünleştirme planı doğrultusunda sistemin tüm bileşenlerinin bütünleştirilmeleri gerçekleştirilecektir.

- Sistemin Modül/Entegrasyon Testleri ile Alfa/Beta Sistem testleri gerçekleştirilecektir.

#### 2.4.3.1. Modül Bütünleştirme Testleri

Modül testleri sistemin en küçük test edilebilir birimlerine yönelik gerçekleştirilen testlerdir. Modüller arasında bilgi paylaşımı ile ara yüzleri doğrulamak için Modül-Modül, Program-Program, Alt sistem-Alt sistem arasında gerekli testler gerçekleştirilecektir.

**Tablo-E13.1: Modül Bütünleştirme Testleri Genel Özellikleri**

|                              |  |
|------------------------------|--|
| Testin Amacı                 | <ul style="list-style-type: none"> <li>• Birim, Modül ve/veya Alt Sistem içerisindeki sorunları ve bu sistem bileşenlerinin birlikte çalışabilirliklerinin,</li> <li>• Bir sistem bileşenine yönelik aşağıdaki sonuçların tespiti amaçlanmaktadır: <ul style="list-style-type: none"> <li>▫ Tam ve eksiksiz mi?</li> <li>▫ Tanımlı fonksiyonlarını yerine getiriyor mu ?</li> <li>▫ Tasarlandığı gibi çalışıyor mu?</li> <li>▫ Sürüme hazır mı?</li> </ul> </li> </ul> |
| Ön Şartlar/<br>Özel Durumlar | <ul style="list-style-type: none"> <li>• Sistem bileşeninin fonksiyonları tanımlanacak,</li> <li>• Sistem bileşeninin arayüz ihtiyaçlarının belirlenecek,</li> <li>• Sistem bileşeninin girdi/çıkı ilişkisi tanımlanacak,</li> <li>• Sistem bileşeninin tasarımı tamamlanmış olacak,</li> <li>• Test ekibi ve test ortamı hazır olacaktır.</li> </ul>  |
| Sorumlu Ekip                 | Modül/Bütünleştirme testleri yazılımı geliştiren proje ekibi tarafından gerçekleştirilecektir.   |
| Tamamlanma<br>Ölçütleri      | <ul style="list-style-type: none"> <li>• Tüm planlanan testler tamamlanmış olacak,</li> <li>• Tespit edilen tüm sorunlar belirlenmiş ve düzeltilmiş olacak,</li> <li>• Sistem bileşenine yönelik hiç bir sorunla karşılaşılması esas kabul edilecektir.</li> </ul>   |

#### 2.4.3.2. Alfa (Alpha) Testi

Alfa testi uygulama yazılımının prototipinin oluşturduğu aşamada gerçekleştirilen sistem testleri olarak tanımlanır. Bu aşamada, uygulama yazılımı tüm hedeflenen özelliklere sahip olmamakla birlikte ana işlevleri gerçekleştirilebilmektedir ve sisteme yönelik girdi/çıkı ilişkilerini gözlemlemek mümkündür.

**Tablo-E13.2:** Alfa Testi Genel Özellikleri

|                                 |  |
|---------------------------------|--|
| <i>Test Amacı</i>               | <ul style="list-style-type: none"><li>• Tüm sistem hatalarının ayıklanması (Debugging)</li><li>• Beta (Kurulum) testine hazırlık yapılması amaçlanmaktadır.</li></ul>  |
| <i>Ön Şartlar/Özel Durumlar</i> | <ul style="list-style-type: none"><li>• Uygulamanın modül ve entegrasyon testlerini başarı ile geçmesi,</li><li>• Uygun nitelikte bir test takımının ve üretim ortamının oluşturulması sağlanacaktır.</li></ul>  |
| <i>Sorumlu Ekip</i>             | <ul style="list-style-type: none"><li>• Alpha testleri yazılımı geliştiren proje ekibi tarafından gerçekleştirilecektir.</li></ul>   |
| <i>Tamamlanma Ölçütleri</i>     | <ul style="list-style-type: none"><li>• Kabul test senaryoları hazırlanmış ve Müşteri tarafından onaylanmış olacak,</li><li>• Tüm sistem testleri başarı ile gerçekleştirilmiş olacak,</li><li>• Sistem bileşenine yönelik sorunlar dâhil tüm sorunlar önceden belirlenmiş ve düzeltilmiş olacaktır.</li></ul> |

#### 2.4.3.3. Beta Testi

*Beta testleri, uygulama yazılımı hemen hemen hedeflenen tüm işlevselliği ile çalışmaya hazır olduğunda, kullanıcı geri bildirimlerini almak amacıyla gerçekleştirilen kurulum testleri olarak tanımlanabilir. Beta testleri, geliştirilen uygulama yazılımının devreye alınmadan önce gerçek kullanıcılarca ve gerçek kullanıcı ortamında test edilebilmesine imkân sağlar.*

**Tablo-E13.3:** Beta Testi Genel Özellikleri

|                                 |   |
|---------------------------------|---|
| <i>Test Amacı</i>               | <ul style="list-style-type: none"><li>• Uygulama yazılımının kullanıcı ortamında test edilmesinin sağlanması,</li><li>• Kullanıcıya, sistem devreye girmeden, görüş ve geri bildirimlerini yansıtılması,</li><li>• Sistemin kabul sürecine ve devreye alınmaya hazır olup olmadığının denetlenmesi amaçlanmaktadır.</li></ul> |
| <i>Ön Şartlar/Özel Durumlar</i> | <ul style="list-style-type: none"><li>• Uygulamanın alfa testlerini başarı ile geçmesi,</li><li>• Uygun nitelikte bir test takımı ve üretim ortamı tesis edilmesi sağlanacaktır.</li></ul>  |
| <i>Sorumlu Ekip</i>             | Beta testleri yazılımı geliştiren proje ekibi ve Müşteri  |

|                           |  |
|---------------------------|--|
|                           | test ekibi tarafından gerçekleştirilecektir.   |
| <i>Tamamlanma Kriteri</i> | <ul style="list-style-type: none"><li>• Kabul test senaryoları hazırlanmış ve Müşteri tarafından onaylanmış olacak,</li><li>• Tüm beta testlerinin başarı ile gerçekleştirilmesi sağlanacak,</li><li>• Sistem bileşenleri dâhil tüm sorunlar önceden belirlenmiş ve düzeltilmiş olacak,</li><li>• Sistem kabul sürecine ve devreye alınmaya/kullanıma sunulmaya hazır olacaktır.</li></ul> |

### 3. KALİTE GÜVENCE ve RİSK YÖNETİMİ

#### 3.1. Yazılım Kalite Güvencesi

Bilgi teknolojilerinin üretimi, geliştirilmesi, devreye alınması, bütünleştirilmesi ve işletimi alanlarında;

- Kullanıcının her düzeyde bilgi yönetim ihtiyacını karşılayacak IT çözümleri geliştirilmesi ve daha hızlı, kaliteli, etkili ve verimli çalışmaları sağlanması,
- Geliştirilecek bilgi sistemleri ile doğru bilgi, doğru zamanda, doğru seviyede, doğru kişiye sürekli olarak iletilmesi,
- Müşterilerinin iş yapma biçimini belirleyen ve işlerinin ölçülebilir olmasını sağlayan bilgi sistemleri kurulması,
- Gerektiğinde yeni teknolojiler üretilebilmesi,
- Geliştirilecek bilgi sistemlerinin, sadece günümüz değil, gelecekteki ihtiyaçlarını da karşılayabilecek şekilde tasarlanması Geliştirici tarafından esas kabul edilmektedir.

PMI, MSF/SDD, IEEE 12207 ve ISO 9001:2000 standartları ile uyumlu ve bütünleşik olarak bir Kalite Yönetim Sistemi oluşturulması, belgelenmesi, sürekliliğin ve sürekli iyileştirmenin sağlanması için gerekli mekanizmalar kurulacaktır. Kalite Yönetim Sistemi için gerekli süreçler geliştirici tarafından tanımlanmış ve organizasyon içerisinde uygulanması sağlanmıştır. Bu süreçlerin sıralanışı ve birbirleri ile olan ilişkileri belirlenmiş, ayrıca etkin olarak işlemeleri ve kontrollerin sağlanması için gerekli yöntem ölçütleri belirlenmiştir. Bu bağlamda,

- Gerekli bilgi ve kaynakların temini ve denetlenmesi,
- Planlanan sonuçlara ulaşılması için gerekli faaliyetlerin sürekli iyileştirilmesi,

- Kalite yönetimi kapsamı içerisinde tedarikçilere yaptırılan işlerin de bu süreçler ile kontrol edilmesi sağlanmaktadır.

