DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

SOFTWARE ENGINEERING PRACTICES ON MOBILE APPLICATIONS

by

Erdem ARIKAN

April, 2013

İZMİR

SOFTWARE ENGINEERING PRACTICES ON MOBILE APPLICATIONS

A Thesis Submitted to the

Graduate School of Natural and Applied Sciences of Dokuz Eylül University

In Partial Fulfillment of the Requirements for the Degree of Master of

Science in Computer Engineering, Computer Engineering Program

by

Erdem ARIKAN

April, 2013

İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **SOFTWARE ENGINEERING PRACTICES ON MOBILE APPLICATIONS** completed by **ERDEM ARIKAN** under supervision of **ASST.PROF. 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.

Asst. Prof. Dr. Kökten Ulaş BİRANT

Supervisor

Prof.Dr. Yalçın ÇEBİ

(Jury Member)

Yrd.Doç. Dr. Özgür Tamer

(Jury Member)

Prof.Dr. Ayşe OKUR

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I would like to thank to my thesis advisor Asst.Prof. Dr. Kökten Ulaş BİRANT for his help, suggestions and guidance.

I also thank to my family and my sincere friends for their patience and support.

Erdem ARIKAN

SOFTWARE ENGINEERING PRACTICES ON MOBILE APPLICATIONS

ABSTRACT

This research aims to the improvement of Mobile Software Engineering approaches. For this purpose the characteristic and types of mobile devices are examined. And then the limitations caused by their features and prominent requirements are expressed for Mobile Software Engineering. Besides, characteristic of mobile operating systems which are used commonly for mobile devices are described. Existing traditional software engineering methodologies are explained. Afterwards the studies which are related to the Mobile Software Engineering literature are examined. Through the obtained experiences we introduce a mobile methodology. This methodology has enough simplicity to be used in mobile applications which will be developed in mobile devices. In addition, the states and basic principles of Mobile Software Engineering which will be applied in the introduced methodology are presented in a detailed way. Finally an appropriate approach is presented for Mobile Software Engineering.

Keywords: Mobile software engineering, mobile devices, mobile applications, mobile software engineering methodology

SOFTWARE ENGINEERING PRACTICES ON MOBILE APPLICATIONS

ÖZ

Bu çalışma, Mobil Yazılım Mühendisliği yaklaşımlarının iyileştirmesini amaçlıyor. Bu amaç için ilk olarak mobil cihazların türleri ve özellikleri incelendi. Sonrasında özelliklerinin getirdiği kısıtlar ve ön plana çıkan ihtiyaçlar Mobil Yazılım Mühendisliği için açıklandı. Ayrıca mobil cihazlar için yaygın olarak kullanılan mobil işletim sistemlerinin özellikleride anlatıldı ve var olan geleneksel yazılım mühendisliği metodolojileri açıklandı. Ardından literatürde yayınlanan Mobil Yazılım Mühendisliğine yönelik çalışmalar incelendi. Elde edilen birikimle bizim ortaya sunduğumuz bir mobil metodoloji ortaya çıkarıldı. Bu metodoloji tamamen mobil cihazlara geliştirilecek olan mobil uygulamalarda kullanılacak sadeliktedir. Ayrıca sunulan metodolojide uygulanacak olan Mobil Yazılım Mühendisliğinin basamakları ve temel ilkeleri ayrıntılı olarak anlatıldı. Sonuçta Mobil Yazılım Mühendisliğine uygun bir yaklaşım ortaya kondu.

Anahtar sözcükler: Mobil yazılım mühendisliği, mobil cihazlar, mobil uygulamalar, mobil yazılım mühendisliği metodolojisi

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

Today the number of mobile devices rapidly increasing. The prime example of this is smartphones. They become indispensable to our daily lives. Many mobile devices such as this make easier our lives and quickly become a part of it. About 22 years ago, we used to use cassette players for music, VHS tapes for video, film cameras in a variety of length of mm for photograph and audio cassette voice recorders for voice recording. In terms of mobility these devices does not reflect the present needs and outdated. Due to the technology we have reached today we able to perform activities such as taking photos, listening music and recording video in a single mobile device via smartphones, tablet PCs, laptops, and personal multimedia products.

Mobile devices are increasingly becoming smaller and equipping with a variety of sensors. This dragged them to the more complex structures on the basis of software. Early mobile devices used to use simple software. However, the software engineering requirements also has come to fore for mobile devices as a result of their progress. Because of its characteristic it has reached to a complex structure on the software aspect.

