DEVELOPING AN AUTOMATED SOFTWARE DEFECT MANAGEMENT SYSTEM AND EFFECTS OF SYSTEM ON SOFTWARE DEVELOPMENT PROCESS

A Thesis Submitted to the
Dokuz Eylül University, Graduate School of Natural and Applied Sciences
In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering, Computer Engineering Program

by
Mustafa ERŞAHİN

February, 2009
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **“DEVELOPING AN AUTOMATED SOFTWARE DEFECT MANAGEMENT SYSTEM AND EFFECTS OF SYSTEM ON SOFTWARE DEVELOPMENT PROCESS”** completed by **MUSTAFA ERŞAHİN** under supervision of **DR. KÖKTEN ULAŞ BİRANT** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Supervisor

Jury Member	Jury Member

Prof.Dr. Cahit HELVACI

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

The author wishes to express his deepest gratitude to his supervisor Dr. Kökten Ulaş Birant for his guidance, advice, criticism, encouragements and insight throughout the research.

Mustafa ERŞAHİN

DEVELOPING AN AUTOMATED SOFTWARE DEFECT MANAGEMENT SYSTEM AND EFFECTS OF SYSTEM ON SOFTWARE DEVELOPMENT PROCESS

ABSTRACT

There has always been major interest in the area of software development is "quality". Today, all kinds of software companies and some departments of universities develop new software's. This products may be very successful and meet all of customer's satisfaction. Across to this, there is impossible to develop software without error practically. Therefore, number of errors and defects can help us to identify software quality.

While defects affect software quality, they must be controlled and managed. Otherwise, we cannot find the exact problems that cause a defect or an error. If we analyze the real problems on our software, we can develop a project more reliable with fewer errors and we can improve our project's quality to better quality standards.

In this approach, a new defect management system that called UHISYS was developed for software development team of a company. The main purpose of this defect management system is controlling the errors and defects on development period. After enough data that has information about the real problem, collected, an analyzing period that includes all parts of software production lifecycle starts. New solutions find on software or new software methodologies practice by development team according to these analyzes. This thesis will show us how a defect management system affects the overall product performance and product quality of a software development company.

Keywords: Defect, fault, management, quality, software

HATA YÖNETİM SİSTEMİ GELİŞTİRİMİ VE BU SİSTEMİN YAZILIM GELİŞTİRME SÜRECİNE ETKİLERİ

ÖZ

Yazılım geliştirme sektöründeki en önemli unsurlardan biri yazılımın kalitesidir. Günümüzde farklı amaçlar ve uygulamalar üzerinde kullanılmak için üniversiteler ve özel yazılım firmaları tarafından yazılım geliştirilmektedir. Yazılımların gerçek amacı müşterilerin yazılımdan beklentilerini karşılayabilmektir. Ancak bunu her zaman başarabilmek mümkün değildir. Çünkü bir yazılımı hatasız bir şekilde üretebilmek mümkün değildir. Bu nedenledir ki, var olan bir yazılımın kalitesini ölçmek için yazılımda ortaya çıkan hata sayısını kullanabiliriz.

Hatalar, yazılımın kalitesine etki ettiğine göre yazılım geliştirme süresince iyi takip edilmeli ve yönetilmelidir. Bu sayede geliştirilen yazılımda hata oluşmasına neden olan gerçek sebepler bulunabilir ve bu hataların oluşmasını engelleyecek sistemler üzerinde çalışma yapılabilir. Dolayısıyla çok daha tutarlı bir yazılım geliştirilerek kalite konusunda ilerlemeler elde edilebilir.

Bu yaklaşımla, UHISYS adında yeni bir hata yönetim sistemi geliştirildi. Bu hata yönetim sisteminin öncelikli amacı, uygulama üzerinde meydana gelen hataların kontrollü bir biçimde yönetilmesini sağlamaktır. Sistem üzerinde yeterli veri toplandıktan sonra, bu verilerin analizinden yola çıkarak yazılım geliştirme süreçlerindeki aksayan noktaları tespit ederek, bu noktalara yeni çözümler üretilmesini sağlamak UHISYS hata yönetim sisteminin diğer amaçları arasındadır. Bu araştırma, bir hata yönetim sistemini kullanan yazılım şirketinde, kullanılan bu sistemin, geliştirilen yazılımın kalitesine ve performansına olan etkilerini görmemizi sağlayacak.

Anahtar sözcükler: Hata, yönetim, kalite, yazılım

CONTENTS **Page**

CHAPTER ONE

INTRODUCTION

1.1 Introduction

"Quality is related with the characteristics of a product or service that bear on its ability to satisfy stated or implied needs." There are some different quality definitions such as "Quality means fitness for use" by Joseph Juran and "Quality means conformance to requirements" for Philip Crosby. But generally quality is related with product characteristics. We all know how important "quality" is during our life for different areas.

The most important area of life's is information technology. In fact, the overall technology is getting better and faster day by day. We use computers, mobile phones and computers, some mechanical products in our jobs to do daily operations. Most of them use some operation specific software. Therefore, we can say "Software's are everywhere in our life." According to this fact, the quality of the software affects our life directly.

There are some different things that identify software quality. One of them is called "error" or "defect". If software cannot meet user's requirements because of errors and cannot fix as soon as possible, the software's quality is under quality standards. Furthermore, fixing software errors is as important as preventing errors. At this point, a new kind of software that called "Defect Management System" is getting necessary.

Defect management systems are developed for software production teams to manage errors or defects in software development lifecycle. We will see advantages or disadvantages of these systems on development process of a software company and how it is affect related software quality.

CHAPTER TWO

QUALITY AND QUALITY MANAGEMENT

Every part of life, human beings want to have better things that all they need. Actually, the whole people have an interesting desire that is reaching perfect one. History and evolution of quality will be explained in next section.

2.1 History of Quality

There is no common definition of quality for everyone. Quality has multi dimensional architecture according to related area. According to this fact quality has many meanings. These are; “Ranging from luxury and merit to excellence, good value for money or convenience and even practicality. It is often defined simply as 'fitness for purpose'. Quality is a multi-faceted concept; different dimensions of quality will be important to different users.”

People started to interest quality at archaic times with their conditions and properties. But, modern quality approach is seen in 20th century with United States of America. The first quality control charts can be seen in 1920’s. After 1930’s, some quality control standards were prepared according to the statistical techniques to identify quality of the product. Then, United States of America introduces the “War Standards” to world as a quality standards guide in 1941. After the quality guide produced, new quality standards were started to use by industry.

Japanese people were interested in quality after the 2nd World War. They understood the importance of the quality control and got some conferences from quality professionals. We meet a new concept that quality of production process in those conferences.

After that, total quality control techniques that include both quality of product and production process are developed. Total quality control techniques were used widely by companies. An idea that “no errors on production” was rose in 1960’s. Japanese were successful while integrating the quality control circles to management system. Therefore, the idea of quality management was born.

Quality management techniques were used by lots of companies and after a practice period it served two alternative production strategies to managers. These strategies are “Low Cost” and “High Quality”. Development period of the quality management were continued in 1970’s till de beginning of the 1980’s. These years were called rivalry years because of companies’ management strategies. In addition to this, standardization was getting important for every industrial sector. At last, we introduced to International Organization for Standardization (ISO).

A lot of countries in the world are member of this organization. Series of standards were identified by ISO. The first series of standards called ISO-9000. These standards guarantee quality assurance for production sectors. New product or production standards are adding these series of standards or some standards are updated if it necessary according to the technological improvements.

In short, United States of America and Japan are very important on evolution of quality and quality based techniques. There are lots of quality standards and quality management techniques for each sector due to their successful researches and practices.

2.2 Quality Measurement

As everyone knows, there are quality standards defined by national and international organizations. Production processes and products are classified according to the related standards. Therefore, we can say quality is a measurable quantity. If it is measurable quantity, there are some questions to ask such as “What are dimensions of quality?” and “What is quality management?”

2.2.1 What are dimensions of Quality?

Quality is measurable quantity on some dimensions. There are six different dimensions quality measurement lies on. These dimensions are;

- Relevance
- Accuracy
- Timeliness
- Accessibility
- Comparability
- Coherence

All of these dimensions are used to identify product’s quality. Product supplies all of customer’s requirements and needs to be relevant. If it does not satisfy customer’s expectation, the product’s quality is bad on relevance dimensions.

Products must supply right outputs according to the inputs. If they cannot supply any output or wrong output, accuracy dimension of that product quality is bad.

Products must be delivered to customers on time that scheduled before. In generally, the timeliness dimension of quality is related with producers’ process quality.

If a customer cannot reach the product when necessary conditions provided, the quality on accessibility dimension is bad.

The product can be comparable with other products. If it cannot compare to others, it may not demand on customers.

At last, a product must be consistent in every condition. If products' output changes with same inputs, system cannot be consistent. Therefore, coherence dimension of the product quality is evaluated in bad way.

2.2.2 What is quality management?

"Quality management is a method for ensuring that all the activities necessary to design, develop and implement a product or service are effective and efficient with respect to the system and its performance."

Quality management concept can be defined as total product or service quality. It includes quality control mechanisms. Therefore the major purpose of the quality management is to offer the goods and services that compatible with customer requirements. The main objectives of the quality management can be summarized with three items.

- Offer the products and services that will satisfy the requirements of the customer

- Provide higher profitability for the company with measures that improve working procedures, less errors, lower costs, less debts and better orders

- Helping the employees fully use their potentials to reach the company goals; to encourage the spread of the company policy, and volunteer activities.

These objectives are very important. If your product or service satisfies all of the customers' requirements, you and your company can gain confidence of the customers.

It is obvious that to do these objectives, company must have an effective leader that possesses the necessary capabilities. This leader is responsible to control their products on all of quality dimensions.

As a result, quality management is method of supplying products or services that satisfies customers’ requirements. The main purpose of the quality management is achieving customer satisfaction.

CHAPTER THREE

SOFTWARE QUALITY

3.1 What is Software Quality

Importance of the software quality is getting higher. Lots of applications are used in the world on different systems and different process. Some of them are more important and critical than others such as airplane systems or traffic control applications. A defect can cause death or injuries for people. Therefore, the software quality is a big problem according to the related applications. Across to this, the software community has been slow to use data to measure software quality. “There are four reasons why the software community has been slow to use numbers for software quality.”

- There is no generally recognized definition for quality measures.
- Even when we know what numbers to use, the data are not easy to gather.
- Even with data, it is not obvious how to interpret and use the numbers.
- People are reluctant to measure the quality of their personal work.

Actually, there are not any generalized definitions to measure software quality. There are many different cases on different software projects. Therefore, the quality of the software projects must be defined by last user. However, their quality concept may be different from other software’s quality concepts, there are some basic measures to that affect software quality such as size, time and defects. For example; unpredictable costs for software according to simple mistakes, after software being used can be evaluated as a quality metric.

There is another approach to software quality by dividing into two different parts. These parts are internal software quality and external software quality. External software quality can be seen as general software quality for last users view. Internal software quality is related with development tools, coding standards and all processes in development lifecycle. We will learn about these measures and their effects to software quality.

3.2 What affects software quality?

There is no common software quality definition to measure software quality. Cost, schedule and functionality are the most important metrics and these are effects that software product quality for customer perspective. Actually, these three metrics affects the software quality in different ways.

Some developers do not care about their quality of codes to do deliver on time. This approach can cause defects after the product delivered last user. In addition to this, it is more difficult and more cost to fix defect after product delivered. Across to this, there are proven ways to solve these problems. The first step is adopting and demanding that vendors follow these six principles of software quality:

- If a customer does not demand a quality product, he or she will probably not get one.
- To consistently produce quality products, the developers must manage the quality of their work.
- To manage product quality, the developers must measure quality.
- The quality of a product is determined by the quality of the process used to develop it.
- Since a test removes only a fraction of a product’s defects, to get a quality product out of test you must put a quality product into test.
- Quality products are only produced by motivated professionals who take pride in their work.

These principles are experienced by many organizations with Software Engineering Institute’s Team Software Process. Therefore they are not any theoretical principles.

In generally, software quality means that supplier delivers the product on time, cost that was defined cost, and performs their all functions without defects. At this point every development team’s performance is very important. All of the big software projects are actually combination of smaller projects. If each of the smaller projects is developed with high quality, the overall project is developed with high quality, too.

Quality of processes on development lifecycle affects total quality of software product, too. Therefore it is very important to organize processes and organizational structure of the development team. Managers of the software companies are responsible for planning and organizing their development processes. If a software company has quality on development processes, they just need developers that can manage their work’s quality. Furthermore, all of the development team’s personals affect the whole quality of software product directly.

CHAPTER FOUR

SOFTWARE QUALITY STANDARD

Software project management is getting necessary in current software development process. All software companies want to develop their products and services faster. Most of companies use high technology environments and tools to develop more successful and quality products. However, software projects still have production and organizational problems. Capability Maturity Model Integration (CMMI) is the most known software quality standards in the software committee.

4.1 CMMI

Capability maturity models (CMM) deal with improving processes in organizations. They supply effective processes according to one or more disciplines. Capability Maturity models are developed for lots of disciplines. Some of the most important include models for systems engineering, software engineering, workforce management and development, and integrated product and process development (IPPD).

The SEI (Software Engineering Institute) created the first CMM design for software organizations in a book. The book that title "*The Capability Maturity Model: Guidelines for improving the software process*" is published in 1995.

The CMMI (Capability Maturity Model Integration) project is raised in 2000 for using multiple CMMs simultaneously. The major models are;

- The Capability Maturity Model for Software (SW-CMM)
- The System Engineering Capability Model (SECM)
- The Integrated Product Development Capability Maturity Model (IPD-CMM)

After we introduced The CMMI version 1.0, CMMI version 1.1 published in 2002 and also CMMI for development version 1.2 published in 2006.

The CMMI versions are improved by The CMMI Product Team according to received feedbacks from user community

CMMI models can be reference model that covers the development and maintenance processes applied to both products and services. Therefore the CMMI models can be used for many organizations on different industries such as banking, defense, automobile manufacturing and telecommunication. There are two different approaches while representing the CMMI on projects.

These approaches are;

- Continuous Representation
- Staged Representation

Each representation has some advantages to other. Actually some organizations use both representations for specific needs at their development process sometimes.

4.1.1 Continuous Representation

The major property of Continuous representation is supplying maximum flexibility when using a CMMI model for process improvement. An organization may choose to improve the performance of a single process-related trouble spot, or it can work on several areas that are closely aligned to the organization's business objectives. But there are some limitations on an organization's choice because of dependencies among the some process areas.

As a result, if you know about processes that need to be improved and if you understand the dependencies among process areas defined in CMMI, you can choose continuous representation.

4.1.2 Staged Representation

The stage representation is structured way to improve process management. The staged representation prescribes an order for implementing process areas according to maturity levels, which define the improvement path for an organization from the initial level to the optimizing level.

In short, if you do not know where to start and which processes to choose to improve, staged representation is a good choice for you.

4.2 CMMI Levels

All CMMI models reflect maturity levels in their design and content. A maturity level consists of related specific and generic practices for a predefined set of process areas that improve the organization's overall performance.