### 3.2. Risk Yönetimi

Günümüzde yazılımların “yaşayabilme” (survivability) ve “idame” (sustainability) kavramları Yazılım Mühendisliği'nin en önemli konularını oluşturmaktadır. En pozitif istatistiklerde dahi yazılım projelerindeki başarı %25'i geçmemektedir. Yine istatistiklere göre bu başarısızlığın en temel sebebi (%43) kullanıcı ihtiyaçlarının sürekli olarak değişmesi ve sistemlerin bu değişime ayak uyduramaması ve yeni istekleri karşılayamamasıdır. Donanımdan kaynaklanan başarısızlık %6, performans problemleri ise %4 civarındadır. Nesne tabanlı teknolojiler, 3 katmanlı mimariler, hızlı prototipleme, spiral metodolojiler, önaktif risk yönetimi uygulanarak idamesi sağlanabilen yazılımların geliştirilmesinde büyük fayda sağlamakla birlikte, bazen yetersiz kalabilmektedirler. Yazılımda “hayatta kalabilirlik” ve “süreklilik” için tüm bu teknolojilere ve yaklaşımlara ilave olarak sağlanması gereken yaklaşım “süreç esaslı (process based)” yaklaşımlar ve “iş akışlarınınca yönlendirilen (workflow oriented)” teknolojilerdir.

Proje kapsamında, PMI, MSF/SDD, IEEE 12207 ve IEEE Std 730-2002 Yazılım Kalite Planı standartlarına uygun olarak risk yönetimi süreçleri uygulanacaktır. Proje risklerinin tanımlandığı, çözümlendiği ve önlemlerin alındığı tüm faaliyetleri tanımlayan risk yönetiminin ana amacı, projede olabilecek riskleri önceden belirlemek, bu risklerin olma ihtimallerini azaltacak tedbirler geliştirmek, risklerin gerçekleştiği anlarda ise etkilerini önleyici ve azaltıcı önlemleri planlamak ve uygulamak olarak özetlenebilir. Risk yönetim süreci uluslararası standartlara paralel olarak; planlama, risk tanımlama, risk niceleme, risk niteleme, riske tepki geliştirme ve risk gözlemlene ve kontrol adımlarını içermelidir.

Risk Yönetim süreci içerisindeki ana faaliyetler ve bu süreçlerde genel olarak kullanılacak araçlar aşağıdaki maddelerde özetlenmiştir.

#### Risk Yönetimi Planlaması

Risk yönetimi planlaması bir proje yaşam döngüsü boyunca risk yönetim sürecine yönelik faaliyetlerin nasıl gerçekleştireceğine yönelik planlamaların yapılacağı aşamadır. Risk yönetimi planlaması proje kapsamında öngörülen risklerin tanımlanması, tehdit düzeylerinin ve görülebilirlik olasılıklarının tespiti, genel bir önlem yaklaşımı ve mekanizmaların tanımlanması adımlarını içerecektir (Şekil-E13.7.).





Şekil-E13.7.: Risk Yönetimi Planlaması

### 3.2.1. Risk Tanımlama

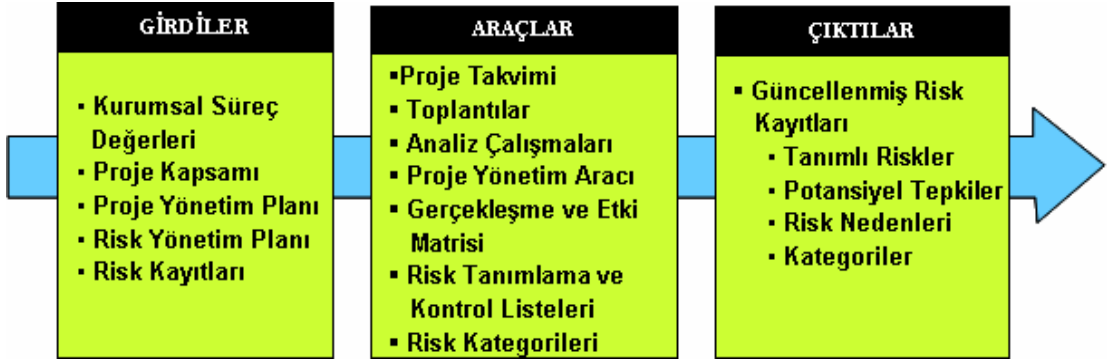
Risk tanımlama aşamasında hangi risklerin projeyi etkileyeceği belirlenecek ve bu risklerin genel karakteristikleri tanımlanacaktır. Tüm proje paydaşlarından risklerin tanımlama sürecine katkıda bulunması beklenen bu aşamaya; Proje Yöneticisi, Sistem Analistler, Yazılımcılar, Kalite Güvence Sorumlusu, Müşteri proje ekibi ve bazı son kullanıcılar katılacaktır. Risk tanımlama, proje yaşam döngüsü boyunca tekrarlamalar ile sürekli devam eden ve yaşayan bir faaliyet süreci olarak yürütülecektir (Şekil-E13.8.).



Şekil-E13.8.: Risk Tanımlama.

### 3.2.2. Risk Niteleme Analizi

Risk niteleme analizi aşamasında, tanımlanan risklerin gerçekleşebilme olasılıklarına, muhtemel etkilerine ve etki alanlarına, tolerans ve önlem yöntemlerinin zorluğuna göre risklerin derecelendirilmesi faaliyetleri gerçekleştirilecektir. Risk niteleme sürecine; Proje Yöneticisi, Sistem Analistleri, Yazılımcılar, Kalite Güvence Sorumlusu ve Müşteri proje ekibinin katılımı sağlanacaktır (Şekil-E13.9.).



Şekil-E13.9: Risk Niteleme/Niceleme Analizi

### 3.2.3. Risk Niceleme Analizi

Risk Niceleme analizi aşamasında, risk niteleme analizi süresince öncelik değerlendirmesi gerçekleştirilen risklerin analizi gerçekleştirilecek ve bu risklere proje süreç ve kaynaklarına olabilecek etkilerine yönelik sayısal değerler atanacaktır. Risk niceleme sürecine; Proje Yöneticisi, Sistem Analistler, Yazılımcılar ve Kalite Güvence Sorumlusu ve Müşteri proje ekibinin katılımı sağlanacaktır (Şekil-E13.9.).

### 3.2.4. Risk Tepki Planlaması (Karşı Planlama)

Risk tepki planlaması, proje amaçlarına ulaşma şansını arttıracak ve tanımlı risklerin gerçekleşme olasılığı ile gerçekleşen tehditlere yönelik önleyici ve azaltıcı tedbirlerin alınmasına yönelik planlama ve adımları içerecektir. Risk tepki planlaması hem proje kaynaklarına hem de tanımlı riskin tipi ve oluşturabileceği tehditlere uygun olarak gerçekleştirilecektir. Risk tepki planlaması sürecine; Proje Yöneticisi, Sistem Analistler/Tasarımcılar, Yazılımcılar ve Kalite Güvence Sorumlusu, test sorumluları ve Müşteri proje ekibinin katılımı sağlanacaktır (Şekil-E13.10.).

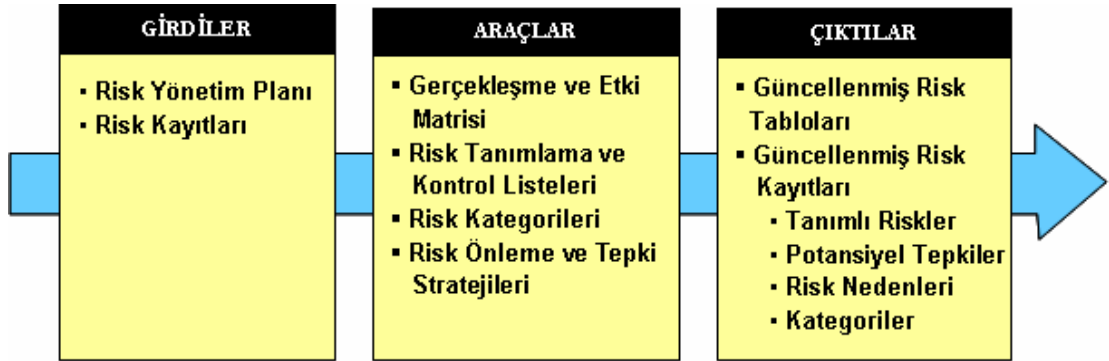


Şekil-E13.10: Risk Tepki Planlaması

### 3.2.5. Risk Gözleme ve Kontrol

Risk gözleme ve kontrol sürecinde, mevcut riskler, gerçekte olasıları, etkileri ve önlem mekanizmaları takip edilecek ve önceden öngörülme yeni riskler belirlendiğinde nitelik/nicelik tanımlamaları sürecine dâhil edilecektir. Bu sürece; tüm proje paydaşlarının katkıda bulunması için girişimlerde bulunulacak ve Proje Yöneticisi, Yazılımcılar, Kalite Güvence Sorumlusu, Test sorumluları ve Müşteri proje ekibinin katılımı sağlanacaktır. Risk Gözleme ve Kontrol sürecinden beklentiler özetle şu şekildedir (Şekil-E13.11.):

- Proje hedef ve varsayımlarının halen geçerli olduğunun tespiti;
- Belirlenen risklerin nitelik/nicelik deęerlendirmelerinde yaşanabilecek deęişikliklerin gözleme,
- Risk takibinin Risk Yönetim Planına uygun olarak gerçekteştiğinin tespiti,
- Önleyici ve/veya etki azaltıcı önlemlerin izlenmesi etkilerinin belirlenmesi.