Various software development platform, mobile device operating systems and hardware diversity of all kinds are available in the world of mobile software. However, due to the market concern of today product-centric softwares are being developed. Mobile softwares are being written in a sloppy and disorganized way only with the market concern against fast-growing technology. Therefore, the applied software engineering are provided with non-privatized methods for mobile software. For this purpose, the existing procedures and methods of software engineering are used directly.

Mobile software engineering must have a unique structure. Because a quick approach is needed which is appropriate for the the fast-growing mobile structure. Although a variety of mobile software engineering approaches that have been released the details are not stated. In a software engineering approach which will be used for mobile devices a comprehensive mobile standard should be ensured from basic phases to maintenance and support service. Mobile software engineering must be modular against every aspect of developing mobile applications.

In this thesis, we first present mobile devices. Then, we analyse traditional software engineering development methodologies with advantages and disadvantages. We also investigate mobile divices features and OS variations in order to understand mobile software limits. Besides, released surveys and market researches are pointed out for operating systems with mobile user’s era. In this way, we introduce a Mobile Software Engineering Methodologhy and explain Mobile Software Engineering approach principles in developement phases.

CHAPTER TWO
MOBILE DEVICES

Mobile devices are handheld electronics, which have touchscreen and/or small keyboard integrated. And as for personal and daily usage, mobile devices are helpful and practical in systems like server and client computers which are in need of communication for long distances. Especially PDAs and Smartphones are the best choices for these type of jobs. Besides of these, Smartphones can view media files, make video or audio calls.

So in short; if we're talking about mobile devices, they are personal, portable, have easy usage and have some type of network connection.

2.1 Netbook

Netbooks are some kind of small sized and inexpensive notebook computers. They are similar to notebooks in base. But their graphic and ROM hardware capabilities are restricted. Many of these products are using Intel's ATOM CPU technology. Because it has its own graphics and memory integration on the chip. Also Netbooks ROMs (DVD etc.) devices are removed and by this, Netbooks battery life has increased.

Typical Netbook features are wireless connection, 5” to 12” screen size, high capacity hard disk, 1 to 4 GB RAM and limited edition operating system. But also remember that, you can't use a netbook while walking around (Firtman, Mobile Device Categories, 2010).

2.2 Smartphone

Smartphones are some kind of devices that combines a mobile phone and mobile technology (3G, Their best features are; multitasking operating systems, high resolution cameras with video recording capability, TV-Out support, Web viewing capabilities and 3D video acceleration for the new types of WiFi, touch support, video formats and games etc.).

Hardware specifications are; 0.6 to 1.5GHz CPU speed, 1.5" to 3.7" screen size, 1 to 64GB storage capacity, 3G and WiFi network interface and 12 to 168Hrs of usage time. A smartphone usually allows the user to install and run more advanced applications and its operating systems allows to extend software developers applications to extend their limits (Firtman, Mobile Device Categories, 2010).

2.3 Palm

These devices are similar to smartphones but their limited functions (or canalized applications) makes them useful for warehouses, delivery firms, hospitals (for patient records), restaurants (for order and receipt tracking), maintenance checks and security measurements.

In Palm systems, user can only update operating system for software compability. Touchscreen/pointer and WiFi technology indicates its usability still with which mentioned specifications above (Firtman, Mobile Device Categories, 2010).

2.4 Mobile Phone

Mobile pocket phones are devices that only have SMS and calling features. Modern mobile pocket phones supports many instruments such as gaming, bluetooth, media player, MMS messaging, video camera capabilities. With its technologies advancing, these features evolves in a speedy way. Some companies still produces low-end products that provides easy usage for basic users. But as for users who needs web contents, this era will stop its continuation.

Low-end mobile devices have basic web browsers that can only use Web 2.0 and WAP services. They also have basic video camera, media player, memory capabilities and none of them have touchscreen technology integrated.

As for Mid-end mobile phone devices, technology advances for user needs. Because of this, Mid-end phones have integrated basic HTML support, color screen, camera that above average, games, an early version of 3G connection compability and medium range application support. Mid-end mobile phone operating systems and their application support is too limited. Their native applications are device based and users can't change or install any of them after.

And for High-end mobile devices, we can say the same things as smartphones. They have high mobile technologies for its users comfort and joy (Firtman, Mobile Device Categories, 2010).

2.5 Tablet Computer

Tablet computer devices are focusing touchscreen or digital pen pointer technologies rather than physical keyboard. Their typical feature is for being slim. Its difference from notebooks, its built-in keyboard that stays on screen. A tablet PC has powerfull hardware. Various mobile OS is used on tablets (Firtman, Mobile Device Categories, 2010).