There are five maturity levels in the CMMI to describe an evolutionary path recommended for an organization that wants to improve development processes. These levels are;

- Performed Level
- Managed Level
- Defined Level
- Quantitatively Managed Level
- Optimized Level

If an organization satisfies all of the necessary goals of the process area or set of the process areas that are targeted, the organization reach next particular level.

4.2.1 Performed Level

The organization usually does not provide a stable environment to support the processes that are usually ad hoc and chaotic. Development period is unpredictable because of poor management and control. Therefore the organization’s, on performed level of CMMI, performance based on competence and heroics of personal. There are no process areas defined in performed level. There is no specific work on processes, just requirements in and the product out. In addition to this, in generally, performed level organizations exceed their budget and do not meet their schedules.

4.2.2 Managed Level

Software project management processes are documented and followed on managed level of the CMMI. Project management system is in place so software process is disciplined and controlled now. Organizations policies guide the project management and expectations. At maturity second level , the status of the work products and the delivery of services are visible to management at defined points such as major milestones. The work products and services satisfy their specified process descriptions, standards, and procedures. Therefore successful practices developed on earlier projects can be repeated.

4.2.3 Defined Level

Processes are well characterized and understood, and are described in standards, procedures, tools, and methods. These standard processes are used to establish consistency across the organization. Organization supports the projects by establishing (defined and documented processes):

- Common procedures, best practices, tailored processes
- Common measurements
- Training

If we compare defined level and managed level, defined level processes are described more strictly than Managed level. As we learn before, managed level organizations focuses on projects. Defined level emphasizes on organizations across the managed level

4.2.4 Quantitatively Managed Level

The organization and projects establish quantitative objectives for quality and process performance and use them as criteria in managing processes. Decisions that related with project management or development are made based on collected data according to measurements. Therefore quantitatively managed level requires measurements that have been defined and collected systematically.

Quality and process performance is understood in statistical terms and is managed throughout the life of the processes. A critical distinction between defined level and quantitatively managed level is the predictability of process performance. The performance of processes is controlled using statistical and other quantitative techniques, and is quantitatively predictable at quantitatively managed level.

4.2.5 Optimized Level

An organization continually improves its processes based on a quantitative understanding of the common causes of poor performance. The major property of the optimizing level is improving process performance through incremental and innovative process and technological improvements. Therefore everyone is involved in continuous improvement of the software process.

The difference from quantitatively managed level is concerning with addressing common causes of process and improve process performance, not specific causes of process.

4.3 Tools used to improve software quality

There are several management systems are used in software companies to improve their process and also their product quality. Some of these systems are;

- Requirement Management Systems
- Software Test Management Systems
- Software Performance Test Systems
- Defect Management Systems

All of these systems aim to do their objectives better and more automatically. We will learn about their general usage and purposes one by one.

4.3.1 Requirement Management Systems

Requirement management systems are the most important systems on software companies. As we told before, the product or service must satisfy customers' requirements. Therefore, if requirement management systems do not work successfully, it is impossible to create a software product that satisfies customers.

If requirements identified a bit difference from customers needs, there can be extra costs to reorganize requirement. Also if this mistake discover after software delivered to the customer, cost that correct to mistake getting higher. Therefore requirement management tools developed to prevent wrong identified requirements.

Most of the requirement management systems provide their users to some checklists to identify the exact requirement. Users can define requirement’s priority, schedule, risk probability, risk severity and risk level, cost, possible defects that requirement can cause, etc:

Requirement engineers of companies enter related information to these systems and get an output. This output must be analyzed by development team manager and planned according to the requirement schedule.

Some requirement management tools provides also requirement monitoring system to see last state of the requirements. For example, requirement engineers can see who develops it, its tests is started.

However, there are many requirement management systems; lots of companies develop their own requirement system according to their product area and their needs.

In conclusion, requirement management systems are the key systems that must work at software companies consistently.

4.3.2 Software Test Management Systems

Software testing systems are getting more important day by day. Software products must be tested detailed to find and correct defects before it delivered to customers.

There is a known fact that, using an independent third party test tool is the most affective test method. Programmers cannot identify their mistakes while testing their parts of product. The other affective test method is independent test department from development department but this manual test can be time spending when a standard tests of a product. Therefore, software development companies need software test management systems to test their products with different test strategies.

There are many testing types defined for software testing. Some of them are;

- Unit Testing
- Alpha Testing
- Module Testing
- System Testing
- White-box testing
- Black-box testing

Most of the software test management systems provide companies to some of these testing types to test their products. It is obvious that, it is impossible to find all possible defects while testing the software but theoretically 95 percent confidence on product can provide by testing.

As a result, software testing systems are necessarily part of the software development. The software product that has minimum number of defect is guaranteed to be satisfying customers’ requirements.

4.3.3 Software Performance Test Systems

Actually, performance test systems are test systems, too. But the main purpose of the system is different from test management systems. Performance test systems aim to understand project performance, not to find defects like test management systems.

Performance of software cannot measure with standard metrics. All of the software applications work on different systems. We can define performance testing that is process of identifying effectiveness or speed of software program or service.

There can be different metrics to measure a software programs performance. For example; time interval between operations' start and end can be evaluated as response time. We know that data density is very important on software product's performance. Therefore it can be used as a performance metric to measure software performance.

Performance test systems provide some tests on products. These tests are;

- Load testing
- Stress Testing
- Soak testing
- Spike testing
- Pre-requisites for Performance Testing

All of these tests can be done by performance testing tools according to specific requirement of software application. For example; if application supports multi user environment, load testing must run when lots of user connected to system. In addition to this, some of the performance testing systems provide automatic data creation function to test software with density data. After seeing the weak parts of the application on this condition, you can improve that parts performance.

In conclusion, performance test systems are useful systems for software companies. It gives to chance to see software product’s performance in worst conditions.

4.3.4 Defect Management Systems

Defect management systems are mostly used tools on software companies to improve their product’s quality. However, there are independent defect management systems, there are many project management systems and requirement management systems include defect management part.

Actually, the other management systems provide defect or request to defect management systems. As we told, test management systems’ aim to find defects before software product delivered to customer. Also, performance test systems can find out performance issues of the software products. If a requirement has developed with mistakes, it will return a defect on product. Therefore, defect management systems are most commonly used systems for software companies.

Next chapter, we will explain defect management systems with analysis of defect management systems.

CHAPTER FIVE

DEFECT MANAGEMENT SYSTEMS

The general purpose of the thesis, creating a defect management system that really affects the project performance and project quality. As we learned, defect management systems are tools those help to improve total software quality.

At first, we have to define some concepts to understand general problems of software products.

5.1 What is Defect?

"The term defect refers to something that is wrong with a program. It could be a misspelling, a punctuation mistake, or an incorrect program statement." Defects are introduced into a software product during every phase of software development. "A major source of defects that is often overlooked is requirements generation." Therefore, we can see defects on every process of the software development period requirement analysis to documentation.

5.2 What is Defect Management System?

Defect management systems are tools that can plan, monitor and manage defects of a software product. These systems are very useful systems for software development companies to manage defects that they faced on development period.

After giving definitions of these concepts, we can start to analyze defect management systems. At first, I will share properties and analysis of a defect management system.

5.3 Analysis of a Defect Management System

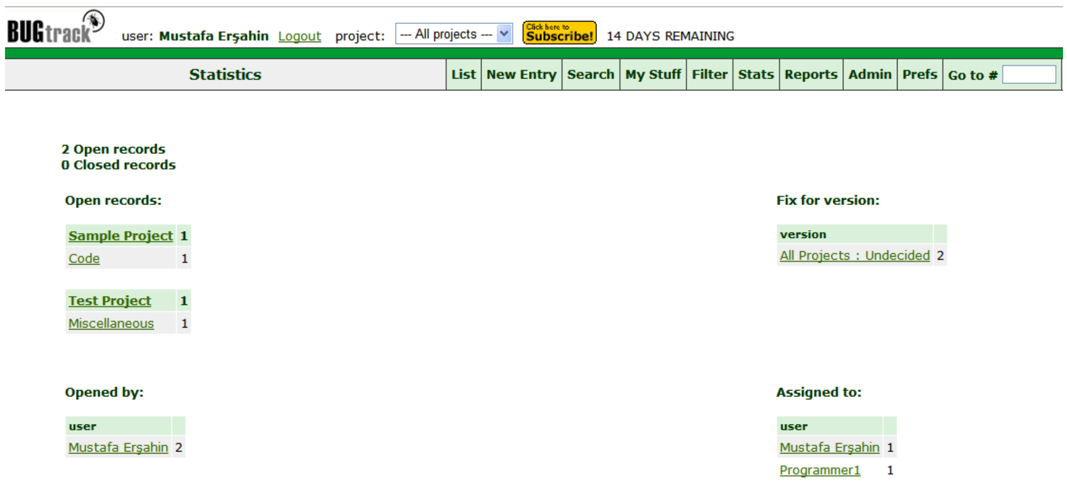
The one of the most common used defect management system is BUGTrack. This defect management systems professional edition is free for fourteen days trial usage. I tried the defect management system after my registration is approved. All of my comments and thoughts related with the application may call as user experience. I do not know anything about implementation details of this defect management system. Furthermore, I will share most of information about BUGTrack and its systems from its official web site.

5.3.1 BUGTrack

“BUGtrack is a comprehensive web-based project management and issue tracking system that helps to improve overall project organization, decrease bug resolution time, promote cross-functional cooperation, as well as plan and assess team performance at all levels.”

However security is big problem, web based systems are still more useful in today’s technology. BUGTrack provides their users maximum security with highest performance rate. Application security and physical security of your data is reliable on their hardware and software systems. As a result, security is not a problem for BUGTrack as they said. Therefore, you can reach the application and project’s data from wherever you are on the internet.

If you register to use trial version, BUGTrack creates a user with an administrator role. After log in, you can see statistics panel and menu. The menu includes list, new entry, search, my stuff, filter, statistics, reports, preferences and administrator panel and we can see opening screen at Figure 5.1. Defect, issue or requests counts listed in statistics grouped by projects. This statistics panel shows us total works assigned to related user.



user: Mustafa Erşahin

project: --- All projects ---

14 DAYS REMAINING

2 Open records
0 Closed records

Open records:

Sample Project	1
Code	1
Test Project	1
Miscellaneous	1

Fix for version:

version	
All Projects : Undecided	2

Opened by:

user	
Mustafa Erşahin	2

Assigned to:

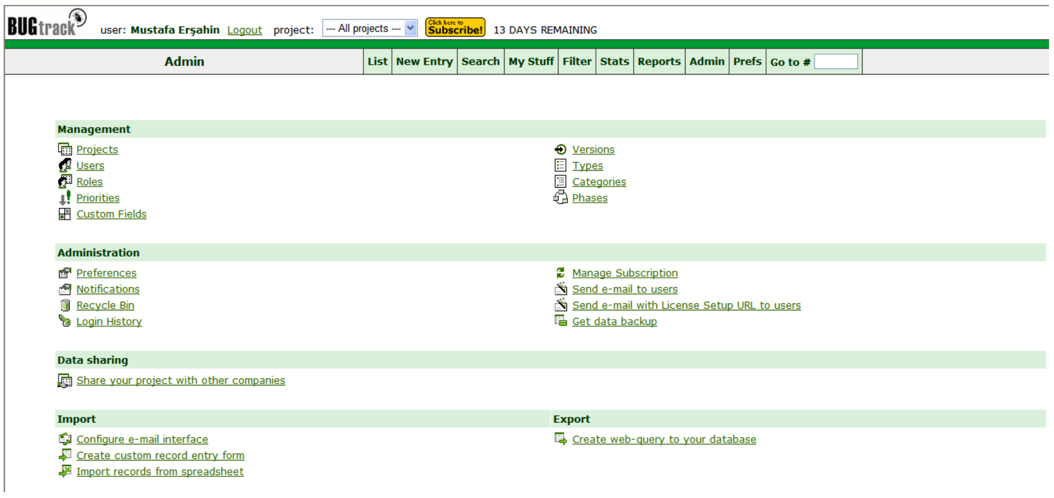
user	
Mustafa Erşahin	1
Programmer1	1

Figure 5.1 Statistics panel of BUGTrack

The most important panel for projects managers is admin panel. This panel includes all definitions such as project definition, user definition, group definition and some other authority definitions. Figure 5.2 shows us managers’ panel of users that has an administrator authorization. “Management” and “Administration” parts are generally used for defining organizational lifecycle.

The main approach of the BUGTrack defect management application is dynamic system control. Projects, users, groups, roles are created by administrator or manager dynamically. Actually, it must be dynamically according to the fact that, there is not any standard project or organization plan that they support for customers. Every organization or company can use the BUGTrack management system.

The other parts of administrator panel are “Data Sharing”, “Import” and “Export”. These parts are used for integration with companies’ original system sources such as databases or spreadsheets. In addition to this, BUGTrack application supplies a data sharing system to customers. If you want to share your projects with any other company, you can do it by creating new share and license.


BUGtrack
user: Mustafa Erşahin Logout project: --- All projects --- Subscribe! 13 DAYS REMAINING
Admin
List | New Entry | Search | My Stuff | Filter | Stats | Reports | Admin | Prefs | Go to #

Management

- Projects
- Users
- Roles
- Priorities
- Custom Fields
- Versions
- Types
- Categories
- Phases

Administration

- Preferences
- Notifications
- Recycle Bin
- Login History
- Manage Subscription
- Send e-mail to users
- Send e-mail with License Setup URL to users
- Get data backup

Data sharing

- Share your project with other companies

Import

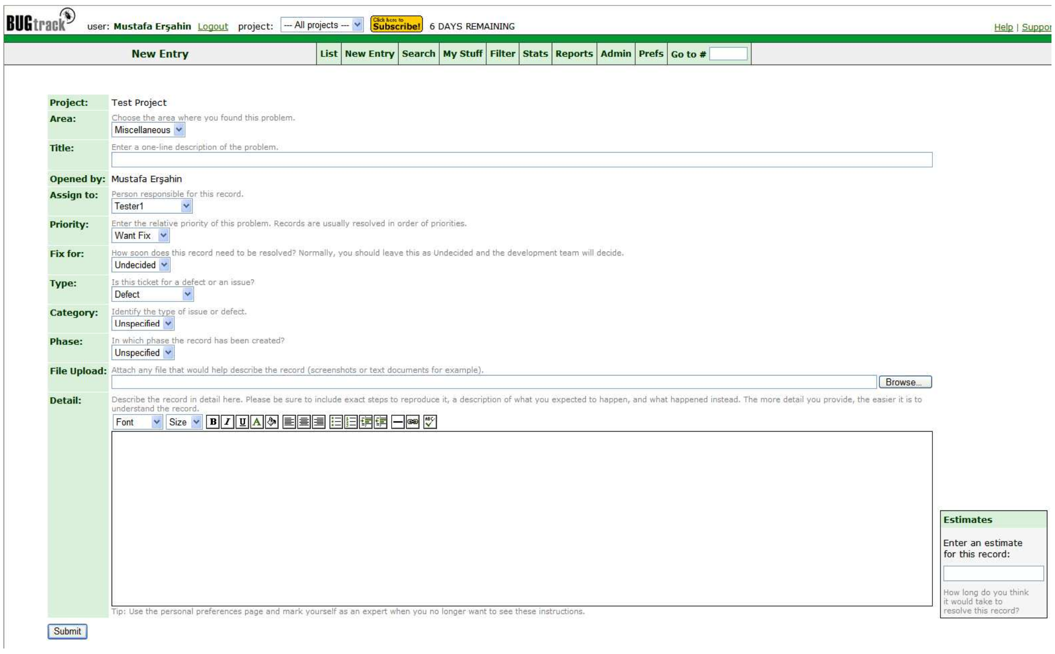
- Configure e-mail interface
- Create custom record entry form
- Import records from spreadsheet

Export

- Create web-query to your database

Figure 5.2 Administrator Panel of BUGTrack application

After the main definitions completed, administrators or users can open new entry and assign it to related user. Some information of an entry must be filled by opener such as project, area, title, priority, type, category, phase, detailed description and screenshot if necessary. Also entry opener can enter estimated response time. We can see the new entry screen in Figure 5.3 below.