Şekil-E13.11: Risk Gözleme ve Kontrol

### 3.3. Konfigürasyon Yönetimi

Konfigürasyon bir ürünün fiziksel ve fonksiyonel tanımlaması anlamına gelmektedir. Konfigürasyon Yönetimi süreci, gerçekteştirilen ürünlerin bütün yaşam döngüsü boyunca hem fiziksel hem de işlevsel olarak şekillendirilmesinin izlenmesi ve kontrol edilmesi olarak deęerlendirilmektedir. Konfigürasyon Yönetimi, kalite yönetimi kapsamındaki bütün öğeleri geliştiren, üretim ve destek yaşam sürecine tekniksel ve yönetsel yaklaşım sergileyen bir yönetim disiplindir.

Tüm proje yaşam döngüsü içerisinde gerçekteştirilen Konfigürasyon Yönetimi faaliyetleri, gayret tekrarını önleyecek ve planlanan, devam eden etkinlikleri kontrol

altında tutarak; gerektiğinde tüm proje paydaşlarına tam ve doğru bilginin iletilmesini sağlayacaktır. Konfigürasyon Yönetimi sürecinde geliştirilen ürün ile ilgili bütün veri ve değişiklikler kaydedildiği için ürünün yeniden yapılanması süreci minimuma indirilmiş olacaktır.

Hizmet/ürün oluşturma aşamasında artan müşteri talepleri ve proje ihtiyaçları en kısa zamanda tam ve doğru olarak karşılanırken Konfigürasyon Yönetim sürecinin;

- Kaynak planlanması,
- Kaynak zamanlanması,
- Maliyet yönetimi,
- Proje yönetimi desteği,
- Ürün yaşam döngüsü boyunca gelişim ve bütünlüğünün kayıt altında

tutulması konularına odaklanması sağlanacak, yöneten bir disiplin ve süreç olarak yorumlanıp değerlendirilecektir.

Projede uygulanabilecek Konfigürasyon Yönetim süreçlerinin temel adımları aşağıda sıralanmıştır:

- Ürünün geliştirilme sürecinde ortaya çıkan/yapılan değişikliklerin ve bu değişikliklerden etkilenen öğelerin doğru ve tam olarak tanımlanması sağlanacaktır.
- Gerçekleştirilen tüm değişiklikler kayıt altına alınacak, yönetimi, planlanması ve kontrolü gerçekleştirilecektir.
- Ayrıntılı ve tam değişiklik tarihleri izlenecektir.
- Tekrarların önlenmesi sağlanacaktır.
- Ürün geliştirme durumunun devamlı kaydı tutulacak ve sapmalar raporlanacaktır.

#### **3.4. Eğitim**

Edinilen tecrübeler, eğitilmiş personelin proje başarısındaki katkısının çok büyük olduğunu göstermiştir. Sistemleri işleten, yöneten ve geliştirilen sistemin bakımı ve idamesini sağlayacak olan iyi eğitilmiş personeldir. Yeterli sayıda yetişmiş personel tarafından yönetilmeyen, işletilmeyen sistemler başarısız olmaya mahkûmdur.

Bilgi sistemi ortamlarında sistemin çok sayıda kullanıcı tarafından işletilmesi gerektiğinden, sistemin etkin bir şekilde kullanımı için bu kullanıcıların belli bir seviyede sağlıklı bir kavramsal ve uygulamalı eğitim alması gerekecektir. Bilgisayar ile henüz tanışmamış kullanıcılar için uygulama yazılımlarına yönelik eğitimler

öncesinde bilgisayar okuryazarlığı eğitimi verilecektir. Kurum ortamında personel değişiklikleri (işten ayrılma, emeklilik, tayin, görev değişikliği) söz konusu olabildiğinden her ne kadar bu hususun enazlanması sağlanmaya çalışılsa da olası durumlar dikkate alınarak eğitim ihtiyacının karşılanması konusunda bir süreklilik sağlanacaktır.

Eğitim süreci her proje için, özellikle sistemin idamesinin sağlanabilmesi açısından, son derece kritiktir. Bu nedenle, planlamasının müşteri ile birlikte, güncellenen son ihtiyaçları yansıtacak şekilde yapılması gerekmektedir. Bu nedenle ilgili birimlerin eğitimlere düzenli olarak katılması sağlanacak ve eğitim planlaması yapılırken, proje takvimi ile birlikte, eğitilecek personelin mevcut iş yükü ve zaman planlaması da dikkate alınacaktır.

Açıklamalı eğitim uygulama planı ve belirlenmiş müfredat için EK-14 Eğitim Uygulama Planı dokümanına bakınız.

### **3.5. Teknik Destek ve Problem Çözümleme/Takip**

Geliştirilecek proje yönetim ve iş takip aracı ile müşteriden gelen ve/veya proje uygulama, geliştirme ve test ekipleri tarafından tespit edilen sorun ve/veya yeni uyarılma/geliştirme isteklerinin yerine getirilme süreçlerinin tümüyle yönetilmesi sağlanacak, kayıt altına alınabilen ve kontrol edilebilen çok amaçlı ve etkin bir araç olarak kullanılacaktır. Bu araç tümüyle MS. NET ile geliştirilecek ve MSF/SDD ve IEEE 12207 tarafından öngörülen yazılım geliştirme yaşam döngüsünü tümüyle destekleyecek özellik ve nitelikte olacaktır. Bu sayede gerek yerel gerekse dağıtık ekipler için birlikte çalışabilirliğe imkân sağlayan son derece etkin bir yazılım geliştirme ve yönetim ortamı sağlamış olacaktır.

### **3.6. Dokümantasyon**

Projenin süresi boyunca üretilen doküman ve raporlar için kullanılacak standart bir doküman yapısı oluşturulacaktır. Standart dokümantasyon formatına sahip olmak ve projenin tüm paydaşlarını, her türlü rapor ve doküman üretiminde bu standart formata uygun hareket etmeleri konusunda zorlamak, tüm proje paydaşlarının tüm proje süreçleri ve faaliyetler hakkında güncel bilgilerle donatılmalarını sağlamak açısından önemlidir. Projelerin başarısı, önceki süreç ve projelerden yeterli tecrübeleri edinmek ve proje süreçleri arasında bilgi akışını ve kayıt mekanizmasını oluşturmak ile yakından ilgilidir.

Aşağıda, Geliştiricinin genel yaklaşımını göstermek üzere sunulan, proje kapsamında projenin gereği ve Müşterinin talepleri doğrultusunda güncellenecek olan, bilişim projelerinde genel olarak üretilen ve bu proje kapsamında da

hazırlanacak olan dokümantasyona yönelik özet bilgiler sunulmaktadır. bilecektir. Proje kapsamında üretilecek tüm dokümanlar kullanıcıya hem elektronik ortamda hem de basılı olarak çoğaltılarak teslim edilecektir.

**Tablo-E13.4:** Proje Dokümanları ve Çalışma Kapsamında Hazırlanma Durumu

| Doküman  | Projelendirme Kapsamında Hazırlanma Durumu  |
|--|---|
| Proje Yönetimi ve Kontrol Dokümantasyonu               | Projenin uygulanacağı firmaya ilişkin özellikler içerdiğinden bu çalışma kapsamında hazırlanmamıştır.   |
| İş Akışları Tanımlama Dokümanı                         | Projenin uygulanacağı firmanın mevcut iş akışlarını içereceğinden bu çalışma kapsamında hazırlanmamıştır.   |
| Sistem ve Kullanıcı İhtiyaçları Özellik Dokümanı (SRS) | EK-6  |
| Kullanıcı Ara Yüzü Tasarım Dokümanı                    | Bu çalışma kapsamında hazırlanmamıştır.   |
| Yazılım Geliştirme Planı                               | EK-12   |
| Yazılım Uygulama Planı                                 | EK-13   |
| Yazılım, Donanım, İletişim Altyapısı Mimarisi          | Bu doküman üç ayrı EK şeklinde hazırlanmıştır. [Sistem Yazılım, Donanım, İletişim Altyapısı Mimarisi (EK-7), Sezgisel Model (EK-8) ve Tekrarlı Taşıma Modelleri (EK-9)] |
| Veri Değişimi  | EK-11.  |
| Veri Tabanı ve Veri Modelleri                          | Kavramsal / Fiziksel İlişkisel Veri Modeli (EK-10)  |
| Test Planı ve Test Senaryo Kitapçığı                   | Bu çalışma kapsamında hazırlanmamıştır.   |
| Proje Tamamlanma Raporu                                | Bu çalışma kapsamında hazırlanmamıştır.   |
| Teknik Dokümantasyon                                   | Bu çalışma kapsamında hazırlanmamıştır.   |
| Uygulama Yazılım Dokümantasyonu ve Kullanıcı El Kitabı | Bu çalışma kapsamında hazırlanmamıştır.   |
| Standart Yazılım Dokümantasyonu                        | Bu çalışma kapsamında hazırlanmamıştır.   |
| Sistem Bakım Kitabı                                    | Bu çalışma kapsamında hazırlanmamıştır.   |
| Sistem Kurtarma Planı                                  | Bu çalışma kapsamında hazırlanmamıştır.   |
| Uygulama Yazılımları Geliştirici Kılavuz               | Bu çalışma kapsamında hazırlanmamıştır.   |
| Eğitim Planı   | (EK-14)   |



EK-14: **Eđitim Uygulama Planı**

# Eđitim Uygulama Planı

**Doküman**

**Versiyon No: .....**

Tarih



## 1. GİRİŞ

Bu doküman proje kapsamında gerçekleştirilecek eğitimlere ilişkin planı detaylandırmaktadır. Proje kapsamındaki eğitimler toplam 19 konu başlığı altında verilecektir.