2.6 Mobile Device Operating Systems

Mobile operating systems are mobile versions of standart operating systems that we are using today. For example Android goes for Linux and Windows Mobile/CE systems. These operating systems are manufactured for the hardware that are using in mobile devices. There are to many mobile operating systems on the market today. iOS, Android, Windows Mobile, Symbian and Bada are the effective ones now. Some of these operating systems are open source, licenced or propietary. Open sourced systems (like Android) are ready to create homebrew applications to use. Licenced ones are (like Windows Mobile) the hardest ones for the mobile users. Users have to pay money to install applications to their smartphones. And as for the propietary operating systems like iOS or BlackberryOS, operating software development can only be approved by the device manufacturer. None of the users can change the manufacturer design. So without operating systems, mobile devices can only perform basic (text messaging, calling etc.) functions. With operating systems, device usability has increased (Salmre, 2005) (Fling, The Mobile Ecosystem, 2009).

Apple's iPhone and the other Apple based variants have the most competitive user quantity versus other mobile operating systems today. It's a licenced OS but the device capacity and OS's quality makes it the hardest side of the mobile OS war. By the past 7 years, Apple used it's programming and hardware knowledge to create a powerful and user friendly OS. iOS is stable, easy to learn and also easy to create applications for different needs. In the mobile era, Apple started the Touch-And-Use thinking and so others went after it. iPhone devices are so powerful that they can use a mini version of Mac OS X which based on Unix. With this ability, developers can create applications that looks and can be use like desktop computers but with more simple attributes. So if you have basic tools for code writing and editing, you got the environment for iOS sofware development. Although, iPad and iPhone OS's are similar to Mac OS X, you can't design what you think without limitation. You have to focus touch ability and user needs on your applications.

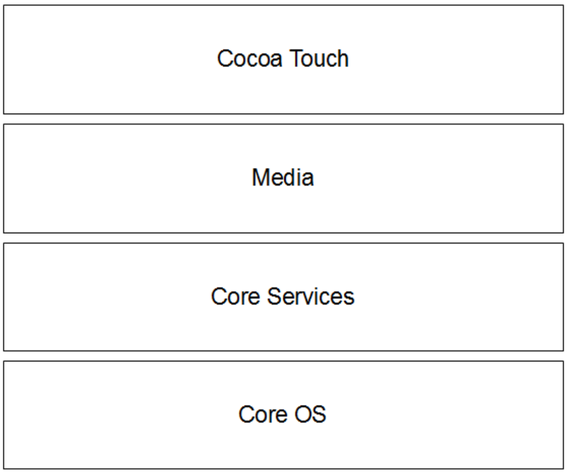


Figure 2.1 iOS basic kernel

iOS have the same basic kernel that used in Mach operating system (Figure 2.1). The first layer on graphic is the kernel. On top of this layer, there are the core services that the OS needs to run applications that user trying to execute. After this layer, there is media layer which are the application frameworks that responds user. And on the top layer, there is Cocoa Touch which is the application interface that you need to create native applications for the iPhone and iPad. On this layer, all software must be approved by Apple before use.

On iOS, if you want to create basic applications, you need to use Xcode. But as for powerful ones, you have to read documentation that you can find in the iOS Dev Center. And if you want to work on basic framework attitude with Objective-C language that used in Cocoa Touch, you can find the needed documentation on Cocoa Fundementals Guide. The designed application can be tested with iOS simulators. For this manner, you need to use The Shark application. This instrument collects and shows the system level events like system messages, interrupts, virtual errors and memory faults. And you can see the responds of your applications behaviour with the iOS on The Shark. It helps to find the bottlenecks on your code. The other side, you can track down the producted errors in your application, check your performance analysis, automaticly test and stress-check your code, perform a system level error check and understand the process of your code work (Wagner, 2008) (Apple, 2013).

Android is the next big thing for users and software developers. It's the new trend and the new boy of the market. Android's SDK released in one year and created a profitable market for all. It's based on Linux kernel, so basicly, Android is the Mobile devices free zone. This zone created by Google Inc. with the Open Handset Alliance and they're still the backbone. After Google Inc., many software and hardware manufacturers like Intel, LG, HTC, ARM, Samsung etc. backed it. As for today, many manufacturers like Sony Ericsson, Samsung or Motorola releases devices with Android OS. And as for future, Dell and Kyocera will release Android devices. With the first release, Android versions grow bigger and as now, the final released version is the 4.x which named as Jelly Bean.