BUGtrack
user: Mustafa Erşahin Logout project: --- All projects --- Subscribe! 6 DAYS REMAINING
Help | Suppor
New Entry
List | New Entry | Search | My Stuff | Filter | Stats | Reports | Admin | Prefs | Go to #
Browse...
Font
Size
Submit

Project:	Test Project
Area:	Choose the area where you found this problem. Miscellaneous
Title:	Enter a one-line description of the problem.
Opened by:	Mustafa Erşahin
Assign to:	Person responsible for this record. Tester1
Priority:	Enter the relative priority of this problem. Records are usually resolved in order of priorities. Want Fix
Fix for:	How soon does this record need to be resolved? Normally, you should leave this as Undecided and the development team will decide. Undecided
Type:	Is this ticket for a defect or an issue? Defect
Category:	Identify the type of issue or defect. Unspecified
Phase:	In which phase the record has been created? Unspecified
File Upload:	Attach any file that would help describe the record (screenshots or text documents for example).
Detail:	Describe the record in detail here. Please be sure to include exact steps to reproduce it, a description of what you expected to happen, and what happened instead. The more detail you provide, the easier it is to understand the record.

Tip: Use the personal preferences page and mark yourself as an expert when you no longer want to see these instructions.

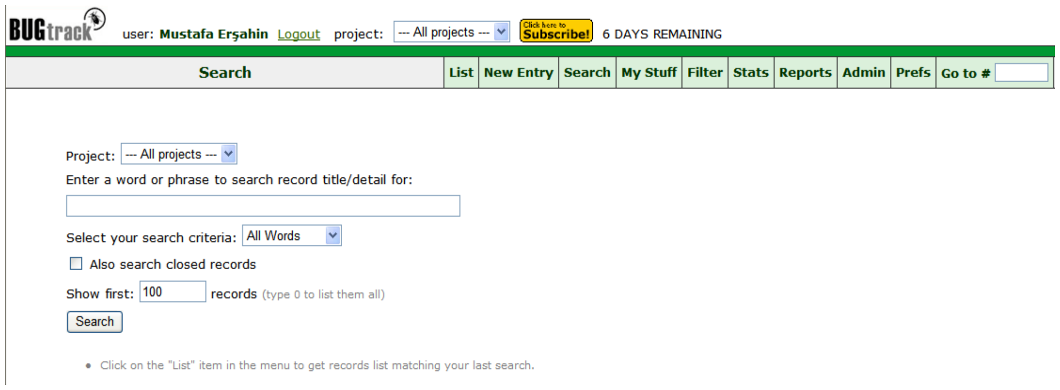
Estimates

Enter an estimate for this record:

How long do you think it would take to resolve this record?

Figure 5.3 New Entry screenshot of BUGTrack

However, BUGTrack defect management system suggests user not to fill some parts except development team such as “Fix for”. Therefore, there is information above the area to warn user. The other panels of BUGTrack defect management system are “Search”, “Filter” and “My stuff”. “My Stuff” panel is used for listing entries that assigned to each user. “Search” and “Filter” panels are used for listing entries according to the related search key or filters. Search panel shown in Figure 5.4 and Filter screen in Figure 5.5. The main difference between “Search” and “Filter” is search key. Filter menu has no keys, it list all entries according to related filters such as project, entry type, entry status, category etc.

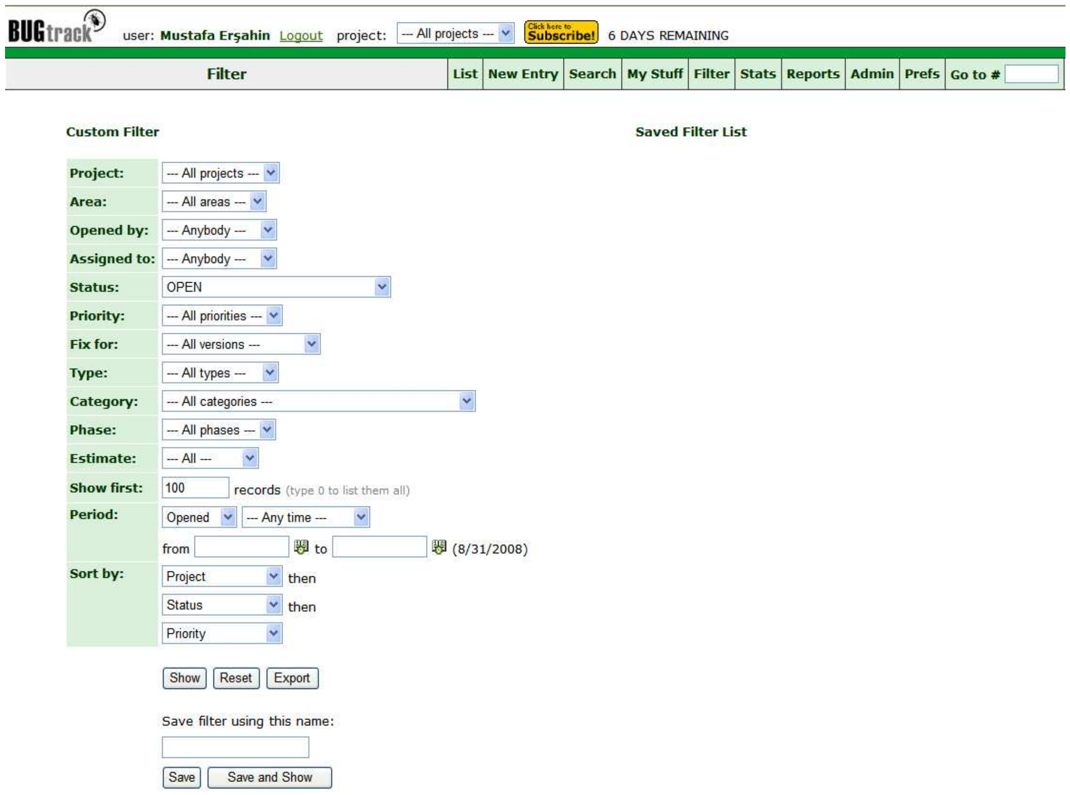


Project: --- All projects ---
Enter a word or phrase to search record title/detail for:
Select your search criteria: All Words
☐ Also search closed records
Show first: 100 records (type 0 to list them all)
Search

- Click on the "List" item in the menu to get records list matching your last search.

Figure 5.4 Search panel of BUGTrack system

“Prefs” panel is used for system and user preferences. User name, e-mail, location and time zone properties can update from this panel. In addition to this, users can define and update three listing order such as priority, project, status, category, type or newest entry.



user: Mustafa Erşahin Logout project: --- All projects --- Click here to Subscribe! 6 DAYS REMAINING

Filter | List | New Entry | Search | My Stuff | Filter | Stats | Reports | Admin | Prefs | Go to #

Custom Filter

Saved Filter List

Project:	--- All projects ---
Area:	--- All areas ---
Opened by:	--- Anybody ---
Assigned to:	--- Anybody ---
Status:	OPEN
Priority:	--- All priorities ---
Fix for:	--- All versions ---
Type:	--- All types ---
Category:	--- All categories ---
Phase:	--- All phases ---
Estimate:	--- All ---
Show first:	100 records (type 0 to list them all)
Period:	Opened --- Any time ---
	from to (8/31/2008)
Sort by:	Project then
	Status then
	Priority

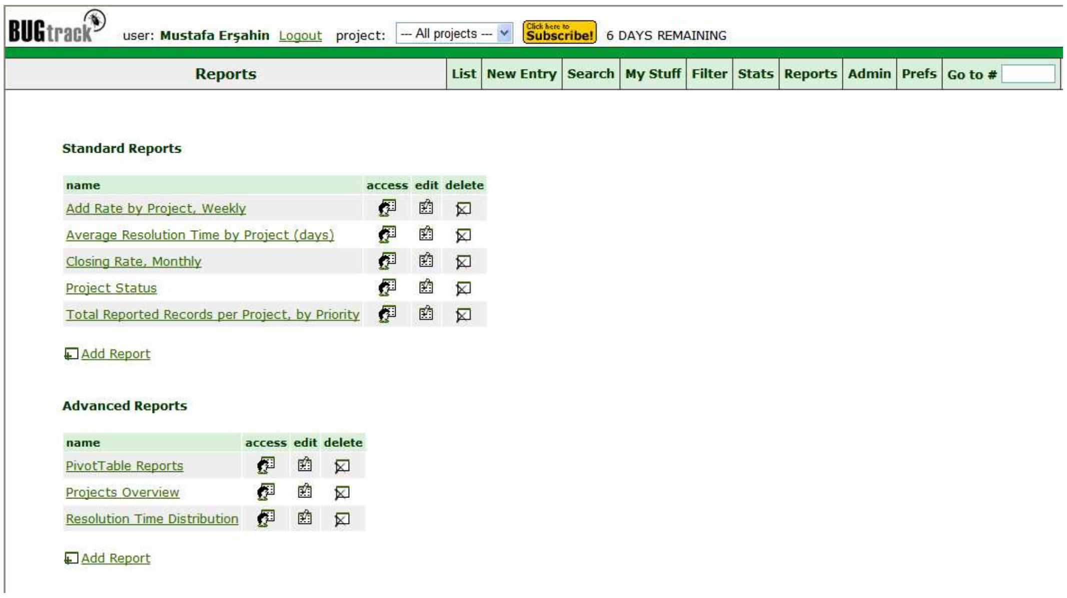
Show Reset Export

Save filter using this name:

Save Save and Show

Figure 5.5 Filter panel of BUGTrack system

The last panel of the BUGTrack system is “Reports” panel. However reports can be prepared dynamically, this panel includes some standard reports and some advanced reports seen in Figure 5.6. We analyze BUGTrack defect management system on its general properties. This defect management system lies on dynamic system architecture.



Standard Reports

name	access	edit	delete
Add Rate by Project, Weekly			
Average Resolution Time by Project (days)			
Closing Rate, Monthly			
Project Status			
Total Reported Records per Project, by Priority			

Add Report

Advanced Reports

name	access	edit	delete
PivotTable Reports			
Projects Overview			
Resolution Time Distribution			

Add Report

Figure 5.6 Reports panel of BUGTrack system

We introduce BUGTrack defect management system generally. This system lies on managing the defects or issues over development teams. If we want to summarize this defect management system's properties such as;

- Web based management system with overall security guarantee.
- System architecture supplies dynamically design and organizations.
- Advanced reporting and e-mail notifications to users.
- Source control integration.

A lot of software development companies use BUGTrack as a defect management system. Although BUGTrack is a successful defect management system, if software development process of the company is wrong or it does not use a statistics of this system, BUGTrack is not able to improve the quality of software development process.

This system and most of defect management systems are generally form of data collective systems. If you do not classify and analyze the collected data, defect management system works as a defect monitoring system.

After general view of a defect management system, BUGTrack, we will introduce the new defect management system UHISYS and try to find out difference between BUGTrack defect management system and UHISYS defect management system in next chapter.

CHAPTER SIX

UHISYS

UNIVERA DEFECT MANAGEMENT SYSTEM

UHISYS is shortening of the Univera Defect and Request management system. This system is combination of defect management and defect analyzing system. The main purpose of this defect management system is analyzing the current software development process and tries to find out current system's weak parts.

6.1 Why developing new system?

There are some necessary reasons to develop new defect management system to use in company that called Univera Bilgisayar Sistemleri A.Ş. The most important reasons are listed below;

- The defect management systems currently used on software development companies do not have Turkish language support. Some people of development team do not know English enough to use a system in English.
- Most of them have huge cost to buy. Their systems can be licensed module by module. Therefore the overall system is getting more expensive.
- This defect management system can develop with runtime experience and can be a new product for sale.

Managers of the software development team decided to develop new defect management system because of these reasons and some other reasons related in companies development strategy.

6.2 Differences from other systems

At first, UHISYS defect management system is developed like company based architecture. Most of the definitions are entered manually to database which overall system data stored. This system is running on one of the company’s server that is secured by IT (Information Technology) manager of the company. Across the local server usage, this system supported with web services technology. Therefore users can reach the system over internet.

There are three different steps on UHISYS defect management system. These steps are;

- Data Collection
- Analyzing the statistical results and think about new strategies
- Apply or perform new strategies to projects.
- Evaluate performance of the system or developers

After all steps passed, company will have information about performance and quality ratio to understand project’s improvement according to the usage of defect management system.

6.3 System lifecycle

Every software development company has some difficulties or some problems to manage software development period. The main problem on software companies about development and maintenance periods is having troublous organizational structure to divide overall development or maintenance processes.

There are three main departments in our organization. These departments are;

- Software development department
- Software testing department
- Customer support department

All of these departments have at least one manager to control processes and manage people who work on department. They assign jobs to people and control their department’s processes on software development lifecycle.

In this organization, there are two different cycles according to development process. One of these cycles includes software development and test departments. This cycle runs on while developing and integrating new requirement to the project. Programmer develops new requirement according to the requirement analysis that prepared by project consultants. After requirement developed, test process starts by testers. If testers find an error or errors, they must supply all the system properties, detailed information of test conditions and data to programmer to solve the problem quickly as possible. Therefore the first cycle between software developers and testers are completed.

The other cycle is more complicated cycle. This cycle contains all departments of development team and it starts to run by a last user problem that received by call center. Call center gives online help to customers and response their questions and problems. If there is an exactly problem on project, customer support department opens a new defect record and send this report to test department to identify the problem with all cases and conditions. After that, the first cycle runs again between developers and testers until the defect corrected.

The development process contains these two cycles in software development team on this approach. Therefore the main structure of the defect management system must supply the organizational structure and authorization mechanism to manage the whole system.

6.3.1 Authorization

In generally, authorization mechanism of the projects is related with number of profiles that use the system. As UHISYS is a software project, its authorization is related with user roles of the software development team.

It is said before, three different departments in development process life cycle. While defining the function of the test department and software support department we said they open defect records that they identifies. Therefore one of the user profiles is “Test-Support” for both of these department’s users.

A user that has “Test-Support” privileges can open a new defect record, modify an existing defect definition, assign the record to another user, cancel or close a defect record.

One another user profile is for developers. Developers cannot open any new defect record or cannot modify any record’s definition. They can change defect state to “In Coding” and after coding process finished they change state to “Fixed”.

The last user profile is “Manager”. Manager profile has all access to system. They can open, edit and cancel defect records, assign to someone. Actually, they are also developers, too. Therefore, they can fix a defect and change its state to “Fixed”.