Sistem eğitimleri alanlarında deneyimli ve uzman eğitmenler tarafından verilecektir. Her kurs sonunda katılımcılar ve eğitmenler tarafından kurs değerlendirme formları doldurulacak ve bu formlar değerlendirilerek kursun başarısına karar verilecektir. Her kurs için kursiyerlere bir eğitim seti, ilgili kursun başlangıcında dağıtılacaktır.

Bir sınıftaki kursiyer sayısı en çok 20 (yirmi) olacak, her kursiyere bir bilgisayar ve gerekli eğitim malzemesi sağlanacak ve günlük eğitim süresi 6 (altı) saat ile sınırlanacaktır.

Eğitim programlarında başarılı olanlara kursların içerik ve süresini belli eden bir katılım ve başarı belgesi verilecektir. Kurs notu, sarf malzemesi gibi her türlü eğitim materyali firmamız tarafından karşılanacaktır. Her kurs için kursiyerlere ayrı bir eğitim seti oluşturacak ve kursun başlangıcında kursiyerlere dağıtılacaktır.

Kullanıcı tarafından başarısız bulunan eğitimler tekrar edilecektir.

## 2. SİSTEM EĞİTİM PLANI

Sistem kapsamındaki eğitimler, katılımcı sayıları, planlanan tarih, süreleri ve açıklamaları aşağıdadır:

**Tablo-14.1 Eğitim Planı**

| S. Nu. | Eğitim Konuları                 | Katılımcı Sayısı | Planlanan Tarih | Eğitim Mekanı | Eğitim Süresi | Açıklama  |
|--------|---------------------------------|------------------|-----------------|---------------|---------------|---|
| 1      | Yazılım Geliştirme Metodolojisi | 2                | -               |               | 15 saat       | Mevcut temel metodolojiler hakkında kavramsal düzeyde genel bilgi, UKS Projesine uygulanacak metodoloji hakkında uygulamalı eğitimidir. |
| 2      | UML Modelleme Dili              | 2                |                 |               | 2 gün         | UML Modelleme dili eğitimi  |
| 3      | Ağ Donanımı                     | 2                |                 |               | 1 gün         | Kurumda mevcut ve UKS Projesi kapsamında yeni eklenecek aktif cihazlar hakkında genel bilgi.  |

| S. Nu. | Eğitim Konuları  | Katılımcı Sayısı | Planlanan Tarih | Eğitim Mekanı | Eğitim Süresi | Açıklama  |
|--------|--|------------------|-----------------|---------------|---------------|---|
| 4      | Ağ Konfigürasyonu  | 2                |                 |               | 1 gün         | Kurumda mevcut ve UKS Projesi kapsamında yeni eklenecek aktif cihazlar için konfigürasyon tasarımı uygulamalı eğitim  |
| 5      | Ağ Yönetimi  | 2                |                 |               | 1 gün         | Ağ yönetimi konusunda uygulamalı eğitim   |
| 6      | Karar Destek Sistemleri  | 2                |                 |               | 15 saat       | Yönetim Süreçlerinde Karar Destek Sistemleri eğitimi  |
| 7      | Çözüm Geliştirme Eğitimi   | 2                |                 |               | 15 saat       | Çözüm geliştirme disiplini  |
| 8      | Veri Tabanı Yönetim Sistemi  | 2                |                 |               | 10 gün        | Projede kullanılacak VTYS'nin programlama, tasarım, yönetimi, optimizasyon ve raporlama aracı eğitimlerini içerecektir.   |
| 9      | Sistem VTYS Uygulamaları   | 2                |                 |               | 1 gün         | Sistem bünyesinde kullanılan veritabanı kavramları, uygulama örnekleri, raporlama örnekleri, Sistem veritabanı işletimi vb. konularda Yüklenici firma tarafından kullanıcı teknik personele verilecek görev başı eğitimleri içermektedir. |
| 10     | Yazılım Geliştirme Araçları  | 2                |                 |               | 2 gün         | Sistemin geliştirilmesinde kullanılacak tüm ticari yazılımlara ait eğitimleri temel ve ileri seviyede içerecektir.  |
| 11     | Unix İşletim Sistemi   | 2                |                 |               | 2 gün         | Projede kullanılacak Unix işletim sistemine ait temel, ileri ve yönetici seviyesi eğitimleri içerecektir.   |
| 12     | Windows İşletim Sistemi  | 2                |                 |               | 5 gün         | Yazılımın kullanacağı Windows işletim sistemine ait temel, ileri ve yönetici seviyesi eğitimleri içerecektir.   |
| 13     | Sistem Kullanma Kursu<br>TPM<br>KURPM<br>VTYM<br>ETARİTM<br>Elektronik Mesaj Formu | 80               |                 |               | 5 gün         | Sistem modüllerinin kullanım eğitimi (Dört grup halinde eğitim verilecektir.)   |
| 14     | Sistem Kurulumu ve İşletim Kursu   | 2                |                 |               | 3 gün         | Sistem Kurulumu ve İşletim Kursu eğitimi  |
| 15     | Ulaştırma Sistemlerine Giriş   | 2                |                 |               | 15 saat       | Ulaştırma sistemlerine giriş eğitimi  |
| 16     | Ulaştırma Politikaları ve Kuralları  | 2                |                 |               | 15 saat       | Ulaştırma Politikaları ve Kuralları eğitimi   |

| S. Nu. | Eğitim Konuları                         | Katılımcı Sayısı | Planlanan Tarih | Eğitim Mekanı | Eğitim Süresi | Açıklama  |
|--------|---|------------------|-----------------|---------------|---------------|---|
| 17     | Intermodal Ulaştırma Kanun ve Kuralları | 2                |                 |               | 15 saat       | Intermodal ulaştırma kanun ve kuralları eğitimi |
| 18     | Ulaştırma Ekonomisi                     | 2                |                 |               | 15 saat       | Ulaştırma ekonomisi eğitimi                     |
| 19     | Intermodal Ulaştırma Sistemleri         | 2                |                 |               | 15 saat       | Intermodal ulaştırma sistemleri eğitimi         |

### 3. İÇERİK

#### 3.1. Yazılım Geliştirme Metodolojisi

Kurs İçeriği;

- **Modül 1: İş Çözümleri Tasarımına Giriş**

Bu modül MSF Süreç Modeli'ni ve uygulama modeli tasarımında yer alan anahtar aktiviteleri tanımlamaktadır. Aynı zamanda kurs boyunca kullanılacak vak'a çalışmasını açıklamaktadır.

Dersler;

- Microsoft Çözüm Çatısına Giriş
- MSF Süreç Modelinin aşamaları
- Vak'a çalışması-Uygulama

Bu modül tamamlandığında, katılımcılar;

- MSF Süreç Modeli ve MSF Takım Modelini tanımlayabilecek,
- MSF disiplinlerini tanımlayabileceklerdir: risk yönetimi, hazırlık yönetimi ve proje yönetimi.
- MSF Süreç Modeli aşamalarının anahtar aktivitelerini ve her aşamanın teslim edileceklerini tanımlayabilecek.
- Vak'a çalışmasını tanımlayabilecektir.

- **Modül 2: Bilginin Toplanması ve Analizi**

Bu modülde ihtiyaç duyulan bilginin toplanması, bilgi kaynakları ve bilgi toplama teknikleri işlenecektir.

Dersler;

- Modelleme İşaretlerinin Kullanımı
- Vaka ve Senaryoların Kullanımı
- Bilgi Toplama
- Bilginin Analizi
- Aktivite: Bilgi Toplama ve Analiz
- Görüşme Hazırlığı
- Örnek Satış Otomasyon Projesi ve Web İyileştirme Projesi için use-case ifadelerinin hazırlanması
- Başlangıç bilgi toplama faaliyeti sonrası taslak ihtiyaç tanımının hazırlanması
- Kullanım senaryosu geliştirme

Bu modülün tamamlanmasıyla kursiyerler aşağıdakileri yapabilecektir;

- UML ve ORM gibi modelleme notasyonlarının tanımlanması
- Bilgi toplama tekniklerini tanımlama
- Bilgi toplanacak kaynakların
- Bilgi toplama stratejisinin oluşturulması
- Use-case ve kullanım senaryoları kullanılarak ihtiyaçların analizi
- Dâhili proje dokümanlarının hazırlanması

- **Modül 3: Çözümün Planlanması**

Bu modül, bu aşamada görev alan takım üyelerinin sorumluluklarını tanımlamaktadır. Projenin vizyonunun tanımlanması ve projenin risklerinin analiz edilmesi işlenecektir.