All users can upgrade, change, edit or fix it. Also all software developers can sell what they created for profit. Because it comes basic to advanced programming stack, so all profit goes to developers. Android uses more less power than other mobile operating systems and uses less memory then others. It's visual quality is also better than the others. Because it uses the OpenGL interface. And it can work with all Java codes because of built-in Android Dalvik Java Virtual Machine interface. All desktop systems can execute Java without error but mobile operating systems need upgrade for new versions. But in Android's Dalvik, system creates a virtual environment for will be run application and executes it without error. Android architecture has four layer and a lot of components that makes the operating system open source (Figure 2.2).

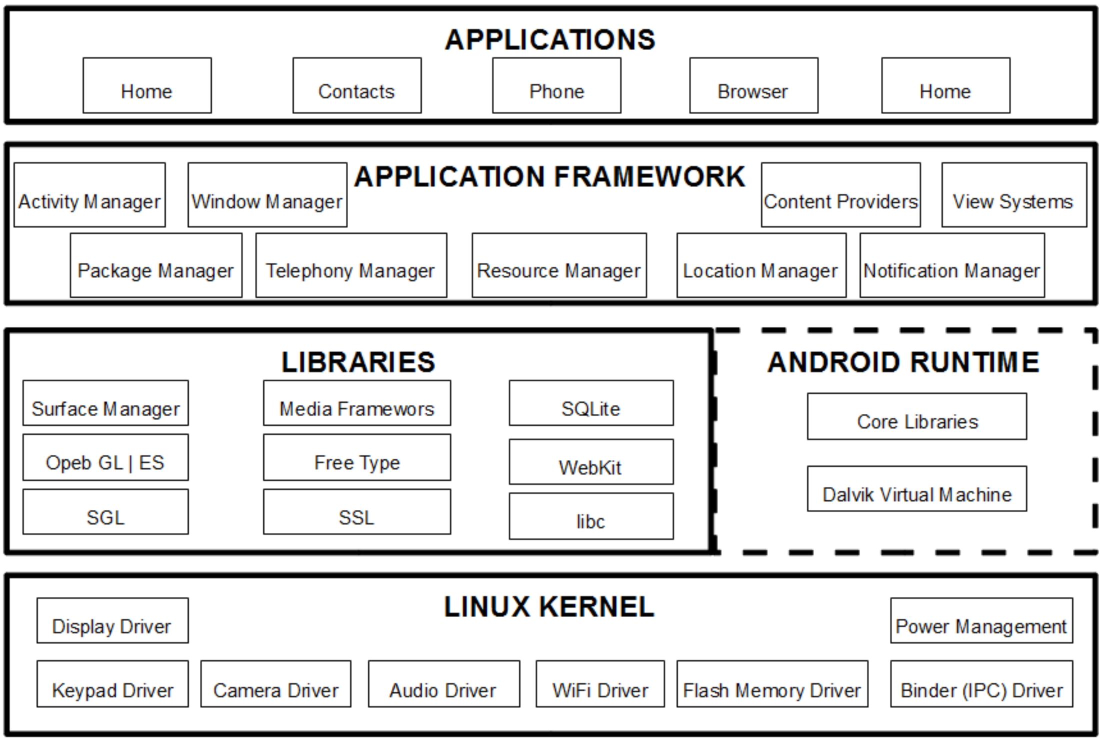


Figure 2.2 Android OS basic kernel

The base layer is the Linux kernel. It handles the memory management, processes, network sub-system and drivers needed for the mobile device. It also acts like a wall that divides hardware from software piles. Above the kernel, there is the libraries segment. In this layer, all librares coded with C or C++ within libc. In this layer, the surface manager works like the Compiz Fusion as the screen manager and creates bitmaps to create efects to show to the user, but in a such simplier way. Inthe same level, there is the Android Runtime section. In this section, there is the Dalvik Machine and the Java Core Libraries. When you run an application, Dalvik creates a new user layer and processes it there to use memory more efficiently. Before the user level, there is the Application Framework. It includes the components that shown on the figure

2.2. The most effective sub-system on this layer is the activity manager. It records and stacks the opened user applications and lets user to switch between them as they wish. The Content Provider keeps the user contents that are needed by the API. The Resource Manager is needed by the software to appoint the resources. The Location Manager uses the base stations to show the user where you are. And the Notification Manager is the layer that shows messages, alerts, appointments to the user. The top layer is the application layer that includes applications and widgets. When you buy an Android device, it comes with a bunch of preinstalled software like browser, e-mail handler, contact lister etc. If you want more, you can download new ones from the Android Market.