All of these three user profile can use all statistical reports, search and filters. Every user can see another user’s stuff with general view.

It is time to introduce UHISYS user interface. Sometimes there will be comparisons with the other defect management system that analyzed before.

6.3.2 UHISYS User Interface

Developing web based systems is more difficult and complex than windows based systems. There are main problems such as security of the system, session management, etc:

UHISYS defect management runs on one of company’s local servers. Users can reach to login page of the system by using internet browser. The login screen shown in Figure 6.1. After user logged in the system, the main page opens as a pop-up page. Therefore, there is no access to main page without login.

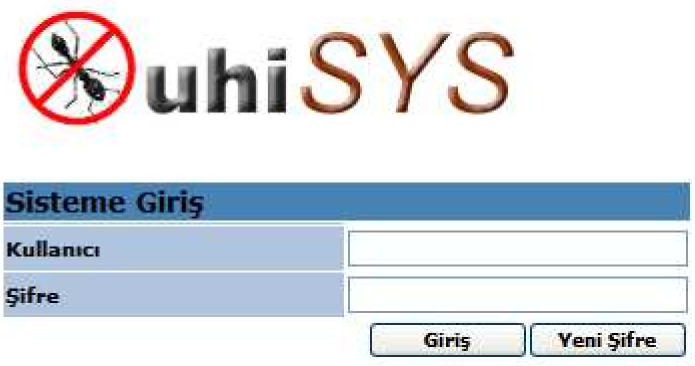


Figure 6.1 Login page of UHISYS

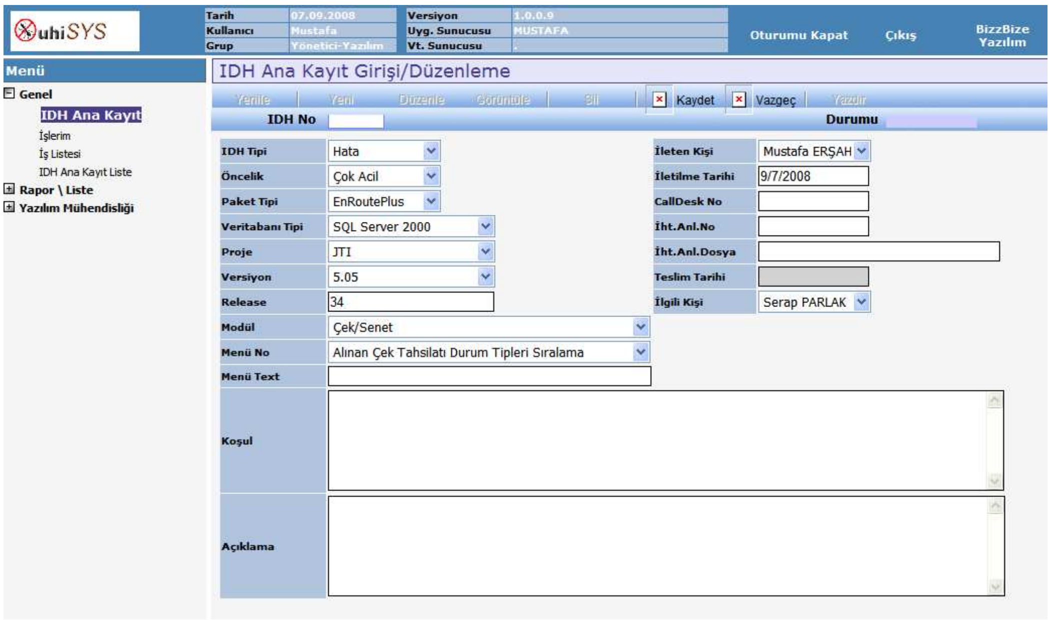
The general information about user, user privileges and system is top of the page. Left side of the page includes operational links as a tree structure. There are three different headers in this tree that shown in Figure 6.2.

One of them is general operations such as viewing a defect record, the other one for listing user’s stuff, one for listing all defect records and the last one for opening new defect record.

The second part of the three includes some reports such as defect count group by module, fixed defect count group by developer, weekly statistics for defect reason and weekly distribution of defects on projects.

The last part of the tree is software engineering part. This part includes line of code by projects, project improvement on lines of code metric.

All of these tree items can provide us company based defect monitoring and management system. There are some screenshots from UHISYS defect management system to see and understand general view.



Tarih	07.09.2008	Versiyon	1.0.0.9
Kullanıcı	Mustafa	Uyg. Sunucusu	MUSTAFA
Grup	Yönetici-Yazılım	Vt. Sunucusu	.

IDH Ana Kayıt Girişi/Düzenleme

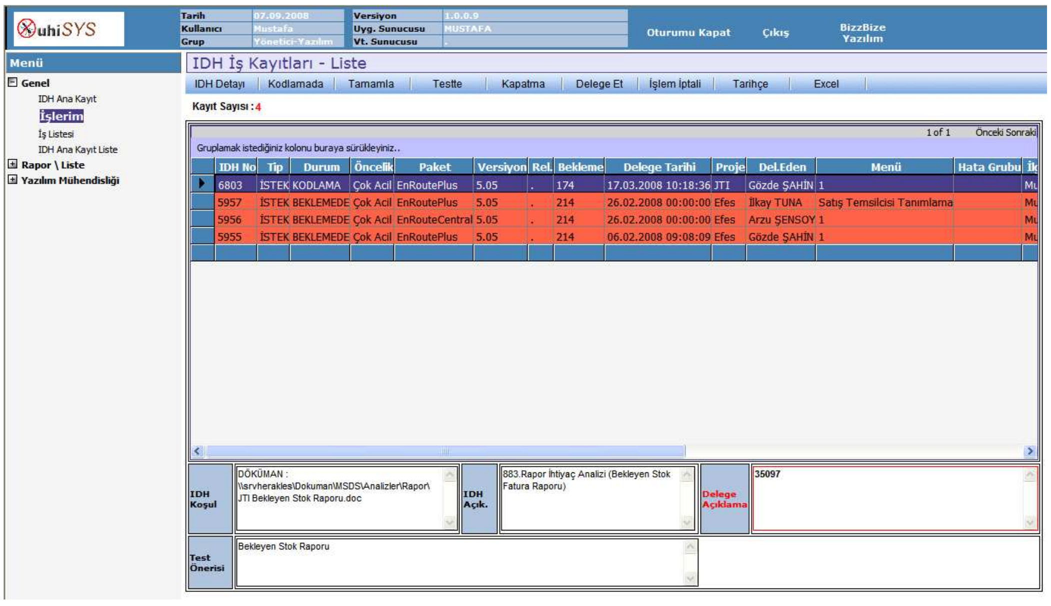
IDH No		Durumu	
IDH Tipi	Hata	İleten Kişi	Mustafa ERŞAH
Öncelik	Çok Acil	İletilme Tarihi	9/7/2008
Paket Tipi	EnRoutePlus	CallDesk No	
Veritabanı Tipi	SQL Server 2000	İht.Anl.No	
Proje	JTI	İht.Anl.Dosya	
Versiyon	5.05	Teslim Tarihi	
Release	34	İlgili Kişi	Serap PARLAK
Modül	Çek/Senet		
Menü No	Alınan Çek Tahsilatı Durum Tipleri Sıralama		
Menü Text			
Koşul			
Açıklama			

Figure 6.2 The main page that defects or requests open

There are many areas to fill while opening a defect. Test or support department's user must fill these fields. These fields identify records either a defect or a request. After that they show priority, severity and their conditions of the records.

The major fields are priority, product type, database type, project, version, release, module and menu key. If type of the record is defect and comes from last user, the call desk number must be filled by customer support department. If the tester or support user can identify exact condition that cause a defect it should write condition clearly. In addition to this, if they have an idea or necessary information about the defect, they can write related information to explanation part.

There are five priority levels on our system. These are low, medium, high, immediate, high immediate. All of the priority levels have different color while listing such as red for immediate, orange for high yellow for low etc. Therefore, all users of the system can know defects or request priority before read detailed when they see on the list. For example; all of the defects priorities are very immediate on my list that seen in Figure 6.3.



Tarih	07.09.2008
Kullanıcı	Mustafa
Grup	Yönetici-Yazılım
Versiyon	1.0.0.9
Uyg. Sunucusu	MUSTAFA
Vt. Sunucusu	.

IDH İş Kayıtları - Liste

Kayıt Sayısı : 4

Gruplamak istediğiniz kolonu buraya sürükleyiniz..

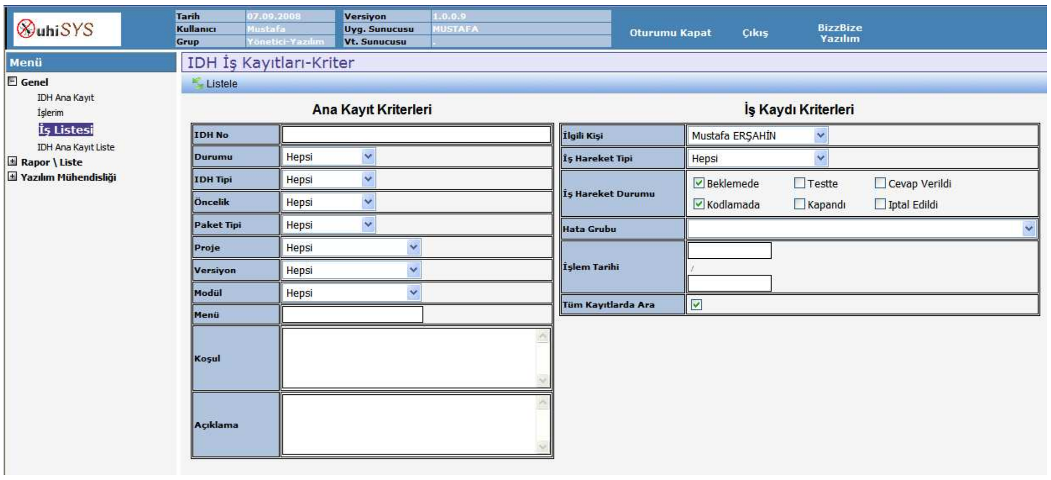
IDH No	Tip	Durum	Öncelik	Paket	Versiyon	Rel.	Bekleme	Delege Tarihi	Proje	Del.Eden	Menü	Hata Grubu	İk
6803	İSTEK	KODLAMA	Çok Acil	EnRoutePlus	5.05	.	174	17.03.2008 10:18:36	JTI	Gözde ŞAHİN	1		Mu
5957	İSTEK	BEKLEMEDE	Çok Acil	EnRoutePlus	5.05	.	214	26.02.2008 00:00:00	Efes	İlkay TUNA	Satış Temsilcisi Tanımlama		Mu
5956	İSTEK	BEKLEMEDE	Çok Acil	EnRouteCentral	5.05	.	214	26.02.2008 00:00:00	Efes	Arzu ŞENSOY	1		Mu
5955	İSTEK	BEKLEMEDE	Çok Acil	EnRoutePlus	5.05	.	214	06.02.2008 09:08:09	Efes	Gözde ŞAHİN	1		Mu

IDH Koşul	DÖKÜMAN : \\srvherakles\Dokuman\MSDS\Analizler\Rapor\JTI Bekleyen Stok Raporu.doc	IDH Açık.	883.Rapor İhtiyaç Analizi (Bekleyen Stok Fatura Raporu)	Delege Açıklama	35097
Test Önerisi	Bekleyen Stok Raporu				

Figure 6.3 Example list of a developer’s stuff

Developers and testers use this stuff list often. They can see details of defect records; they can change records state to “In Coding”, ‘In Test’. After that they delegate related user or manager. Developer has to change record state to “In Coding” when he/she starts and then change state to “Fixed” after checked in related files to the project. There is a difference between testers and developers after they completed their work on a defect. Developers must select an important property that called defect reason classification to identify real problems that causes defects on system. This property of UHISYS will be explained later with more detailed.

It is said before; all of the users can reach the whole records. A defect occurs with same error message with a defect that fixed before. Therefore, it is easy to fixed new defect according to fixed defect's information. If number of records getting higher it is difficult to find to related record. This menu includes very detailed filter to find exact results. In Figure 6.4, you can see the filters to find a defect.



Tarih	07.09.2008	Versiyon	1.0.0.9
Kullanıcı	Mustafa	Uyg. Sunucusu	MUSTAFA
Grup	Yönetici-Yazılım	Vt. Sunucusu	.

IDH İş Kayıtları-Kriter

Ana Kayıt Kriterleri

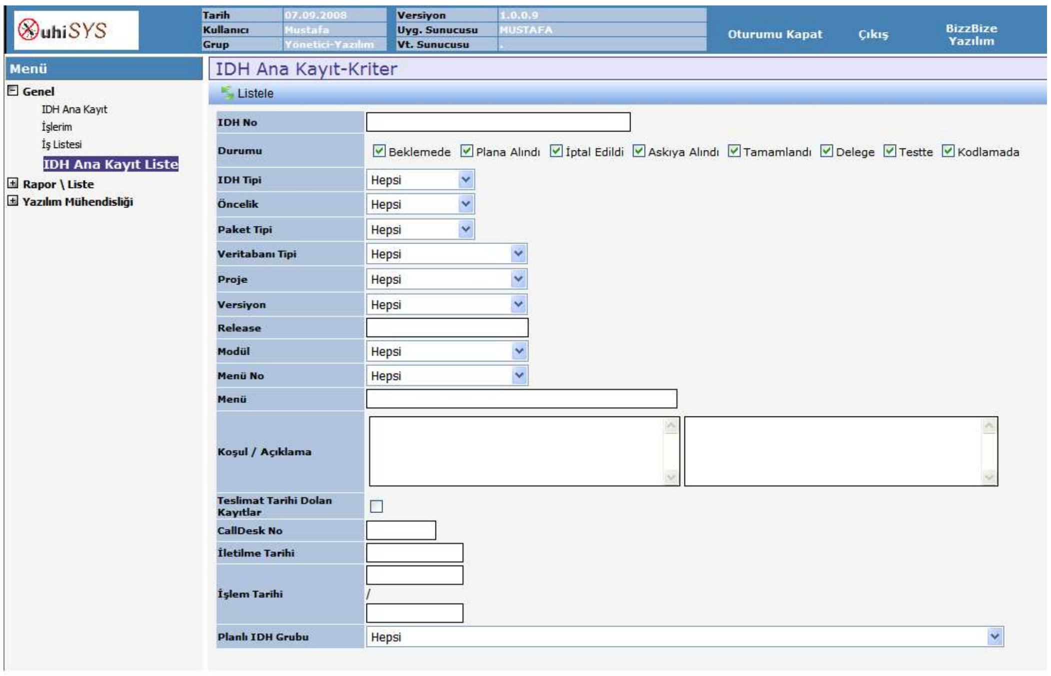
IDH No	
Durumu	Hepsi
IDH Tipi	Hepsi
Öncelik	Hepsi
Paket Tipi	Hepsi
Proje	Hepsi
Versiyon	Hepsi
Modül	Hepsi
Menü	
Koşul	
Açıklama	

İş Kaydı Kriterleri

İlgili Kişi	Mustafa ERŞAHİN
İş Hareket Tipi	Hepsi
İş Hareket Durumu	✓ Beklemede ☐ Testte ☐ Cevap Verildi ✓ Kodlamada ☐ Kapandı ☐ Iptal Edildi
Hata Grubu	
İşlem Tarihi	/
Tüm Kayıtlarda Ara	✓

Figure 6.4 Defect transaction research for users

At this point, the master record and operational records of a defect is keeping different physical tables. All users have access to see only defect transaction records. Defect master records can be reachable for users that have manager privileges. This master defect records can be planned by managers. We can say defect or request management can be done from this menu context showed in Figure 6.5.