Dersler;

- Planlama Aşaması
- Vizyon Kapsam Dokümanının Hazırlanması
- Proje Yapısı Dokümanının Hazırlanması
- Riskerin Analizi

Faaliyet: Vizyon Kapsam Dokümanının Hazırlanması;

- Problemin İfade Edilmesi
- Vizyon İfadesinin Hazırlanması
- Proje Hedeflerinin Tanımlanması

Bu modülün tamamlanmasıyla kursiyerlere uygulatılacak hususlar;

- MSF Süreç Modeli Planlama aşamasının amacı, aktiviteleri ve teslim edilecekleri dikkate alınarak tanımlanması,
- Vizyon Kapsam dokümanının bileşenlerini tanımlanması,
- Proje Yapısı dokümanının bileşenlerini tanımlanması,
- Projenin risklerini analizi..

#### • **Modül 4: Kavramsal Tasarımın Hazırlanması**

Bu modül Kavramsal, Fiziksel ve Mantıksal tasarım aşamalarını açıklar. Ayrıca kavramsal tasarım işlemleri ve fonksiyonel özellikleri de tanımlar.

Dersler;

- Planlama Aşamasına Ait bir Takdim.
- Fonksiyonel Speklere Genel Bakış.
- Kavramsal Tasarım İşlemlerine Genel Bakış.
- Kavramsal Tasarımı İnşaa Etmek. (Building the Conceptual Design)

- Kavramsal Tasarımın Optimize Edilmesi

Faaliyet: Gereksinimlerin Analizi;

- Use Case'leri ve Gereksinimleri elemek.
- Kavramsal Model Diyagramına Gözetmek.

Bu modülün tamamlanmasıyla kursiyerler şu yetenekleri kazanacaktır:

- MSF süreç modelinin planlama aşamasının amacını tarifi,
- Planlama aşamasında işlevsel spektlerin rolünü tarifi,
- Kavramsal tasarımın amacını tarifi
- Kavramsal bir tasarımı çözümlemesi
- Kavramsal tasarım optimizasyonu
- Mantıksal tasarımın yararlarının tarifi
- Bir iş çözümü için bir mantıksal tasarım modelini oluşturulması
- Kavramsal tasarım dokumanlarını oluşturulması
- Kavramsal tasarımın optimizasyonu.

• **Modül 5: Mantıksal Tasarım Oluşturmak**

Bu modül iş çözümü için mantıksal katmanın nasıl tasarlandığını, araçların ve rapor tekniklerinin nasıl kullanıldığını tarif eder. Aynı zamanda mantıksal tasarımın nasıl optimize edildiğini tarif eder.

Dersler;

- Mantıksal Tasarıma Genel Bakış.
- Mantıksal Tasarım Oluşturmak.
- Mantıksal Tasarım Raporu.
- Mantıksal Tasarımı Optimize Etmek

Faaliyet: Mantıksal Tasarım için Nesnelere Belirlemek;

- Use Caseler’le Nesnelere Belirlemek
- Servis Matrisi Oluşturmak
- Zaman Diyagramı Oluşturmak

Bu modül tamamlandıktan sonra, kursiyerlere uygulanacak aşamalar;

- Mantıksal tasarımın yeteneklerini tarifi.
- Çözüm için bir mantıksal tasarım modeli oluşturulması

- Mantıksal tasarım çıktılarının belirlenmesi
- Mantıksal tasarımın optimizasyonu.

- **Modül 6: Fiziksel Tasarım Oluşturmak**

Bu modülde fiziksel tasarım ve aşamaları anlatılır.

Dersler;

- Fiziksel Tasarıma Genel Bakış.
- Fiziksel Tasarım Analizleri.
- Fiziksel Tasarım Rasyonalizasyon'u.
- Fiziksel Tasarımı Oluşturmak.

Faaliyet: Fiziksel Tasarım Çalışmaları;

- Sınıf modeli oluşturmak.
- Bileşen Model Diyagramı Oluşturmak.

Bu modül tamamlandıktan sonra, kursiyerlere uygulanacak aşamalar;

- MSF süreç modelinde fiziksel tasarımın rolünü tanımlama
- Fiziksel tasarımın aşamalarını tanımlama
- Bir hazırlık servisler topolojisini oluşturulması
- Çözüm için bir teknoloji seçimi
- Bileşenleri farklı katmanlara dağıtım ve paketlenmesi
- Bir programlama modelinin tasarımı.

- **Modül 7: Sunu Katmanını Tasarlamak**

Bu modül sunu katmanının iki aşamada nasıl tasarlandığını tanımlar.

Dersler;

- Kullanıcı Arayüzü Temelleri.
- Kullanıcı Arayüzü Tasarımı.
- Kullanıcı Kontrolleri Tasarımı.

Faaliyet: Kullanıcı Arayüzü Tasarlamak;

- Kullanıcı Arayüzü Prototipi Tasarlamak.

Bu modül tamamlandıktan sonra, kursiyerlere uygulatılacak aşamalar;

- Bir iş çözümü için sunu katmanının işlevinin açıklanması.
- Kullanıcı gereksinimlerini ve iş ihtiyaçlarını barındıran bir kullanıcı arabirimi tasarlanması.

- **Modül 8: Veri Katmanını Tasarlamak**

Bu modül veri katmanının nasıl tasarlandığını, en iyi şekilde veri katmanına nasıl ulaşıldığını ve datanın nasıl doğrulandığını açıklar.

Dersler;

- Veritabanını Tasarlamak.
- Veri Erişimini En İyi Şekilde Kullanmak.
- Verilerin Doğrulanması.

Faaliyet: Bir Veri Şemasını Oluşturmak.

- Bir Veri Şemasını Oluşturmak.

Bu modül tamamlandıktan sonra, kursiyerlere uygulatılacak aşamalar:

- Bir çözüm için veritabanı tasarımı
- Veritabanına erişimin sağlanması
- Veri doğrulaması oluşturulması

- **Module 9: Güvenlik Özelliklerini Tasarlamak**

Bu modül güvenliğin bir uygulama için nasıl tasarlandığını açıklar.

Dersler;

- Uygulama Geliştirmede Güvenliğe Genel Bakış.
- Uygulama Güvenliği için Planlama Yapmak.
- .NET Platformunun Güvenlik Özelliklerini Kullanmak.
- Yetkilendirme, Denetim Stratejileri ve Kimlik Doğrulamayı

Tasarlamak.

Faaliyet: Tehlike Modellenmesi ve Önleme



- Potansiyel Tehditlerin Tanımlanması
- Önleme Teknolojilerinin Uygulanması

Bu modül tamamlandıktan sonra, öğrencilere uygulandırılacak aşamalar:

- Bir uygulama için kimlik doğrulama sorgularının tanımı
- Bir uygulamanın security stratejilerinin planlanması
- .NET ile sağlanmış güvenlik özelliklerinin açıklanması
- Uygulama katmanları için yetkilendirme, denetim stratejileri ve kimlik doğrulama tasarımı

- **Modül 10: Planlama Aşamasını Tamamlamak**

Bu modül, planlama aşamasını tamamlamak için istenmiş görevler ve planları açıklar.

Dersler;

- Tasarımları Birleştirmede Dikkat Edilecekler.
- Yönetici Özelliklerini Planlamak.
- Gelecek Aşamaları Planlamak.
- Teknik Spektleri Oluşturmak.

Faaliyet: Test Planını ve Teknik Spektleri Gözden Geçirmek.

- Test Planını Gözden Geçirmek.
- Teknik Spektleri Gözden Geçirmek.

Bu modül tamamlandıktan sonra, öğrencilere uygulandırılacak aşamalar:

- Tasarımların birleştirilmesini dikkat et ve Planı oluştur.
- Projenin sonraki aşamaları için planları ve stratejileri oluşturunuz.
- Planı oluştur ve yönetsel özellikleri plana ilave et.
- Çözüm için teknik özellikleri ilave et.

- **Modül 11: Çözümün Yaygınlaştırılması ve Stabilize Edilmesi.**

Bu modül bir çözümün nasıl stabilize edildiğini ve nasıl yaygınlaştırıldığını açıklar.

Dersler;

- MSF'de Aşamanın Stabilize Edilmesi.
- Stabilizasyon Tesleri.
- MSF'de Yaygınlaştırma.
- Ürünün Ortama Yaygınlaştırılması.

Faaliyet: Hatalara Öncelik Atamak

- Gruplandırmak ve Öncelik Atamak.

Bu modül tamamlandıktan sonra, kursiyerlere uygulanacak aşamalar.

- MSF'in stabilizasyon aşama kilometre taşlarının açıklanması
- Testin içindeki görevlerin açıklanması.
- MSF'in yaygınlaştırma aşamasının kilometre taşlarının açıklanması
- Ürünün yaygınlaştırma aşamalarının tarifi
- Bir projeyi tamamlamak için gerekli faaliyetlerin tarifi.