As for programming in Android, you have two ways. One is Java and the other one is the XML. In Java, you have to code the frames, buttons etc. to design the interfaces. But in XML, you just say what you want. Because XML is a HTML based markup language, codes can answer almost every question that system needs an answer. These two have most likely the same answer but XML is easier to understand and code than Java (Burnette, 2010) (Android, 2013).

Other operating systems, for example Symbian has the largest OS share on the market but because of its stopped development and only Nokia's continuous support can't stop its falling. Also as for today, Nokia stopped Symbian development in Symbian Inc. and now they're continuing with MeeGo and Maemo. Symbian has versions 1, 2 and 3. While they say there will be a fourth edition, Nokia quit announces about its development.

On third quarter at 2010 ,RIM has the %14.8 of the market share. It's main focus is easy usage and it's design for business. For market needs, it redesigned for multimedia additions after few releases. RIM is a propietary OS that needs digital sign for applications. This way, you can learn the application origin but quality or code security can't be measured. Blackberry OS RIM has 1.0 to 10 (released on 2011) versions and with final release, it reached Harman International agreement with this RIM acquired ONX Software Systems. With this agreement , The Blackberry OS considered to enchance user experience between smartphones, in vehicle audio and infotainment system.

Windows Mobile has %2.8 of the market share on third quarter on 2010. It's losed source and formerIy known as Windows CE. Windows for PDAs and PocketPCs, created a windows based mobile phone revolution for Microsoft. For today, is not receiving a good share from users but as for business world, you can see, It still has some power to show.

After releasing a very intuitive Touch UI with Windows Mobile 6.5, it gained some point from end users; especially on Asia. Unlike iPHONE OS, Windows Mobile 6 (professional and standart) is not optimized for touch input but still needs stylus. On final version (6.5) based on Windows CE 5.2 kernel has basic dekstop Windows features, have more features and look more aesthetic. And with this release, Microsoft opened ' Windows market place for mobile' and put third party additions to it. Windows Mobile 7 (code name photon) scrapped from microsoft plans and they renamed it to Windows Phone 7. But Microsoft released a new version OS which name is Windows Mobile 8 and its development still continues.

It's the next generation for Palm OS. And it's development continues by Palm INC. Access Linux Platform (ALP). And it was planned to be launched at first half of 2007. If this was done the OS will use technicaI specifications from the linux phone standarts forum. But this plan didn't go well and the company can stay in the makets so in 2010 HP acquired PALM and promised a big evolition for web OS so we can expect PALM Web OS on HP netbooks, tablets and mobile phones on future.

Bada is a mobiIe operating system that being deveIoped by Samsung Electronics and Samsung claims that Bada will replace their phone platforms on near future and the convert phones to smart phones. The first device that uses BADA platform named as 'Wave' and with it is release Samsung opened it is mobile market called Samsung Apps. Samsung says Bada is not a smartphone platform but a kernel configurable platform architecture which allows real time operating system kerneI (RTOS), or linux kernel. Samsune has different devices that uses native, Symbian or Windows OS at the end of 2009. But after launch of BADA this operating systems started the use in low and mid-end devices. With BADA, Samsung integrated a UI Iayer to it is new operating system and a feature that calIed Samsung Widgets.

At the 2010 mobile world congress Nokia and Intel uncovered a new mobile OS that combines Moblin and Maemo ta create a truly open sourced experience for end users. But at 2011 Nokia closed it is doors to MeeGo and took Windows Phone 7-8 as it is mobile OS (Fling, The Mobile Ecosystem, 2009).

CHAPTER THREE
TRADITIONAL SOFTWARE ENGINEERING DEVELOPMENT MODELS

3.1 Waterfall Model

Waterfall model first mentioned in 1970 by Winston W. Royce in his article. Waterfall model is consists of analysis, design, coding, testing, release and maintenance stages (Figure 3.1). At the starting phase, each stage finds the products of the previous stage. In accordance with the changes which has in its own structure, It adopts what has been delivered to it allowing the next phase to use it in its own way.