Tarih	07.09.2008
Kullanıcı	Mustafa
Grup	Yönetici-Yazılım
Versiyon	1.0.0.9
Uyg. Sunucusu	MUSTAFA
Vt. Sunucusu	.

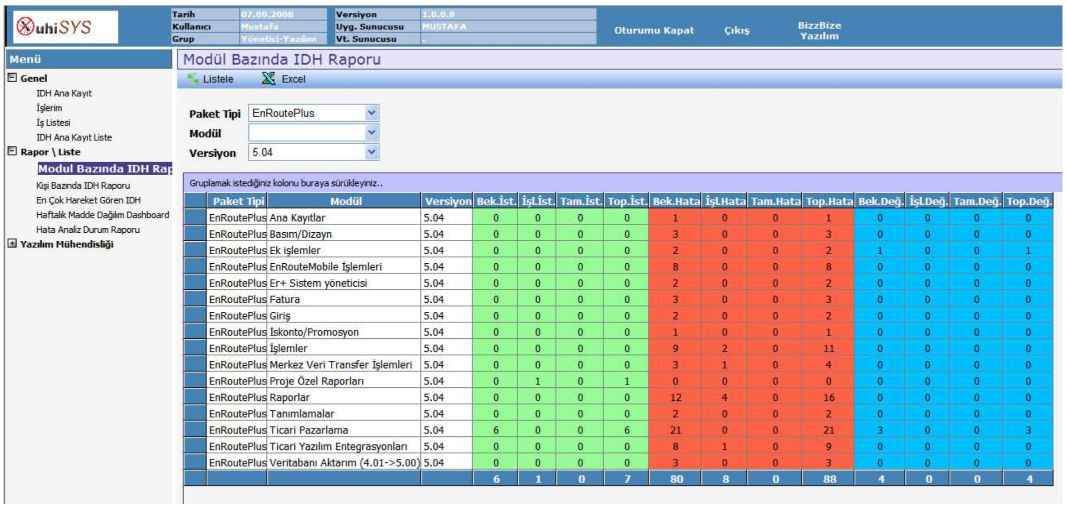
IDH Ana Kayıt-Kriter

IDH No	
Durumu	✓ Beklemede ✓ Plana Alındı ✓ İptal Edildi ✓ Askıya Alındı ✓ Tamamlandı ✓ Delege ✓ Testte ✓ Kodlamada
IDH Tipi	Hepsi
Öncelik	Hepsi
Paket Tipi	Hepsi
Veritabanı Tipi	Hepsi
Proje	Hepsi
Versiyon	Hepsi
Release	
Modül	Hepsi
Menü No	Hepsi
Menü	
Koşul / Açıklama	
Teslimat Tarihi Dolan Kayıtlar	☐
CallDesk No	
İletilme Tarihi	
İşlem Tarihi	/
Planlı IDH Grubu	Hepsi

Figure 6.5 Master defect record list

Real operation part of the defect management system was introduced up to now. One of the other parts of the UHISYS defect management system is reports. There are some statistical reports according to the collected data before. These reports are lies on defect numbers on module, user and project.

Figure 6.6 shows us the module based defect and request records in overall system. We can use filters to list exact records according to the packet type, module and version criteria.



Tarih	07.09.2008	Versiyon	1.0.0.9
Kullanıcı	Mustafa	Uyg. Sunucusu	MUSTAFA
Grup	Yönetici-Yazılım	Vt. Sunucusu	.

Modül Bazında IDH Raporu

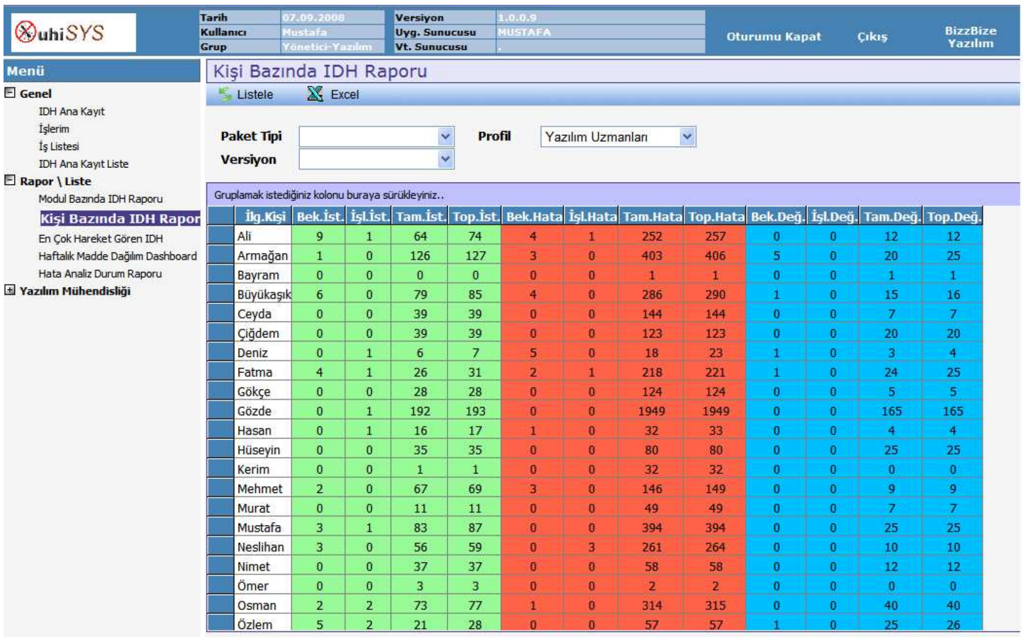
Paket Tipi: EnRoutePlus
Modül:
Versiyon: 5.04

Paket Tipi	Modül	Versiyon	Bek.İst.	İşl.İst.	Tam.İst.	Top.İst.	Bek.Hata	İşl.Hata	Tam.Hata	Top.Hata	Bek.Değ.	İşl.Değ.	Tam.Değ.	Top.Değ.
EnRoutePlus	Ana Kayıtlar	5.04	0	0	0	0	1	0	0	1	0	0	0	0
EnRoutePlus	Basım/Dizayn	5.04	0	0	0	0	3	0	0	3	0	0	0	0
EnRoutePlus	Ek işlemler	5.04	0	0	0	0	2	0	0	2	1	0	0	1
EnRoutePlus	EnRouteMobile İşlemleri	5.04	0	0	0	0	8	0	0	8	0	0	0	0
EnRoutePlus	Er+ Sistem yöneticisi	5.04	0	0	0	0	2	0	0	2	0	0	0	0
EnRoutePlus	Fatura	5.04	0	0	0	0	3	0	0	3	0	0	0	0
EnRoutePlus	Giriş	5.04	0	0	0	0	2	0	0	2	0	0	0	0
EnRoutePlus	İskonto/Promosyon	5.04	0	0	0	0	1	0	0	1	0	0	0	0
EnRoutePlus	İşlemler	5.04	0	0	0	0	9	2	0	11	0	0	0	0
EnRoutePlus	Merkez Veri Transfer İşlemleri	5.04	0	0	0	0	3	1	0	4	0	0	0	0
EnRoutePlus	Proje Özel Raporları	5.04	0	1	0	1	0	0	0	0	0	0	0	0
EnRoutePlus	Raporlar	5.04	0	0	0	0	12	4	0	16	0	0	0	0
EnRoutePlus	Tanımlamalar	5.04	0	0	0	0	2	0	0	2	0	0	0	0
EnRoutePlus	Ticari Pazarlama	5.04	6	0	0	6	21	0	0	21	3	0	0	3
EnRoutePlus	Ticari Yazılım Entegrasyonları	5.04	0	0	0	0	8	1	0	9	0	0	0	0
EnRoutePlus	Veritabanı Aktarım (4.01->5.00)	5.04	0	0	0	0	3	0	0	3	0	0	0	0
			6	1	0	7	80	8	0	88	4	0	0	4

Figure 6.6 Module based defect and request report

The other report is used based defect and request records. This report shows us average personal performance in Figure 6.7. This report can be filtered by packet type, version and user profile.

There is a different kind of report here to find a defect that has the most transaction over the defects. It can be seen in Figure 6.8. This report provides some important information. Maybe communication problem with test user and developer causes these transactions. It is a known fact that, every transaction on developer or test user has a cost because of related time to spend. Therefore we can find a defect that has more transactions by using packet type, version and state filters to identify the defect that has more cost for development team.



Tarih 07.09.2008
Kullanıcı Mustafa
Grup Yönetici-Yazılım
Versiyon 1.0.0.9
Uyg. Sunucusu MUSTAFA
Vt. Sunucusu .

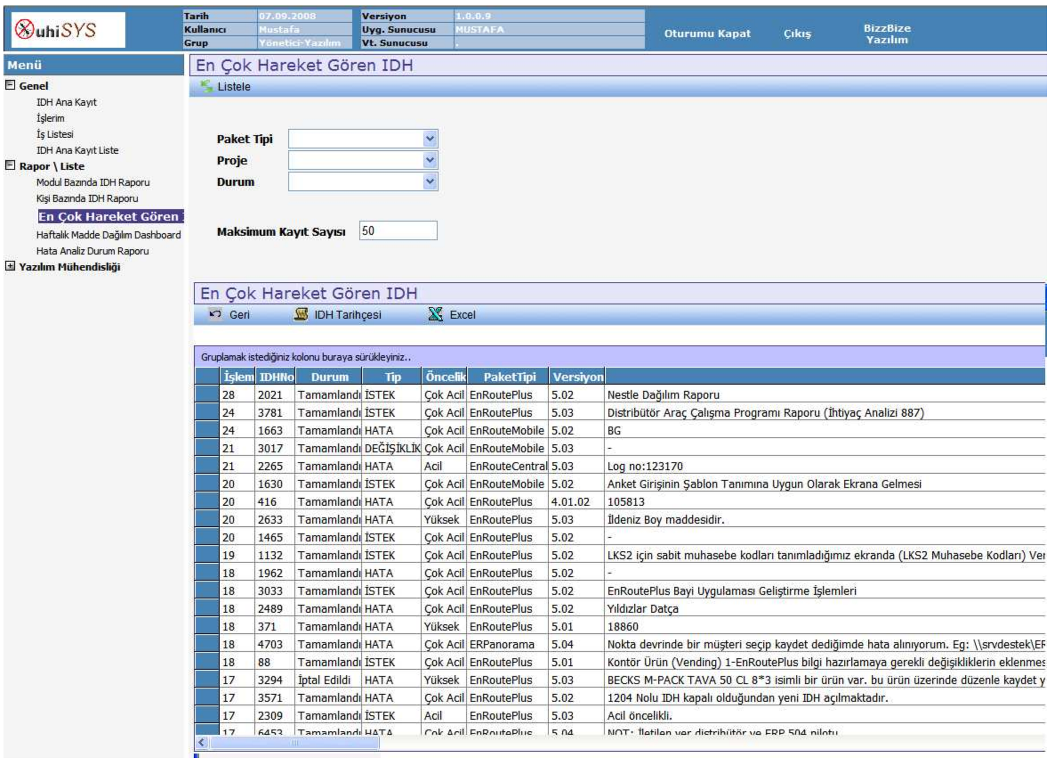
Kişi Bazında IDH Raporu

Paket Tipi
Versiyon
Profil Yazılım Uzmanları

Gruplamak istediğiniz kolonu buraya sürükleyiniz..

İlg.Kişi	Bek.İst.	İşl.İst.	Tam.İst.	Top.İst.	Bek.Hata	İşl.Hata	Tam.Hata	Top.Hata	Bek.Değ.	İşl.Değ.	Tam.Değ.	Top.Değ.
Ali	9	1	64	74	4	1	252	257	0	0	12	12
Armağan	1	0	126	127	3	0	403	406	5	0	20	25
Bayram	0	0	0	0	0	0	1	1	0	0	1	1
Büyükaşık	6	0	79	85	4	0	286	290	1	0	15	16
Ceyda	0	0	39	39	0	0	144	144	0	0	7	7
Çiğdem	0	0	39	39	0	0	123	123	0	0	20	20
Deniz	0	1	6	7	5	0	18	23	1	0	3	4
Fatma	4	1	26	31	2	1	218	221	1	0	24	25
Gökçe	0	0	28	28	0	0	124	124	0	0	5	5
Gözde	0	1	192	193	0	0	1949	1949	0	0	165	165
Hasan	0	1	16	17	1	0	32	33	0	0	4	4
Hüseyin	0	0	35	35	0	0	80	80	0	0	25	25
Kerim	0	0	1	1	0	0	32	32	0	0	0	0
Mehmet	2	0	67	69	3	0	146	149	0	0	9	9
Murat	0	0	11	11	0	0	49	49	0	0	7	7
Mustafa	3	1	83	87	0	0	394	394	0	0	25	25
Neslihan	3	0	56	59	0	3	261	264	0	0	10	10
Nimet	0	0	37	37	0	0	58	58	0	0	12	12
Ömer	0	0	3	3	0	0	2	2	0	0	0	0
Osman	2	2	73	77	1	0	314	315	0	0	40	40
Özlem	5	2	21	28	0	0	57	57	1	0	25	26

Figure 6.7 Used based defect and request report



Tarih 07.09.2008
Kullanıcı Mustafa
Grup Yönetici-Yazılım
Versiyon 1.0.0.9
Uyg. Sunucusu MUSTAFA
Vt. Sunucusu .

En Çok Hareket Gören IDH

Paket Tipi
Proje
Durum

Maksimum Kayıt Sayısı 50

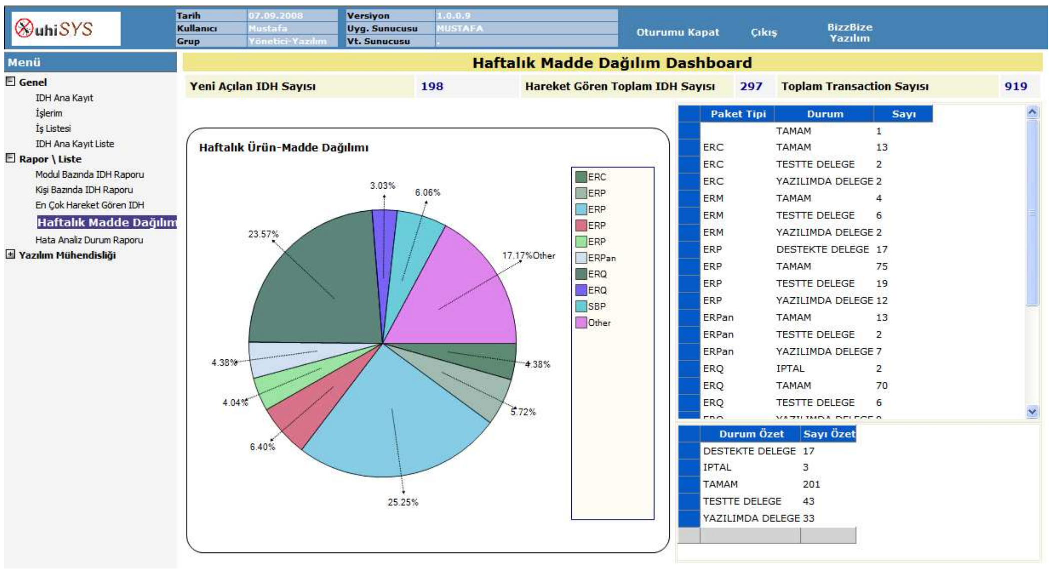
En Çok Hareket Gören IDH

Gruplamak istediğiniz kolonu buraya sürükleyiniz..