### **3.2. Birleştirilmiş/Tekleştirilmiş Modelleme Dili (UML) Eğitimi**

#### **3.2.1. UML Genel Bakış**

- UML Özeti
- UML Tarihçesi
- UML'in hedefleri

#### **3.2.2. Modeller**

- Model nedir
- Model ne işe yarar
- Model'in seviyeleri
- Model'in içinde neler yer alır

### 3.2.3. UML Kavramları

- UML Bakış noktaları
  - UML Bakış noktaları
  - Statik bakış noktası
  - Vaka bakış noktası
  - Etkileşim bakış noktası
  - Aktivite bakış noktası
  - Fiziksel bakış noktası
  - Model Yönetim bakış noktası
  - Genişleme yapıları
  - Bakış noktaları arasında bağlantılar
- Statik bakış noktaları
  - Giriş
  - Sınıflandırıcı
  - İlişkiler
  - Bağlantılar
  - Genellemeler
  - Gerçekleştirme
  - Bağımlılıklar
  - Kısıtlar
  - Örnekler
- Vaka bakış noktaları
  - Giriş
  - Aktörler
  - Vakalar
- Faaliyet bakış noktaları

- Giriş
- Aktivite Diyagramı
- Aktiviteler ve diğer bakış noktaları
- Etkileşim bakış noktası
  - Genel Bakış
  - Sequence Diyagramı
  - Collaboration Diyagramı
- Fiziksel Görünümler
  - Genel Bakış
  - Component Diyagramı
  - Deployment Diyagramı
- Model Yönetimine Bakış
  - Paket
  - Paket Bağları
  - Bağlara Erişim ve Dışarı Aktarma
  - Model ve Alt Sistem
- Uzama Mekanizmaları
  - Kısıtlamalar
  - Etiketlenen Değerler
  - Stereo Tipler

#### **3.2.4. UML Model Araçlarının Takdimi**

- Dönüşümlü Yükselme
- Diğer Araçlar

#### **3.3. Ağ Donanımı**

- Ağ Kurulumları
- OSI/802 Modeli

- Ağ Tasarımı
- Ağ Kablolama Ortamları
- Kablosuz İletişim

#### **3.4. Ağ Konfigürasyonu**

- Ağ Veri İletimi
- Ağ Mimarisi
- Ağ Altyapı Bileşenleri
- Ağ İletişim Yapılandırması

#### **3.5. Ağ Yönetimi**

- Sunucu Kurulum Hazırlıkları
- TCP/IP
- Geniş Alan Ağ Bağlantısı
- Geniş Alan Ağ Cihazları
- Geniş Alan Ağ Protokolleri ve Standartları

#### **3.6. Yönetim Süreçlerinde Karar Destek Sistemleri**

- Bilgi, Yönetimsel Karar Verme ve Bilgi Kullanımı: Veri, Bilgi ve Karar Verme İlişkisi, Sistem Tanımı ve Problem Çözmede Sistem Yaklaşımı, Yönetim Seviyeleri

- Yönetim Bilgi Sistemleri ve Karar Destek Sistemleri: Bilgi Sistemleri, Tarihçe, Manuel ve Bilgisayara Dayalı Bilgi Sistemleri, Bilgi Sistem Tipleri, Yönetim Bilgi Sistemleri-Karar Destek Sistemleri Genel Tanımları ve İlişkisi

- Karar Destek Sistemleri: Karar Tipleri, Özellikleri, Bileşenleri, Karar Destek Sistemlerinin Fonksiyonları, Model Geliştirme Aşamaları, Grup Karar Destek Sistemleri (GKDS), GKDS Alternatifleri

- Veri Odaklı Karar Destek Sistemleri: Veri Ambarı (Data Warehouse), Veri odacıkları (Data Mart), Veri Madenciliği (Data Mining)

- Akıllı Karar Destek Sistemleri: Yapay Zeka ve Uzman Sistemler
- KDS Uygulamaları

### 3.7. Çözüm Geliştirme Eğitimi

- Gereksinimlerin Elde Edilmesi (Sosyo-teknik sistemler, dependability, süreçler)
- Analiz (çalışma grubu oluşturma ve yönetim, maliyet analizi)
- Tasarım (nesne yönelimli tasarım, gerçek zamanlı sistem tasarımı)
- Geliştirme (iteratif geliştirme, tekrar kullanılabilirlik, kritik sistem geliştirme, süreç iyileştirme)
- Test ve Geçerleme (doğrulama ve geçerlilik analizi, kritik sistem geçerleme)
- Sürekli Geliştirme (konfigürasyon yönetimi)

### 3.8. Veri Tabanı Yönetim Sistemi

- **Oracle Veri Tabanı İşletme Esasları I**
  - Oracle Veritabanı Mimari Bileşenleri
  - Oracle Veritabanı ile Başlarken
  - Universal Installer Özelliklerini Belirlemek.
  - Optimum Esnek Mimari'yi tanımlamak.
  - Yükleme Esnasında Veritabanını Oluşturmak.
  - Yeni bir Veritabanı Oluşturmak.
  - Yönetici Yetkili Kullanıcıları Belirlemek.
  - Oracle Veritabanını Yönetmek.
  - Bir Veritabanı Oluşturmak.
  - Veri Sözlüğünü Kullanmak.
  - Kontrol Dosyalarını Takip Etmek.
  - Geçmişe Yönelik Kayıt Dosyalarını Takip Etmek.
  - TableSpace ve Data Dosyalarını Yönetmek.
  - Depolama Ortam Hiyerarşisini Tarif Etmek.

Kurmak.

- SISTEM'in ve Non-SYSTEM'in Tablespacesi Arasında Farklılığı

Anlamak.

- Depolama Yapıları ve İlişkileri.
- Veri Geri Almak.
- Otomatik Geri Almayı ve Manuel Geri Alma Arasındaki Farkı

- Geri Alma Tablespaces'lerini oluşturmak.
  - Tablo'ları Yönetmek.
  - İndeks'leri Yönetmek.
  - Kullanılmamış Dizinleri Teşhis Etmek.
- **Oracle Veri Tabanı İşletme Esasları II**
    - Ağ İletişimine Genel Bakış.
    - Temel Oracle Net Mimarisi.
    - Temel Oracle Net Servislerinin Sunucu Tanımlamaları.
    - Temel Oracle Net Servislerinin İstemci Tanımlamaları.
    - Oracle Shared Sunucusunun Kullanımı ve Yapılandırılması.
    - Yedekleme ve Geri Almaya Genel Bakış.
    - Ortam Kurtarma Yapıları.
    - Veritabanı Arşivleme Modunun Yapılandırılması.
    - Oracle Kurtarma Yöneticisine Genel Bakış ve Konfigurasyonu.
    - Kullanıcı bazlı Yedekleme.
    - RMAN Yedekleme.
    - Kullanıcı Bazlı Tam Kurtarma.
    - RMAN Tam Kurtarma.
    - Kullanıcı Bazlı Tamamlanmamış Kurtarma.
    - RMAN Tamamlanmamış Kurtarma.
    - RMAN Havuz Bakımı.

- RMAN Katalog Oluşum ve Kullanımı.
- Veritabanları Arasında Veri Aktarımı.
- Yedekleme Kurtarma ve Ağ Çalışmaları.

### 3.9. Sistem VTYM Uygulamaları

- Veritabanına Giriş
- VTYM Kurulumu
- Sistemde Kullanılan Veritabanı Kavramları
- Veritabanı Yönetimi
- Veritabanı Uygulama Örnekleri
- Veritabanında Raporlama
- Veritabanı İşletimi
- Veritabanı Sorunlarını Giderme

### 3.10. Yazılım Geliştirme Araçları

Microsoft .NET ile C# programlamasına giriş kapsamında;

- **Modül 1: Başlarken**

Bu modül Microsoft .NET Framework ve platformlarının merkezinde yer alan konseptleri sunmaktadır. Microsoft Visual Studio .NET bütünleşik geliştirme ortamı (IDE); geliştirme ortamında nasıl çalışılacağını ve and explains how to use Microsoft .NET kapsamında uygulamaların oluşturulmasında Visual Studio .NET'in nasıl uygulanacağını açıklamaktadır.

- **Modül 2: C# Dil Temel Kullanım Esaslarının Anlaşılması**

Bu modül ile C# programlama dili yazımsal ve yapısal özelliklerini öğrenilmesi sağlanacaktır.

- **Modül 3: C#'ta Nesne Yaratımı**

Bu modülle nesne, sınıflar ve metotları da içerecek şekilde nesne tabanlı programlama temelleri öğretilmektedir.

- **Modül 4: C#'ta Nesne Tabanlı Programlama Tekniklerinin Kullanımı**



Bu modül, nesne, sınıflar ve metotları da içerecek şekilde nesne tabanlı programlama temelleri öğretilmektedir.

- **Modül 5: C# ile Programlama**

Bu modül ile ışımlar, koleksiyonlar ve bunların uygulamalarda kullanımını da içeren çeşitli veri yapıları öğretilmektedir. Modül aynı zamanda özel durum elleçlemesi yazım ve konseptlerini tanımlayan ara yüzleri ve olay elleçlemede delege yapısı kullanımını açıklamaktadır.