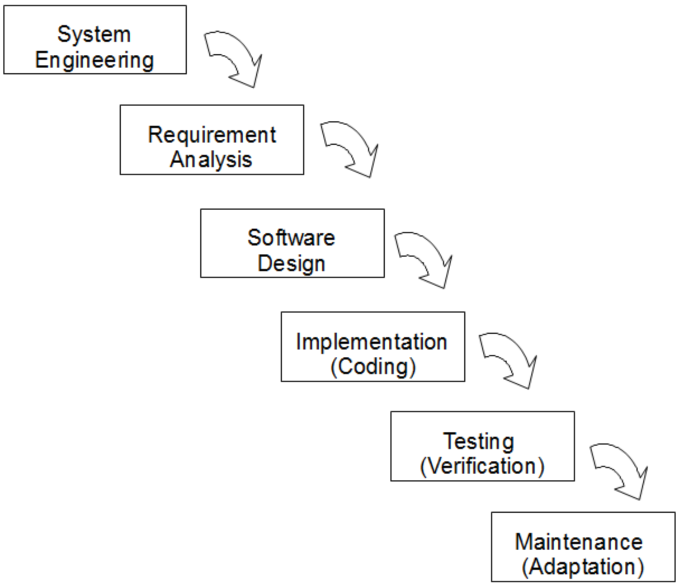


Figure 3.1 Waterfall model

The two fundamental elements of the waterfall model is to provide repeated amendments with less cost due to the feedback between every step, and to examine steps of prototype in its life cycle with the adherence to the needs analysis and design (Mishra & Mohanty, 2011).

The properties of Waterfall model are listed as follows:

- Requirement of being able to move next step of the model is to perform complete activities in each step.
- At the end of each phase, a document is generated so that the participants who join the project in middle stages, understand what has been done in the project easily and can be adopted.
- The user requirements are identified and elaborated at first stage. Later in the design and coding phases, it is not communicated with the customer and the users.

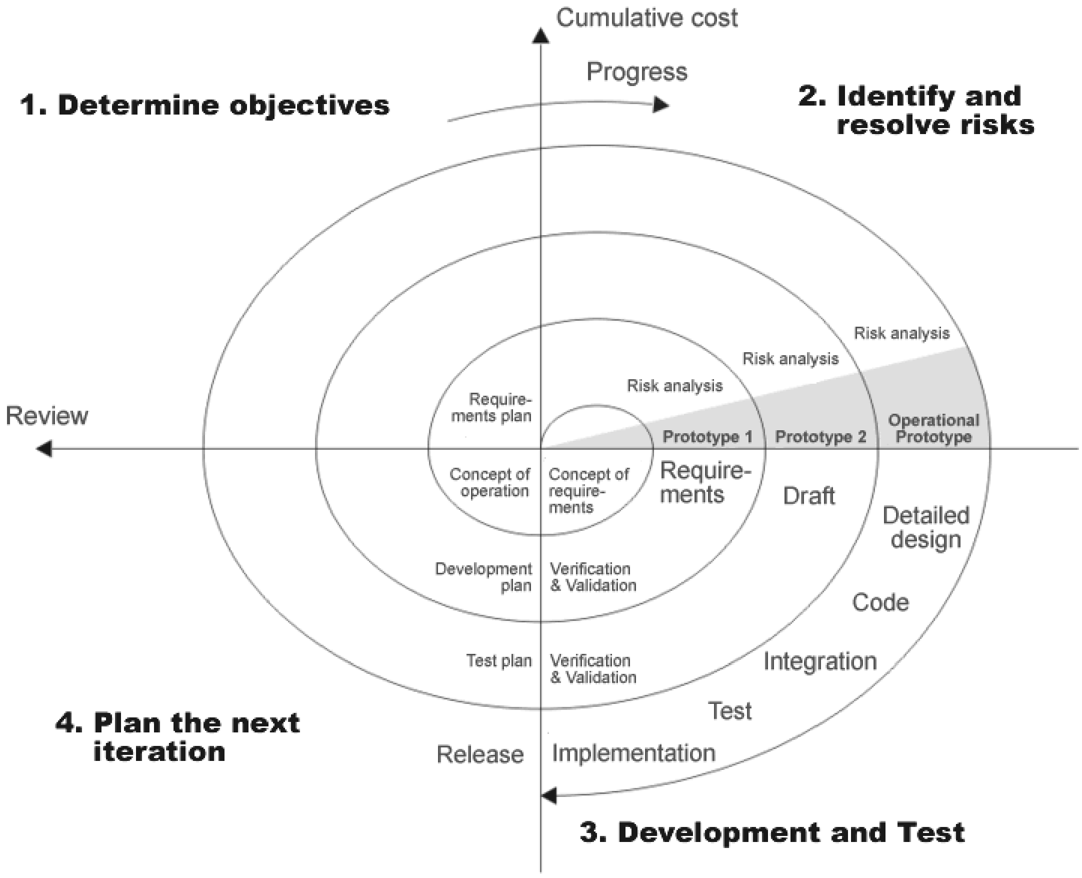
Problems brought about by this model is listed as follows:

- The phases can not be separated from each other.
- In theory, the stages follows successively, when it is switched from one stage to another, it is irreversible to turn back the previous stage again.
- Because it is not possible to turn back the previous stages, model is closed to change.
- Requirements which is not not determined correctly and precisely at the first stages will cause the loss in terms of time and cost when a new requirement is issued.
- Therefore, the requirements should be determined very well. When a mistake is made, it can take a long time to realize, which means cost.
- Being able to use the system can take a long time.