İşlem	IDHNo	Durum	Tip	Öncelik	PaketTipi	Versiyon	
28	2021	Tamamlandı	İSTEK	Çok Acil	EnRoutePlus	5.02	Nestle Dağılım Raporu
24	3781	Tamamlandı	İSTEK	Çok Acil	EnRoutePlus	5.03	Distribütör Araç Çalışma Programı Raporu (İhtiyaç Analizi 887)
24	1663	Tamamlandı	HATA	Çok Acil	EnRouteMobile	5.02	BG
21	3017	Tamamlandı	DEĞİŞİKLİK	Çok Acil	EnRouteMobile	5.03	-
21	2265	Tamamlandı	HATA	Acil	EnRouteCentral	5.03	Log no:123170
20	1630	Tamamlandı	İSTEK	Çok Acil	EnRouteMobile	5.02	Anket Girişinin Şablon Tanımına Uygun Olarak Ekrana Gelmesi
20	416	Tamamlandı	HATA	Çok Acil	EnRoutePlus	4.01.02	105813
20	2633	Tamamlandı	HATA	Yüksek	EnRoutePlus	5.03	İldeniz Boy maddesidir.
20	1465	Tamamlandı	İSTEK	Çok Acil	EnRoutePlus	5.02	-
19	1132	Tamamlandı	İSTEK	Çok Acil	EnRoutePlus	5.02	LKS2 için sabit muhasebe kodları tanımladığımız ekranda (LKS2 Muhasebe Kodları) Ver
18	1962	Tamamlandı	HATA	Çok Acil	EnRoutePlus	5.02	-
18	3033	Tamamlandı	İSTEK	Çok Acil	EnRoutePlus	5.02	EnRoutePlus Bayi Uygulaması Geliştirme İşlemleri
18	2489	Tamamlandı	HATA	Çok Acil	EnRoutePlus	5.02	Yıldızlar Datça
18	371	Tamamlandı	HATA	Yüksek	EnRoutePlus	5.01	18860
18	4703	Tamamlandı	HATA	Çok Acil	ERPanorama	5.04	Nokta devrinde bir müşteri seçip kaydet dediğimde hata alınıyorum. Eg: \\srvdestek\EF
18	88	Tamamlandı	İSTEK	Çok Acil	EnRoutePlus	5.01	Kontör Ürün (Vending) 1-EnRoutePlus bilgi hazırlamaya gerekli değişikliklerin eklenmes
17	3294	İptal Edildi	HATA	Yüksek	EnRoutePlus	5.03	BECKS M-PACK TAVA 50 CL 8*3 isimli bir ürün var. bu ürün üzerinde düzenle kaydet y
17	3571	Tamamlandı	HATA	Çok Acil	EnRoutePlus	5.02	1204 Nolu IDH kapalı olduğundan yeni IDH açılmaktadır.
17	2309	Tamamlandı	İSTEK	Acil	EnRoutePlus	5.03	Acil öncelikli.
17	6453	Tamamlandı	HATA	Çok Acil	EnRoutePlus	5.04	NOT: İletilen yer distribütör ve ERP 504 pilotu

Figure 6.8 Defect that has most transaction on system

The last two reports are more statistical reports. One of them is weekly defect distribution on projects and their states. This report shows managers general view of the development team’s density on projects by weekly in Figure 6.9.



Yeni Açılan IDH Sayısı	198	Hareket Gören Toplam IDH Sayısı	297	Toplam Transaction Sayısı	919

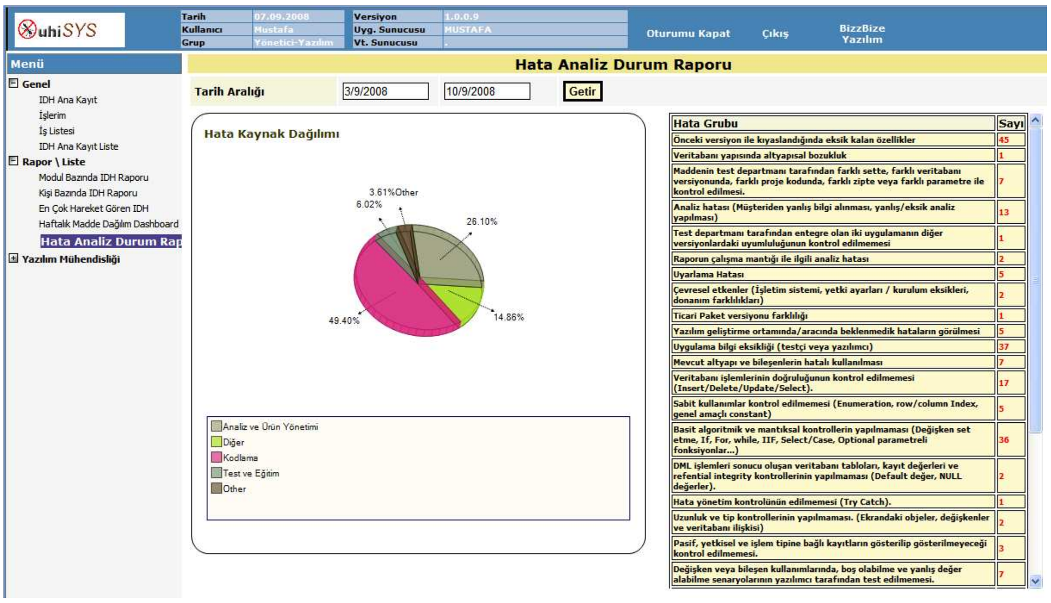
Paket Tipi	Durum	Sayı
	TAMAM	1
ERC	TAMAM	13
ERC	TESTTE DELEGE	2
ERC	YAZILIMDA DELEGE	2
ERM	TAMAM	4
ERM	TESTTE DELEGE	6
ERM	YAZILIMDA DELEGE	2
ERP	DESTEKTE DELEGE	17
ERP	TAMAM	75
ERP	TESTTE DELEGE	19
ERP	YAZILIMDA DELEGE	12
ERPan	TAMAM	13
ERPan	TESTTE DELEGE	2
ERPan	YAZILIMDA DELEGE	7
ERQ	IPTAL	2
ERQ	TAMAM	70
ERQ	TESTTE DELEGE	6

Durum Özet	Sayı Özet
DESTEKTE DELEGE	17
IPTAL	3
TAMAM	201
TESTTE DELEGE	43
YAZILIMDA DELEGE	33

Figure 6.9 Distribution of defects over projects and their last conditions.

This report also has some other information such as new opened defect count, total number of defects that has transactions and total numbers of transactions on the system. The information can be used as a metric for evaluating development team’s performance and project quality.

The last report is maybe the most important report for defect management system that defect analysis report. This report runs on weekly as a default but it can be run with any time interval criteria. It shows us defect groups and number of defects belong that group detailed according to related time interval. We can see the report in Figure 6.10. The real advantages of reports can be seen after evaluated by managers detailed.



Tarih	07.09.2008	Versiyon	1.0.0.9
Kullanıcı	Mustafa	Uyg. Sunucusu	MUSTAFA
Grup	Yönetici-Yazılım	Vt. Sunucusu	.

Hata Analiz Durum Raporu

Tarih Aralığı: 3/9/2008 – 10/9/2008

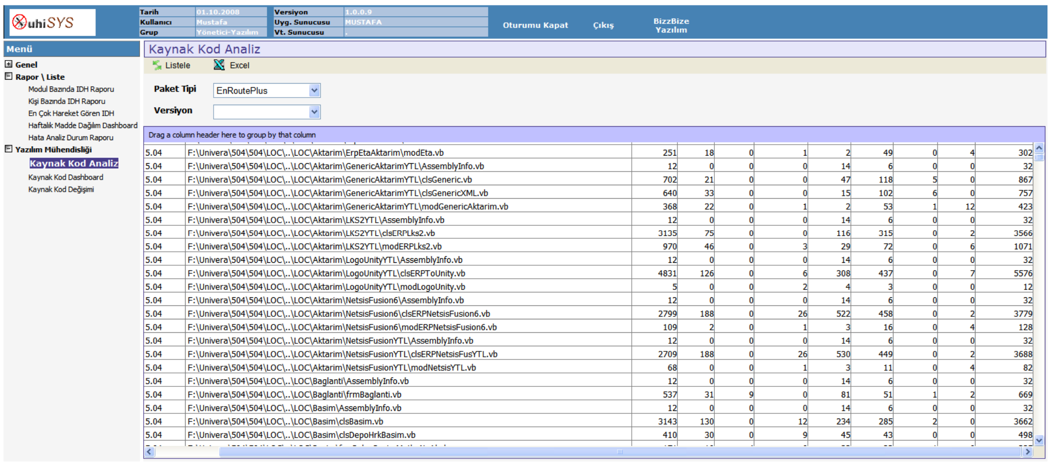
Hata Grubu	Sayı
Önceki versiyon ile kıyaslandığında eksik kalan özellikler	45
Veritabanı yapısında altyapısal bozukluk	1
Maddenin test departmanı tarafından farklı sette, farklı veritabanı versiyonunda, farklı proje kodunda, farklı zipte veya farklı parametre ile kontrol edilmesi.	7
Analiz hatası (Müşteriden yanlış bilgi alınması, yanlış/eksik analiz yapılması)	13
Test departmanı tarafından entegre olan iki uygulamanın diğer versiyonlardaki uyumluluğunun kontrol edilmemesi	1
Raporun çalışma mantığı ile ilgili analiz hatası	2
Uyarlama Hatası	5
Çevresel etkenler (İşletim sistemi, yetki ayarları / kurulum eksikleri, donanım farklılıkları)	2
Ticari Paket versiyonu farklılığı	1
Yazılım geliştirme ortamında/aracında beklenmedik hataların görülmesi	5
Uygulama bilgi eksikliği (testçi veya yazılımcı)	37
Mevcut altyapı ve bileşenlerin hatalı kullanılması	7
Veritabanı işlemlerinin doğruluğunun kontrol edilmemesi (Insert/Delete/Update/Select).	17
Sabit kullanımlar kontrol edilmemesi (Enumeration, row/column Index, genel amaçlı constant)	5
Basit algoritmik ve mantıksal kontrollerin yapılmaması (Değişken set etme, If, For, while, IIF, Select/Case, Optional parametreli fonksiyonlar...)	36
DML işlemleri sonucu oluşan veritabanı tabloları, kayıt değerleri ve refential integrity kontrollerinin yapılmaması (Default değer, NULL değerler).	2
Hata yönetim kontrolünün edilmemesi (Try Catch).	1
Uzunluk ve tip kontrollerinin yapılmaması. (Ekrandaki objeler, değişkenler ve veritabanı ilişkisi)	2
Pasif, yetkisel ve işlem tipine bağlı kayıtların gösterilip gösterilmeyeceği kontrol edilmemesi.	3
Değişken veya bileşen kullanımlarında, boş olabilme ve yanlış değer alabilme senaryolarının yazılımcı tarafından test edilmemesi.	7

Figure 6.10 Defect analysis report

Report part of the UHISYS defect management system can be very useful for with these reports that are introduced. The last part of the system is software engineering part. The first item of this part is source code analysis part. Line analysis of the all code files physically linked to project that is selected on criteria. UHISYS needs to another small project to identify projects for line analysis. Actually, some of the projects are developing different development platforms and versions. This small project searches all of the files detailed according to their platform and version. It returns some important information.

- Line of executable code
- Function count
- Event count
- Public variable count
- Comment line count
- Empty line count
- Code Region count
- Public enumeration count
- Total line of code

These values returned from that project and listed in Figure 6.11 and we can see the different version’s results on source code dashboard on Figure 6.12.



Tarih	01.10.2008	Versiyon	1.0.0.9
Kullanıcı	Mustafa	Uyg. Sunucusu	MUSTAFA
Grup	Yönetici-Yazılım	Vt. Sunucusu	.

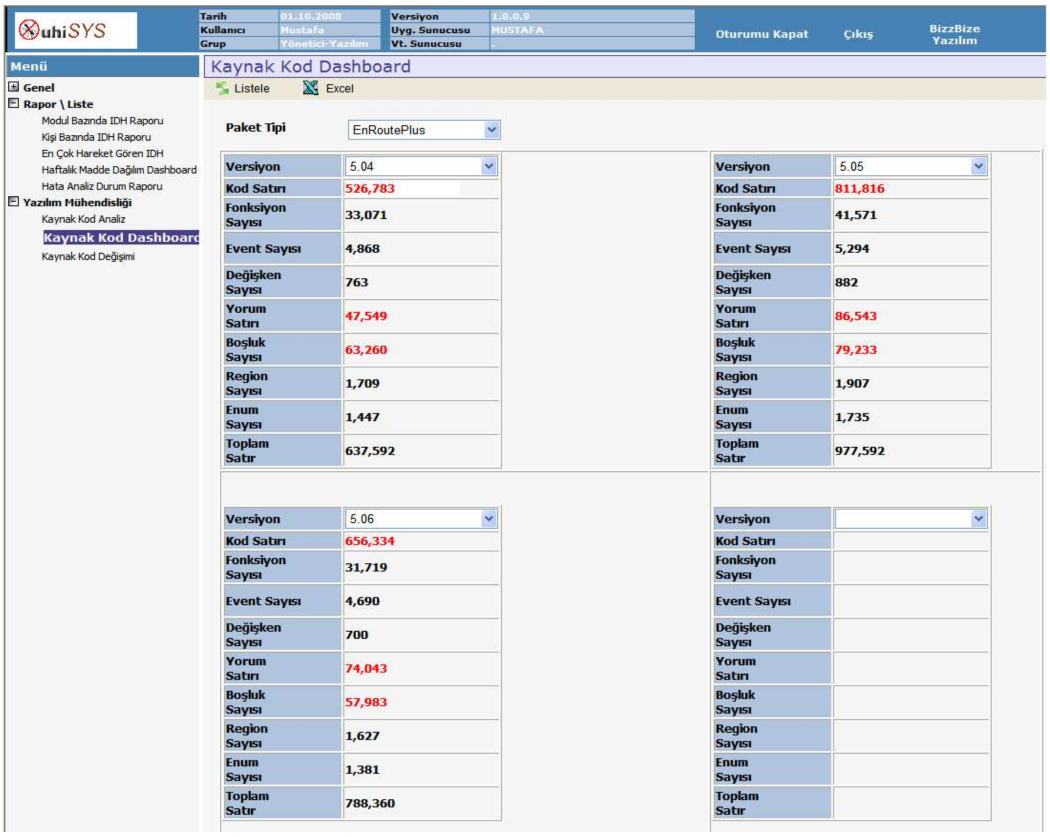
Kaynak Kod Analiz

Paket Tipi: EnRoutePlus

Versiyon:

5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\ErpEtaAktarim\modEta.vb	251	18	0	1	2	49	0	4	302
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\GenericAktarimYTL\AssemblyInfo.vb	12	0	0	0	14	6	0	0	32
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\GenericAktarimYTL\clsGeneric.vb	702	21	0	0	47	118	5	0	867
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\GenericAktarimYTL\clsGenericXML.vb	640	33	0	0	15	102	6	0	757
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\GenericAktarimYTL\modGenericAktarim.vb	368	22	0	1	2	53	1	12	423
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\LKS2YTL\AssemblyInfo.vb	12	0	0	0	14	6	0	0	32
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\LKS2YTL\clsERPLks2.vb	3135	75	0	0	116	315	0	2	3566
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\LKS2YTL\modERPLks2.vb	970	46	0	3	29	72	0	6	1071
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\LogoUnityYTL\AssemblyInfo.vb	12	0	0	0	14	6	0	0	32
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\LogoUnityYTL\clsERPToUnity.vb	4831	126	0	6	308	437	0	7	5576
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\LogoUnityYTL\modLogoUnity.vb	5	0	0	2	4	3	0	0	12
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\NetsisFusion6\AssemblyInfo.vb	12	0	0	0	14	6	0	0	32
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\NetsisFusion6\clsERPNetsisFusion6.vb	2799	188	0	26	522	458	0	2	3779
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\NetsisFusion6\modERPNetsisFusion6.vb	109	2	0	1	3	16	0	4	128
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\NetsisFusionYTL\AssemblyInfo.vb	12	0	0	0	14	6	0	0	32
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\NetsisFusionYTL\clsERPNetsisFusYTL.vb	2709	188	0	26	530	449	0	2	3688
5.04	F:\Univera\504\504\LOC\..\LOC\Aktarim\NetsisFusionYTL\modNetsisYTL.vb	68	0	0	1	3	11	0	4	82
5.04	F:\Univera\504\504\LOC\..\LOC\Baglanti\AssemblyInfo.vb	12	0	0	0	14	6	0	0	32
5.04	F:\Univera\504\504\LOC\..\LOC\Baglanti\frmBaglanti.vb	537	31	9	0	81	51	1	2	669
5.04	F:\Univera\504\504\LOC\..\LOC\Basim\AssemblyInfo.vb	12	0	0	0	14	6	0	0	32
5.04	F:\Univera\504\504\LOC\..\LOC\Basim\clsBasim.vb	3143	130	0	12	234	285	2	0	3662
5.04	F:\Univera\504\504\LOC\..\LOC\Basim\clsDepoHrkBasim.vb	410	30	0	9	45	43	0	0	498

Figure 6.11 Source code analysis of UHISYS



Tarih	01.10.2008	Versiyon	1.0.0.9
Kullanıcı	Mustafa	Uyg. Sunucusu	MUSTAFA
Grup	Yönetici-Yazılım	Vt. Sunucusu	.

Kaynak Kod Dashboard

Paket Tipi: EnRoutePlus

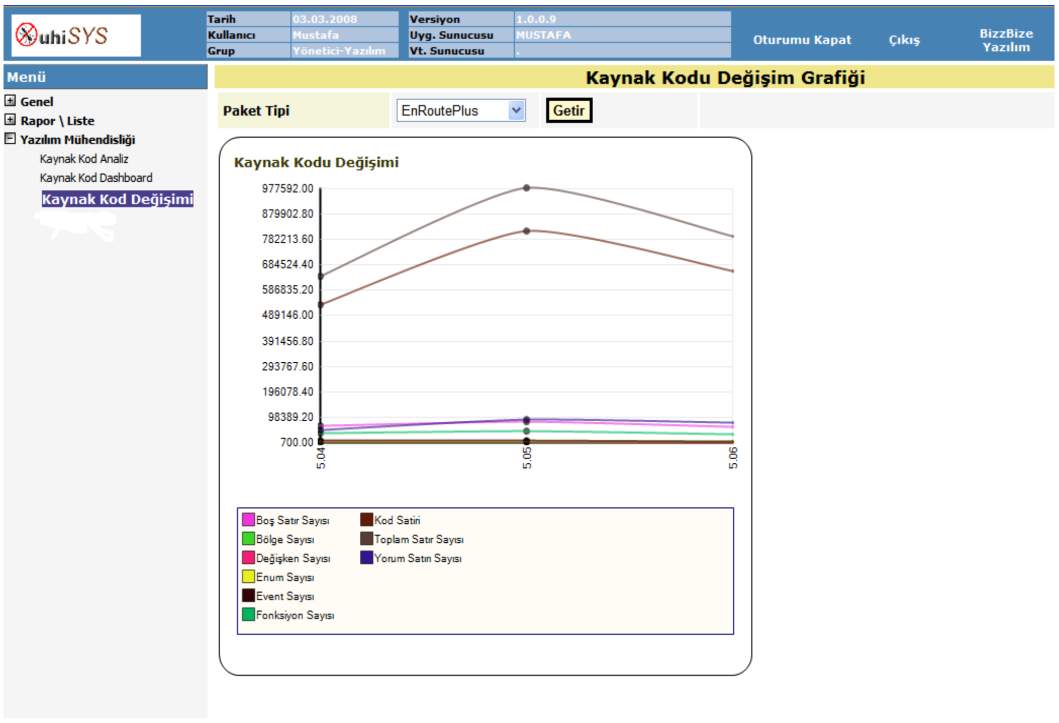
Versiyon	5.04	Versiyon	5.05
Kod Satırı	526,783	Kod Satırı	811,816
Fonksiyon Sayısı	33,071	Fonksiyon Sayısı	41,571
Event Sayısı	4,868	Event Sayısı	5,294
Değişken Sayısı	763	Değişken Sayısı	882
Yorum Satırı	47,549	Yorum Satırı	86,543
Boşluk Sayısı	63,260	Boşluk Sayısı	79,233
Region Sayısı	1,709	Region Sayısı	1,907
Enum Sayısı	1,447	Enum Sayısı	1,735
Toplam Satır	637,592	Toplam Satır	977,592

Versiyon	5.06	Versiyon	
Kod Satırı	656,334	Kod Satırı	
Fonksiyon Sayısı	31,719	Fonksiyon Sayısı	
Event Sayısı	4,690	Event Sayısı	
Değişken Sayısı	700	Değişken Sayısı	
Yorum Satırı	74,043	Yorum Satırı	
Boşluk Sayısı	57,983	Boşluk Sayısı	
Region Sayısı	1,627	Region Sayısı	
Enum Sayısı	1,381	Enum Sayısı	
Toplam Satır	788,360	Toplam Satır	

Figure 6.12 Source code dashboard with comparable results

This report serve us selected projects information about their source code analysis with different tables. Therefore we can compare last version of the project to previous versions. The most important parts of the results are colored such as executable line of code, comment line and empty line.

Executable line of code shows us code functionality. The other important fields are unnecessary code lines that cause projects performance getting worse. The last one of the software engineering part of the system is shows us the difference between project fields with graphical interface like Figure 6.13.



Tarih	03.03.2008	Versiyon	1.0.0.9
Kullanıcı	Mustafa	Uyg. Sunucusu	MUSTAFA
Grup	Yönetici-Yazılım	Vt. Sunucusu	.

Paket Tipi: EnRoutePlus

🟪 Boş Satır Sayısı	🟫 Kod Satiri
🟩 Bölge Sayısı	🟫 Toplam Satır Sayısı
🟥 Değişken Sayısı	🟦 Yorum Satırı Sayısı
🟨 Enum Sayısı	
⬛ Event Sayısı	
🟩 Fonksiyon Sayısı	

Figure 6.13 Source code difference graphic

We introduced graphical user interface of all operational and reporting services of UHISYS defect management system. Working strategy of UHISYS and data collection process will be explained next part of the article.

6.3.4 How System Works and Collects Data?

Before UHISYS, communication between departments can established with mail messages and excel worksheets that includes records and necessary information about records.

This way is not an optimal choice for communication. It is difficult to reach actual document that last updated. If the manager of the any department reaches common excel worksheet that is last updated, he/she must also update his/her own department's worksheet. These operations are complex, time consuming and creates very important network traffics on servers.

UHISYS provides more powerful communication between departments and users. If a defect record delegated by user to another without changing defect's state to fixed or canceled, all of the transactions are also message between users. Users can delete their mail messages but they cannot delete defect or request transaction records. Therefore, it is impossible to lost information about a defect on system.

At first, UHISYS defect management system was started to use with default configuration. There was no valuable information about carried on records of defects or requests. Actually, it was just defect and request monitoring system. The first purpose of the system is providing better communications between departments, reduce complexity of process and time to spend manage these processes. In addition to this, e-mail notification about a new record send to related managers and related user automatically to inform him/her.

After UHISYS started to use, the managers of these departments can see the last states of the related defects or request records. There is not possible to create a defect or request record without managers' knowledge. Furthermore they can plan jobs in order and assign to people to do it.

Two months later, a new idea was rose such as improving UHISYS to get new benefits to increase its advantages. This decision makes an important milestone on defect prevention or reducing number of defects on projects.

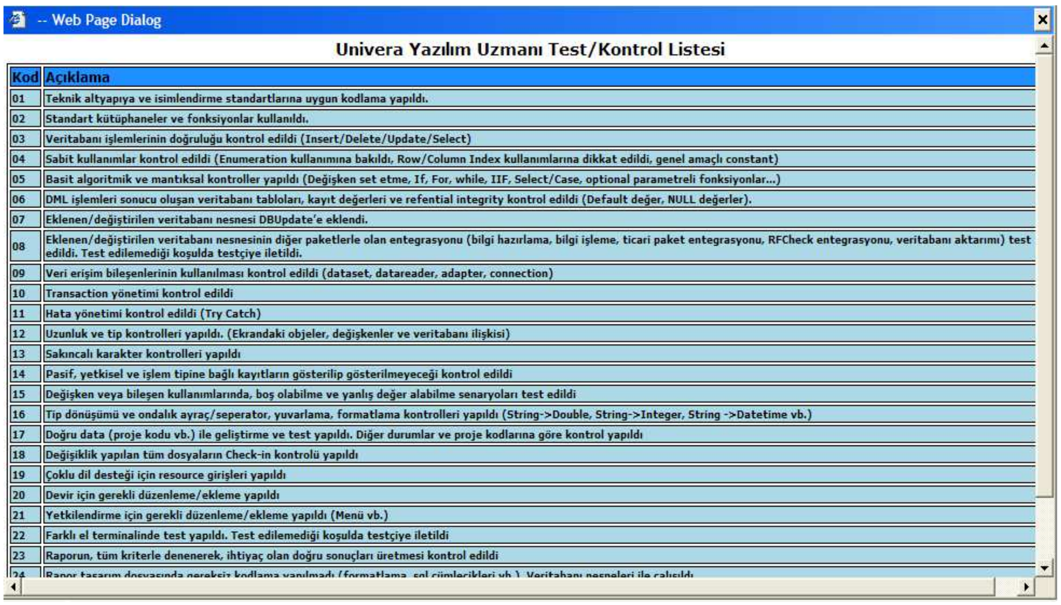
Developers noted general information about defects such as real condition of the defects, why this defect occur, how defect fix and suggestion to prevent this defect. These are very important and valuable information on quality of development process. Project technical managers collect all information from each developer to analyze. They categorize the reasons that cause defects and also what developers may control while coding the projects.

Actually, a checklist was generated according to the managers' analysis for developer to control some coding routines before integrating a request to the project. These can be called "coding procedures" or "coding standards". This standard includes some coding habits in development in company and general standards of software development. There is software developer checklist as coding standards in Figure 6.14. If the developer codes the requirement according to these standards, he/she provides more performed and reusable system with minimum number of defects. It is also provides more understandable code for other developers. As we know, software development sector force developers and programmers to catch new technology and learn better coding techniques that have better performance.

Some of the important standards and checklist that must be controlled by a developer while both defect clearing and integrating a requirement is listed below.

- Conform coding standards and according to technical infrastructure.
- Using standard libraries and common functions.
- Checking database operations results and be sure of their correctness.
- Checking transaction management.
- Exception management code (Try Catch)
- Check variables across to be null or nothing.
- Check integration with other projects.

- Check performance test with a data that has huge records.
- Check resource files to support multi language
- Check authorization of user access.



Univera Yazılım Uzmanı Test/Kontrol Listesi

Kod	Açıklama
01	Teknik altyapıya ve isimlendirme standartlarına uygun kodlama yapıldı.
02	Standart kütüphaneler ve fonksiyonlar kullanıldı.
03	Veritabanı işlemlerinin doğruluğu kontrol edildi (Insert/Delete/Update/Select)
04	Sabit kullanımlar kontrol edildi (Enumeration kullanımına bakıldı, Row/Column Index kullanımlarına dikkat edildi, genel amaçlı constant)
05	Basit algoritmik ve mantıksal kontroller yapıldı (Değişken set etme, If, For, while, IIF, Select/Case, optional parametreli fonksiyonlar...)
06	DML işlemleri sonucu oluşan veritabanı tabloları, kayıt değerleri ve refential integrity kontrol edildi (Default değer, NULL değerler).
07	Eklenen/değiştirilen veritabanı nesnesi DBUpdate'e eklendi.
08	Eklenen/değiştirilen veritabanı nesnesinin diğer paketlerle olan entegrasyonu (bilgi hazırlama, bilgi işleme, ticari paket entegrasyonu, RFCheck entegrasyonu, veritabanı aktarımı) test edildi. Test edilemediği koşulda testçiye iletildi.
09	Veri erişim bileşenlerinin kullanılması kontrol edildi (dataset, datareader, adapter, connection)
10	Transaction yönetimi kontrol edildi
11	Hata yönetimi kontrol edildi (Try Catch)
12	Uzunluk ve tip kontrolleri yapıldı. (Ekrandaki objeler, değişkenler ve veritabanı ilişkisi)
13	Sakıncalı karakter kontrolleri yapıldı
14	Pasif, yetkisel ve işlem tipine bağlı kayıtların gösterilip gösterilmeyeceği kontrol edildi
15	Değişken veya bileşen kullanımlarında, boş olabilme ve yanlış değer alabilme senaryoları test edildi
16	Tip dönüşümü ve ondalık ayraç/seperator, yuvarlama, formatlama kontrolleri yapıldı (String->Double, String->Integer, String ->Datetime vb.)
17	Doğru data (proje kodu vb.) ile geliştirme ve test yapıldı. Diğer durumlar ve proje kodlarına göre kontrol yapıldı
18	Değişiklik yapılan tüm dosyaların Check-in kontrolü yapıldı
19	Çoklu dil desteği için resource girişleri yapıldı
20	Devir için gerekli düzenleme/ekleme yapıldı
21	Yetkilendirme için gerekli düzenleme/ekleme yapıldı (Menü vb.)
22	Farklı el terminalinde test yapıldı. Test edilemediği koşulda testçiye iletildi
23	Raporun, tüm kriterle denenerek, ihtiyaç olan doğru sonuçları üretmesi kontrol edildi
24	Rapor tasarım dosyasında gereksiz kodlama yapılmadı (formatlama, sql cümlecikleri vb.). Veritabanı nesneleri ile çalışıldı

Figure 6.14 Developer coding standards and checklist

The other part of the defect preventing milestones is defect reason classification. Defect reason classification is very important like developer checklist. There can be many defects on projects before developers start to obey checklist rules. It is known fact, all of developers and programmers are human beings and we cannot prevent them not to do any mistake. Therefore, a defect can be occurs after start to use this checklist. This classification helps us to categorize defects every time.