- **Modül 6: C# ile .NET Tabanlı Uygulama Hazırlama**

Bu modül Microsoft .NET Framework sınıf kütüphanesini, nesne göstericiyi ve sistem nesne sınıfından elde edilen metotları öğretecektir. İplerin ve numaraların şekillendirilmesi akış ve dosyaların kullanımı da bu modül kapsamında açıklanmaktadır.

- **Modül 7: Veri Erişimi İçin ADO.NET Kullanımı**

Bu modül ile, Microsoft ADO.NET ve sistem nesnelерinin kullanımı öğretilmektedir. ADO.NET ile Microsoft Windows tabanlı uygulama oluşturulması sağlanacaktır. Bu modül, veri tabanına bağlanma, arama yapma, veri yönetiminde veri seti nesnelерinin kullanımı, veri tabanına veri girme, ilişkilendirme, güncelleme ve kayıt silme işlemlerinin nasıl yapılacağını da öğretecektir.

- **Modül 8: Windows Tabanlı Uygulama Oluşturulması**

Bu modül, menülerin, genel ve seçimli iletişim kutucuklarının, durum çubuklarının ve Microsoft Windows tabanlı uygulamaların kullanışlılığını artırıcı araç çubuklarının oluşturulması öğretilmektedir. Bu modül ile, kursiyerlerin yeni öğrendikleri C# özelliklerini uygulayabilmeleri ve faydalı Windows tabanlı uygulama geliştirmelerinin sağlanması amaçlanmaktadır.

- **Modül 9: XML Web Hizmetlerinin C# Programında Kullanımı**

Bu modül, "System.Web.Services" isim alanı ve C# uygulamalarında XML web hizmetlerinin kurulması ve kullanımı sürecini öğretecektir.

- **Modül 10: Web Formları Kullanılarak Bir Web Uygulamasının Oluşturulması**

Bu modülde, kursiyerlere Microsoft Visual Studio .NET kullanılarak programlanabilir web sayfası oluşturmada “Web Forms” kullanımı öğretilenektir. Kursiyerler, Web Form kontroller eklenmesini ve müteakiben de veri aktarımı olaylara cevap verilmesini öğrenecektir. Aynı zamanda Microsoft ASP.NET yönetimi, güvenlik ve konfigürasyon ayarlarının yapılması da bu kapsamda öğretilenektir.

- **Modül 11: Uygulama Ayarları ve Geliştirilmesi**

Bu modül ile, Microsoft Visual Studio .NET kullanılarak bir C# uygulamasının (ister web tabanlı ister Windows tabanlı olsun) aktarımına ilişkin süreçler öğretilenektir. Kullanıcı tercihleri ve uygulama yarılarının konfigürasyonu da bu kapsamda öğretilenektir.

- **Modül 12: Gelişmiş C# Özelliklerinin Kullanımı**

Bu modül ile kursiyerlerin C# programı bazı gelişmiş özelliklerinin kullanımı konusunda bir fırsat sunmaktadır. Kursiyerlerin ileride kendi kendilerine öğrenebilecekleri özellikler için ortam hazırlamaktadır.

### 3.11. Unix İşletim Sistemi

- Unix İşletim Sistemine giriş
  - UNIX'in özellikleri
  - UNIX'in tarihçesi
  - UNIX destekleyen donanım ve üreticiler
- UNIX'in Temelleri
  - Donanım
  - Yazılım (Kernel, Shell)
- UNIX'e ulaşım
- Dosya ve Dizinlerin Anlaşılması
  - UNIX Dosya Sistemine Giriş
  - UNIX Komutlarının elemanları
  - Temel UNIX Dosya ismi komutları

- Dizinleri anlamak
- Özel dosya ve dosya tiplerini anlamak
- Dosya bağlantıları
- UNIX'te çoklu dosya sistemleri
- Güvenlik
  - ◆ Dosya düzeyinde güvenlik
  - ◆ Sistem seviyesinde güvenlik
  - ◆ Sahip, grup ve izinlerin değiştirilmesi
- UNIX Shell
  - Login süreci
  - Shell'de komut işleme
  - Multitask mantığı
  - Diğer shell komutları
  - Değişkenler
  - Çevre
  - Shell programlama
    - ◆ CCH/TCSH Shell
    - ◆ Korn Shell
    - ◆ Bourne Shell
  - X Windows Sistemleri
- UNIX Araçları
  - UNIX Batch facilities (at, cron commands)
  - UNIX Print Spooling Systems
  - Command Line Processing
  - Diğer UNIX Araçları
- UNIX'te metin işleme

- Vi Metin Editörü
- Diğer Görsel Editörler
- Sed & awk
- İletişim ve Ağ
  - TCP/IP'ye giriş
  - NIC ayarları
  - Yönlendirme ayarları
  - DNS, NIS ve NFS
  - Ağ iletişimde temel sorunların giderilmesi
- İç yapının anlaşılması
  - Proses
  - UNIX Dosya Sistemi
  - UNIX Kernel
  - Değiş-tokuş ve sayfalama
- UNIX Sistem Yönetimi
  - UNIX Sisteminin kurulumu
  - Açma ve kapama sırası
  - Sistem yönetimi dosya ve komutları
  - Kullanıcılar için sistemin idamesi
  - Sistem yedekleme ve geri yükleme
  - Sistem performansının izlenmesi
  - Sistem Logları
  - UNIX Sistemlerinde temel problemlerin giderilmesi
- UNIX Dokümantasyonu
  - Man sayfaları
  - Çevrimiçi dokümanlar

- UNIX için yararlı kaynaklar
- DOS'tan UNIX/Linux'e

### 3.12. WINDOWS İşletim Sistemi

- Microsoft Windows XP'nin Kurulumu
  - Microsoft Windows XP Profesyonel kurulumu ve planlanması
  - Windows XP Profesyonel'in ürün CD'sinden kurulumu
  - Windows XP Profesyonel'in Ağ üzerinden kurulumu
  - Microsoft Windows XP Profesyonel'in yükseltilmesi
  - USMT ile kullanıcı ayarlarının transferi
  - Windows XP Profesyonelin aktive edilmesi
  - Problemler kurulumlar
- Microsoft Windows XP kurulumunun otomatikleştirilmesi
  - Giriş
  - Cevap dosyalarının oluşturulması
  - Tekil Veritabanı Dosyası oluşturma
  - Cevap dosyaları ve UDF'lerin kurulumu
  - Windows XP Profesyonel otomatik kurulumunu gerçekleştirme
  - İmaj dosyası yaratma ve dağıtma
  - Referans bilgisayarın hazırlanması
  - İmajın yaratılması, testi ve dağıtımı
  - Uzaktan kurulum hizmetleri
- Windows XP ile çalışan bir bilgisayarda donanım
  - Donanım kurulumu ve ayarlanması Devices
  - Sürücülerle çalışma
  - Donanımlarda problem giderme
- Disklerin Yönetimi

- Disk Yönetimi ile çalışma
- Temel Disklerle çalışma
- Dinamik disklerle çalışma
- Windows XP Profesyonel yükseltilirken disklerin hazırlanması
- Disklerin yönetimi
- Disklerin bütünlenmesi
- Dosya Sistemlerinin ayarlanması ve yönetimi
  - Dosya Sistemleri ile çalışma
  - Veri sıkıştırmanın yönetimi
  - EFS kullanılarak verinin güvenlik altına alınması
- Bilgisayarın açılışındaki problemlerin giderilmesi
  - Examining the Windows XP Professional Boot Process
  - Açılışta sistem ayarlarının kontrolü
  - Changing Startup Behavior Using the Boot.ini File
  - Using Advanced Boot Options to Troubleshoot Startup Problems
  - Bilgisayarın kurtarma konsolu ile açılması
  - Bilgisayarı daha önceki bir durumuna geri döndürme
- Masaüstünün ayarlanması
  - Masaüstü ayarlarının değiştirilmesi
  - Masaüstünün kişiselleştirilmesi
  - Sistem ayarları
  - Kullanıcı profilleri ve grup politikalarının masaüstüne etkileri
  - Uzaktan yardımın kullanımı
- TCP/IP Adresleme ve isim çözümü
  - IP Adreslerinin ayarlanması
  - IP Adres problemlerinin giderilmesi

- TCP/IP İsim Çözümleme Metotları
- DNS ve WINS ayarları
- Uzaktaki bir ana bilgisayara bağlantı
- Microsoft Windows Ağlarının işletimi için Microsoft Windows XP'nin ayarlanması
  - Çalışma grupları ve kullanıcı hesapları
  - Yerel kullanıcı hesaplarının yaratılması ve doğrulanması
  - Yerel güvenliğin ayarlanması
  - Çalışma gruplarında logon ayarları
  - Çalışma grubunda ağ ayarlarının gerçekleştirilmesi
  - Alana dahil olma
  - Alanın işlenmesi
- Uzaktaki kullanıcıların desteklenmesi
  - Uzaktan bağlantıların kurulması
  - Özel ağlara bağlantı
  - Gelen bağlantıların ayarlanması
  - Doğrulama Protokollerinin Ayarları
  - Uzak Masaüstü Erişimi
  - Uzaktan bağlantılar için kullanıcı isimleri ve şifrelerinin kaydedilmesi
- Microsoft Windows XP mobil işlem ayarları
  - Mobil işlemler için donanım ayarları
  - Mobil bilişim için güç yönetim ayarlamaları
  - Çevrimdışı kullanım için dosya, izin ve web sayfaları hazırlama
- Kaynaklar ve performansın izlenmesi
  - Sistem bilgisinin anlaşılması
  - Görev Yöneticisi ile sistem performansının izlenmesi