3.2 Spiral

Spiral model is defined by Barry Boehm in 1988. Spiral model does not have a flat flow such as waterfall model. It is a software model which is prototype development-oriented. This model consists of a mixture of characteristics of a prototype approach and the waterfall model.Spiral model is advantageous for big, expensive and complex projects .

In Spiral model, each cycle starts with a design goal,standard software process is completed through the customer reviews & opinions with some of the stages. The purpose in each cycle is to capture the business value. By taking advantage of the results of the previous process, it continues in an improving structure. each phase begins with the objectives and ends with the re-examination of the user.



Figurel 3.2 Spiral model

Basically, the structure of the software development stages are based on continuous improvement (Figure 3.2). As a result, prototype products which are generated in a continuous and uninterrupted way are presented according to the views of user. The program which is developed together with the user is a more beneficial approach for the design by creating a user interface (Mishra & Mohanty, 2011).

Proceed in several stages, starting from the center:

- The objectives, alternatives and constraints of the new prototype are defined.
- The risks are determined by evaluating.
- The development and validation are performed for the loop.
- The validation is performed for the next prototype.

3.3 Incremental Lifecycle Model

It is evolved form of Waterfall model. Product design, completion and testing is performed with the development process in which one series is increased (Figure 3.3). This model is used in demand by commercial software companies and vendors .The software needs of the model is well-defined, but the model can be delayed or it is more applicable for the projects of which basic software quality is defined at early stages.

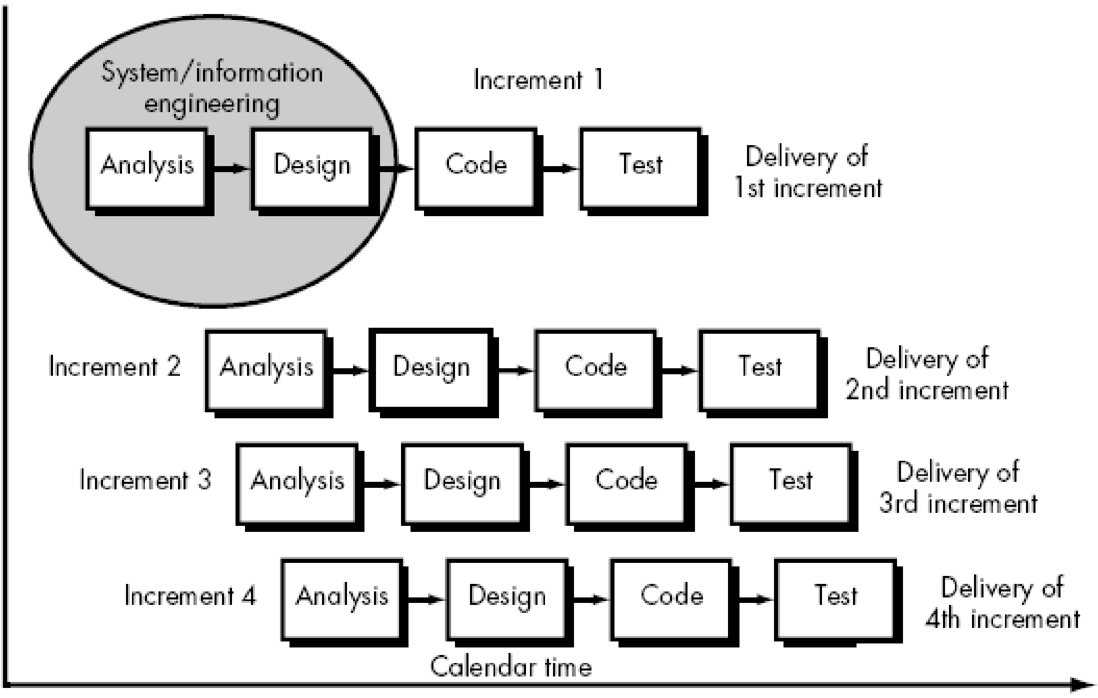


Figure 3.3 Incremental model

At first, it starts with the planning then, continues with the application of design stages and ends with the opening. While it is composed of well-established and essential part of advanced programming, it includes the general framework of agile software development. The basic idea is to ensure the development through repeated cycles and to gain the advantage of what has been learned from the former versions in a short time (Mishra & Mohanty, 2011).