In software development process, all software companies can face to defects all the time but the main purpose of the system must decrease defect density on projects. The main purpose of defect monitoring and preventing systems is to get fewer defects on projects. Now, this defect classification system on UHISYS supplies us to analyze actual reasons that cause a defect.

In generally, if you know the real sources of problems of the current system, you can manage the developing process to eliminate these problems before they cause a defect. These problems can be anything such as weakness of technical infrastructure, coding habits, developers' technical skills and software development tool's performance. Developers have very important role to identify the actual problem of the related defect that fixed. Developers must select a defect reason in Figure 6.15 while delegating the defect record to test department with changing record's state to "Fixed".

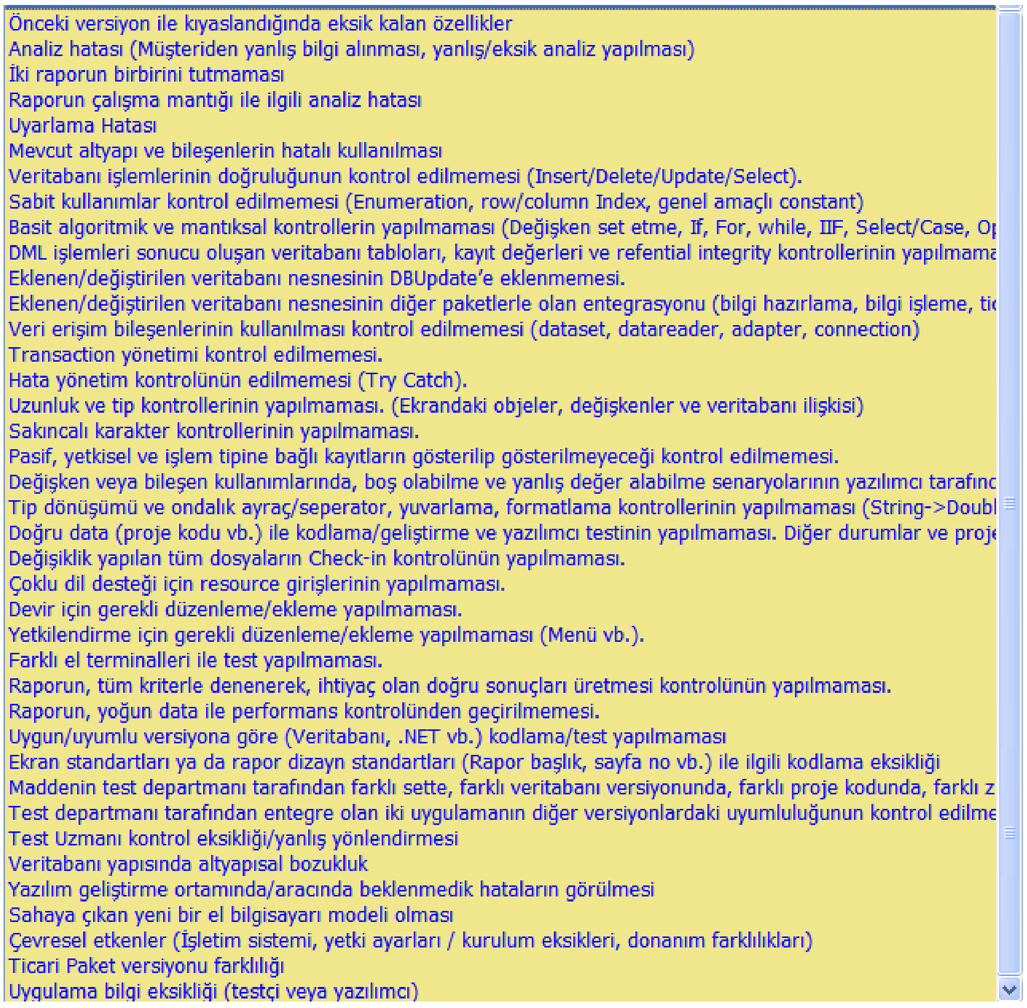

Önceki versiyon ile kıyaslandığında eksik kalan özellikler
Analiz hatası (Müşteriden yanlış bilgi alınması, yanlış/eksik analiz yapılması)
İki raporun birbirini tutmaması
Raporun çalışma mantığı ile ilgili analiz hatası
Uyarlama Hatası
Mevcut altyapı ve bileşenlerin hatalı kullanılması
Veritabanı işlemlerinin doğruluğunun kontrol edilmemesi (Insert/Delete/Update/Select).
Sabit kullanımlar kontrol edilmemesi (Enumeration, row/column Index, genel amaçlı constant)
Basit algoritmik ve mantıksal kontrollerin yapılmaması (Değişken set etme, If, For, while, IIF, Select/Case, O
DML işlemleri sonucu oluşan veritabanı tabloları, kayıt değerleri ve refential integrity kontrollerinin yapılmama
Eklenen/değiştirilen veritabanı nesnesinin DBUpdate'e eklenmemesi.
Eklenen/değiştirilen veritabanı nesnesinin diğer paketlerle olan entegrasyonu (bilgi hazırlama, bilgi işleme, ti
Veri erişim bileşenlerinin kullanılması kontrol edilmemesi (dataset, datareader, adapter, connection)
Transaction yönetimi kontrol edilmemesi.
Hata yönetim kontrolünün edilmemesi (Try Catch).
Uzunluk ve tip kontrollerinin yapılmaması. (Ekrandaki objeler, değişkenler ve veritabanı ilişkisi)
Sakıncalı karakter kontrollerinin yapılmaması.
Pasif, yetkisel ve işlem tipine bağlı kayıtların gösterilip gösterilmeyeceği kontrol edilmemesi.
Değişken veya bileşen kullanımlarında, boş olabilme ve yanlış değer alabilme senaryolarının yazılımcı tarafınc
Tip dönüşümü ve ondalık ayraç/seperator, yuvarlama, formatlama kontrollerinin yapılmaması (String->Doubl
Doğru data (proje kodu vb.) ile kodlama/geliştirme ve yazılımcı testinin yapılmaması. Diğer durumlar ve proj
Değişiklik yapılan tüm dosyaların Check-in kontrolünün yapılmaması.
Çoklu dil desteği için resource girişlerinin yapılmaması.
Devir için gerekli düzenleme/ekleme yapılmaması.
Yetkilendirme için gerekli düzenleme/ekleme yapılmaması (Menü vb.).
Farklı el terminalleri ile test yapılmaması.
Raporun, tüm kriterle denenerek, ihtiyaç olan doğru sonuçları üretmesi kontrolünün yapılmaması.
Raporun, yoğun data ile performans kontrolünden geçirilmemesi.
Uygun/uyumlu versiyona göre (Veritabanı, .NET vb.) kodlama/test yapılmaması
Ekran standartları ya da rapor dizayn standartları (Rapor başlık, sayfa no vb.) ile ilgili kodlama eksikliği
Maddenin test departmanı tarafından farklı sette, farklı veritabanı versiyonunda, farklı proje kodunda, farklı z
Test departmanı tarafından entegre olan iki uygulamanın diğer versiyonlardaki uyumluluğunun kontrol edilme
Test Uzmanı kontrol eksikliği/yanlış yönlendirmesi
Veritabanı yapısında altyapısal bozukluk
Yazılım geliştirme ortamında/aracında beklenmedik hataların görülmesi
Sahaya çıkan yeni bir el bilgisayarı modeli olması
Çevresel etkenler (İşletim sistemi, yetki ayarları / kurulum eksikleri, donanım farklılıkları)
Ticari Paket versiyonu farklılığı
Uygulama bilgi eksikliği (testçi veya yazılımcı)

Figure 6.15 Defect reason groups defined by technical managers

Actually, some of these reasons are listed in developer coding standards and checklist. Some of other reasons are related with regional settings of system that application runs, development tool's exceptions and hardware exception.

After the upgraded system that has developers checklist and defect reason classification part, getting used, we can chance to find out development processes' weakness. There were many meetings set with technical managers, project managers and developers to find answers to questions like these;

- What can we do to decrease defect density?
- How can we improve overall system performance?
- How can we develop system with less effort?

There are many different answers to each question according to the defect classifications statistics from UHISYS. Actually, the effects of the defect management system to software development process are related with managers encourage. If they cannot decide to do anything for system improvement, every defect management system works for just defect monitoring system and that only supplies project statistics.

Managers have some very important decisions after discussion about these questions according to data that UHISYS collected. These critical decisions are;

- To get Microsoft Certified Professional Course on SQL Server 2005 for developers.
- Force developers to code on server side to reach better performance
- Upgrade development tool the newest version(.NET 2003 to .NET 2008)
- Use pair programming methodologies on developer integration
- Designing new software development infrastructure for new projects
- Software Re-Engineering on projects that are currently used

All of these decisions are very important for a software company. Developing and coding standards change on different ways step by step. Company will share improvements on development processes and product quality according to decisions.

6.3.5 Effects of Practiced Decisions according to the Defect Management System on Development Process and Quality of Product

Now, it is time to see the effects of defect management systems on development processes and total product quality. All of these decisions were practiced on some parts of current projects that really need to improve and also completely used on new projects on company. The main purposes of these decisions and effects are explained one by one.

6.3.5.1 SQL Server 2005, MCP Certification Course

SQL Server 2005 course is very useful for all developers. Developers learned to create queries that have optimum performance while they running. In addition to this, operations that include lots of database connection moves from the application to database server by using SQL Server's commands. The main purpose of these changes is to increase application performance and decrease system memory that application used. All of the developers of the company pass the exam that organized by Microsoft and get Microsoft Certified Professional ID.

6.3.5.2 Server Side Coding Techniques

After this decisions accepted we practice server side coding on some problematic operation and reports. This approach provides us approximately 85 percent performance gain and less system memory usage while executing these operations.

Therefore we just see the affects of analysis that runs on data collected by defect management systems.

6.3.5.3 Development Environment Upgrade

One of the other decisions is upgrading version of development tools. Microsoft Visual Studio .NET 2008 has many advantages on coding for developers. Coding on new version is easier and quicker than previous version of Microsoft Visual Studio .NET development environment.

6.3.5.4 Pair Programming Technique

“Pair programming is programming done by a pair of programmers. While one of them is creating a software artifact (e.g. a UML diagram or C++ code) the other is continuously assuring quality, i.e. he/she is watching, trying to understand, asking questions, looking for alternative approaches, helping to avoid defects. Two programmers are given one computer and they have to share it. After some time they are switching the roles: creator becomes quality assurer and vice versa.”

We start to use pair programming methodology while orienting a new developer to an existing project. Generally, newer developers have very different sight to the projects and they can suggest new ideas to implement operations with different approaches.

In addition to this, developing a requirement with an experienced developer has many advantages for new developer such as to learn general structure of the project quickly, reduce adaptation period to development team and developing request with less defects.

6.3.5.5 New Software Development Infrastructure

Developing new software development infrastructure is the most important decision according to the analyzed data comes from defect management system. The capabilities of software’s technical infrastructure can measure quality of the product and developing processes. The previous technical structure cannot meet today’s professional coding methodologies. Previous structure developed before software development technology was introduced to effective methodologies on software development.

6.3.5.6 Software Re-Engineering on current projects

“Re-engineering is the examination, analysis and alteration of an existing software system to reconstitute it in a new form, and the subsequent implementation of the new form. The process typically encompasses a combination of other processes such as reverse engineering, re-documentation, restructuring, translation, and forward engineering. The goal is to understand the existing software (specification, design, implementation) and then to re-implement it to improve the system's functionality, performance or implementation. The objective is to maintain the existing functionality and prepare for functionality to be added later.”

The main purpose of the reengineering study is improving overall system performance. The scope of this study is working on coding processes to get more reusable and more understandable code that application runs on. Therefore, there are no changes on specification and design on project, only implementations that really need to improve, changed.

More effects can be seen on current projects that reengineering study has done. These effects are; 21 percent less code than original code and 16 percent better performance while operation running. In addition to this, memory that is used for application reduced. Therefore, analysis of defect management system is right for product performance and quality and reengineering decision is very useful for current project.

6.4 Future Works

UHISYS defect management system can grow some different concepts but a couple of concepts will be more important. One of these ways is total software project management system that includes test case automations and regression analysis.

A requirement analysis can be effect a lot of different part of the system sometimes, actually in large software projects. If developer does not have enough information about business, it cannot develop that requirement consistently. In addition to this, a tester does not have enough information; too, it cannot test all cases. Therefore, UHISYS defect management system can be more powerful with test case automation.

The other important way is using data mining techniques on collected data by UHISYS. Classify defects according to their definitions both using text mining and classification algorithms. It is difficult to collect enough data to implement a data mining algorithm to reach reliable knowledge but if enough data collected, these algorithms produce valuable knowledge.

CHAPTER SEVEN

CONCLUSION

7.1 Conclusion

Software development processes may be seen very complicated. Companies can use the best development tools that the world’s top companies produced with today’s technology. However, it is impossible to produce software without any defect theoretically. Developers are human beings, so they can mistake every time. Therefore, defects are a part of our life while developing software.

Defect management systems are very important systems that help to manage this part of the life. Basically, these systems used for managing and monitoring the defect lifecycle on development process.

In generally, standard defect management systems affect the development process just one way that is monitoring defects on overall development system. Extra properties of the UHISYS can provide managers to think about their systems in many ways. If the manager of the software development team can decide to use new techniques or approaches according to the collected data with UHISYS, development process of software can affected on positive ways. Therefore, it is necessary to have an encouraged software manager that can analyze UHISYS data and practice the result that analyzed,

REFERENCES

Basic Concepts (n.d) Retrieved July 26, 2008 from http://www.asq.org/glossary/q.html

BUGTrack Products-Overview Retrieved August 23, 2008 from http://www.bugtrack.net/index.htm

Humphrey, W. S, *Acquiring quality software*, The Journal of Defense Software Engineering, December, 2005

Humphrey,W. S, *The software quality profile*, (n.d) Retrieved October 01, 2008 from http://www.sei.cmu.edu/publications/articles/quality-profile/index.html

K. Beck, *Extreme programming explained: Embrace change*, Addison Wesley, Boston. 2000.

Methodology-quality? (March 12, 2004) Retrieved July 27, 2008 from http://www.statistics.gov.uk/about/data/methodology/quality/projects/what_is_quality.asp

Quality Management,(n.d) Retrieved October 01, 2008 from http://en.wikipedia.org/wiki/Quality_management

Robert J. Kosman, *A Two Step methodology to reduce requirement defects*, Annals of engineering 3, 1997

Rosenberg L. H., *Software Re-engineering*, Software Assurance Technology Center Unisys Federal Systems 301-286-0087