▪ Performans ve izleme araçları kullanılarak performansın artırılması

- Olay Kayıt Kütükleri
- Program uyumluluklarını ayarlama

### 3.13. Sistem Kullanma Kursu

- **TPM**

- İzleme
  - ◆ Ulaştırma Altyapısı Analizi
  - ◆ Analiz ve Raporlama
  - ◆ İntikal Planı Simülasyon
- Ulaştırma Ağı
  - ◆ Ulaştırma Ağı Oluşturma
  - ◆ Alternatif Güzergâhlar Oluşturma
  - ◆ Ağın Kullanılabilirliği
  - ◆ İlave Kısıtlamalar
- Taşıma Planlama
  - ◆ Yeni Plan Hiyerarşisi Yaratma
  - ◆ Taşınacak Birim Bileşenlerine Ayırma
  - ◆ Yolcu/Malzeme Tahsisi
- Ulaştırma Aracı Seçme
  - ◆ Bekleme Noktası Seçme
  - ◆ Yükleme Planlama
  - ◆ Taşıma Planı Çatışma Giderme

- **KURPM**

- Sefer İşlemleri
  - ◆ Kurye/Ring Seferi İhdas Etme



- ◆ Birleřtirme, Kaldırma, İyileřtirme
- ◆ Kurye/Ring Planlaması
- ◆ Ulařtırma Ađı Oluřturma
- ◆ Güzergah Tespit Etme
- ◆ İlave Kısıtlama
- Talep Karřılama
  - ◆ Yolcu/Yük Planlama ve Yönetim
  - ◆ Yolcu/Yük Aktarma
  - ◆ Kurye/Ring Etkinlik Analizi
  - ◆ Planlama/Uygulama Farklılıkları
- **VTYM**
  - Ulařtırma
    - ◆ Ulařtırma Araç Bilgileri (Havayolu, Demiryolu, Karayolu, Denizyolu)
    - ◆ Ulařtırma Filosu Bilgileri
  - Kurye/Ring Seferi
    - ◆ Kurye/Ring Görüntüleme
    - ◆ Yükleme Öncelikleri Bilgileri
    - ◆ Kurye/Ring Seferi Planlaması
    - ◆ Taleplerin Onaylanması
    - ◆ Kurye/Ring Planlaması Yolcu/Malzeme Bilgileri
  - Tařınacak Birim Envanter İřlemleri
    - ◆ Raporlanabilir Envanter Kodu (REK) Görüntüleme
    - ◆ Envanter
    - ◆ Tařınacak Birim REK Bilgileri
  - Cođrafi Bilgi Sistemi

- ◆ Coğrafi Lokasyon Veri Girişi
  - Demiryolu Altyapısı
  - Liman Altyapısı
  - Havaalanı Altyapısı
  - Diğer
- ◆ Harita İşlemleri
- ◆ Harita Projesi İşlemleri
- Ulaşım Ağları
  - ◆ Ulaşım Ağları Verileri
  - ◆ Ulaşım Ağları Verileri ve Harita Eşleştirme
- Taşınacak Birim Yönetimi
  - ◆ Taşınacak Birim İşlemleri
    - Taşınacak Birim Özellikleri
    - Taşınacak Birim Profili
    - Taşınacak Birim Tipleri
  - ◆ Taşınacak Birim Yer Değiştirme İşlemleri
    - Taşınacak Birim Envanter
    - Ulaştırma Altyapısı
    - Plan
- **ETARİTM**
  - ETARİTM Modülünün Sistemle Senkronizasyonu
    - ◆ Sistem Senkronizasyonu
    - ◆ ETARİTM Sunumcusuna Bilgilerin Aktarılması
  - Araç Verileri İzleme
    - ◆ İz Bilgileri
    - ◆ Yolcu/Yük Bilgileri

- Kurye/Ring Seferi
  - ◆ Malzeme Bilgisi Sistem Senkronizasyonu
  - ◆ Kurye/Ring Planlama Mesajları Sistem Senkronizasyonu
  - ◆ Araç Bilgileri ile İlişkilendirme
- Araç Bilgilerinin Sisteme Aktarımı
- Malzeme Bilgisi Senkronizasyonu
- Elektronik Mesaj Formu
  - ◆ Elektronik Mesaj Formu ile ilgili gerekli bilgiler

### **3.14. Sistem Kurulumu ve İşletim Kursu**

- ETARİTM Sunucu (Server) Kurulumu
  - İşletim Sistemi Kurulumu
  - Gerekli Sunucu Programlarının Kurulumu
  - Sunucu için gerekli Yönetim Programları
- İstemci (Client) Kurulumu
  - İşletim Sistemi Kurulumu
  - Gerekli İstemci Programlarının Kurulumu
  - İstemci için Gerekli Yönetim Programları
- Yönlendirici (Router) ve Modemlerin Kurulumu
  - Router ve Modemlerin Konfigürasyonlarının Yapılması
  - Router'ın Modem ile Entegrasyonu
  - Router ve Modem'in Yönetimi
- GPS, Cep telefonu ve Notebook bilgisayar Kitinin Kurulumu
  - Notebook'a İşletim Sistemi Kurulumu
  - GPS ve Cep Telefonunun Notebook ile Entegrasyonu
- GPS Entegreli Cep Telefonu Kurulumu

- Cihazın Gerekli Konfigürasyonlarının Yapılması
- Cihazın Yönetimi
- Barkod Terminal ve Yazıcının Kurulumu
  - Barkod Terminal ve Yazıcının Konfigürasyonlarının Yapılması
  - Barkod Terminal, Yazıcı ve İletişim Cihazının Sisteme Entegrasyonu
- Diğer Donanım Kurulumları
  - Lazer Yazıcının Sisteme Entegrasyonu
  - CD Yazıcının Sisteme Entegrasyonu
- Sisteme Genel Bakış

### **3.15. Ulaştırma Sistemlerine Giriş**

- Genel Tanımlar ve Terminoloji
- Ulaştırma Sistemleri (Kara Ulaşımı, Demiryolu Ulaşımı, Havayolu ve Su Yolu Ulaşımı)
  - Ulaştırma Sistemlerinin Elemanları (Araç, İz, Terminal, Enerji Kaynağı, Kontrol Cihazları)
  - Ulaştırma Sistemlerinin Karşılaştırması (Sistemlerin birbirine göre avantaj ve dezavantajları)
  - Türkiye'deki Ulaştırma Sistemleri (Mevcut durum, problemler ve çözüm önerileri)
  - Karayolu Ulaşımı ve Problemleri (Türkiye ve Dünyada karayolu ulaşımı ve problemleri)

### **3.16. Ulaştırma Politikaları ve Kuralları**

- Genel Tanımlar ve Terminoloji
- Ulaşım ve Kentsel Ulaşım Genel İlkeleri
- Ulaşım Planlamasına Giriş
- Kentsel Ulaşım Politikaları
- Kentsel Ulaşım Stratejileri

- Avrupa'da Kentsel Ulaşım Uygulama Örnekleri

### **3.17. Intermodal Ulaştırma Kanun ve Kuralları**

- Tanımlar ve Terminoloji
- Intermodal Ulaştırma Sistemleri
- Ulaştırma Sistemleri Mevzuatı
- Ulaştırma Sistemlerinde Güvenlik
- Türkiye'de Intermodal Ulaştırma Sistemleri (Mevcut Durum, Öneriler)
- Intermodal Ulaştırma Sistemleri – AB Entegrasyonu

### **3.18. Ulaştırma Ekonomisi**

• Ulaştırma ve Ekonomi: Ulaştırma Sektörünün Ekonomik Yapıdaki Yeri ve Ulaştırma Talebi

• Ulaştırma ve Kalkınma: Ulaştırma ile Ekonomik Kalkınma arasındaki Bağlantıların Analizi

- Ulaştırma Hizmetleri, Üretim, Yerleşim ve Dış Ticaret
- Ulaştırma ve Lojistik: Dünyanın Önemli Lojistik Merkezleri ve Özellikleri
- Avrupa Birliği Ortak Ulaştırma Politikası
- Avrupa Birliği ve Avrasya Ulaştırma Koridorlarında Rekabet
- Ulaştırma 2023 Vizyonu ve Türkiye'de Ulaştırma Sektörünün Geleceği

### **3.19. Intermodal Ulaştırma Sistemleri**

- Tanımlar ve terminoloji
- Intermodal Taşımacılık ve Kombine Taşımacılık
- Türkiye'de Intermodal Taşımacılığın Gelişimi
- Intermodal – Kombine Taşımacılık Problemleri ve Çözümleri.