Advantages:

- Designers can have software which works early and rapid.
- To change the perspective and the needs is more flexible and cost effective.
- Testing and debugging are easy with a small iteration.
- Risk management is easy because it is defined in each iteration.

Disadvantageous:

- Iteration in each phase is not flexible, and does not cover each other.
- Problems can arise from the system architecture. Because, all requirements can not be specified in advance before software project.

3.4 V Model

In this model, the "V" shaped path is followed. The left side of the road includes the need analysis and the creation of the system characteristics and the right side of the road includes validation of the elements on the left side. In simple terms, the left side refers to the production and the right side refers to the testing process (Figure 3.4).

Transparency and control of the problem are increased through identified results and reduces the risk by allowing deviations from the plan. The quality is increased by testing provisional results early. With the standard approach all efforts to improve the product is kept under control. All the related elements of the model are based on lowering friction between the developer and the seller.

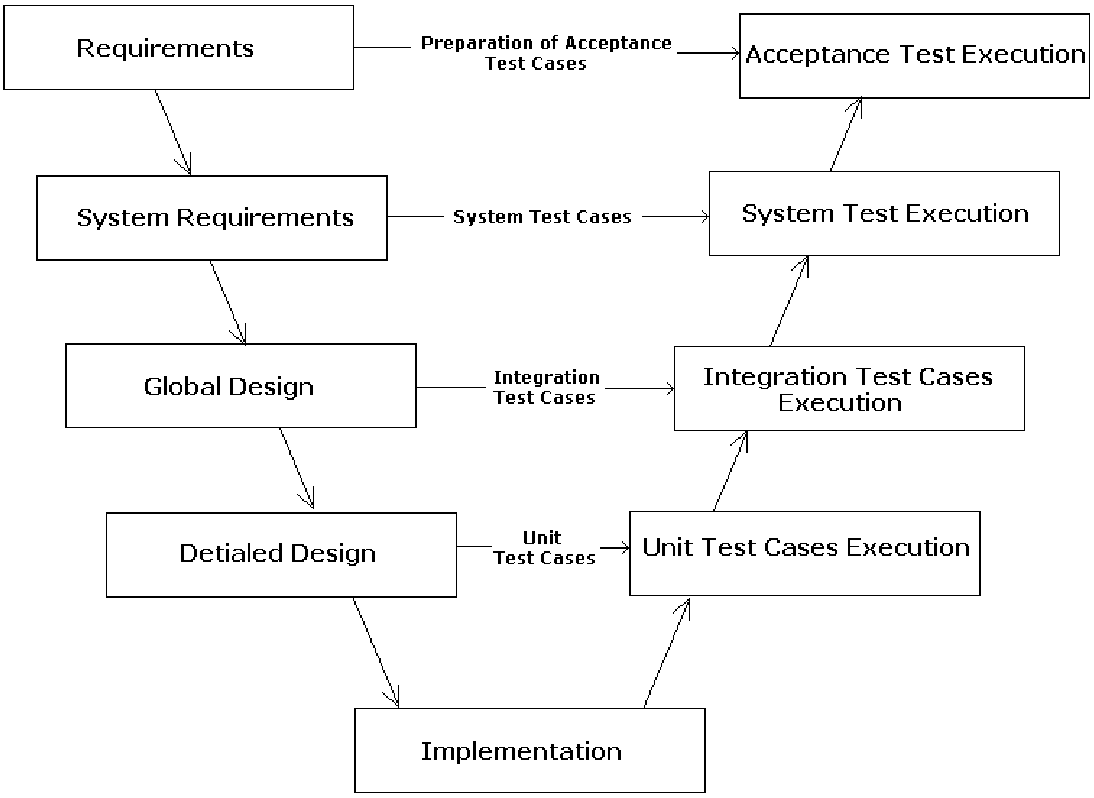


Figure 3.4 V model

While this model covers planning and preparation stages, it has maintenance vulnerability regarding its constraints, The place of service contracts are not determined. It is suitable for the projects in which very small uncertainties issued rather than organizations (Mishra & Mohanty, 2011).

3.5 Prototyping

It is model in which determination of the needs are discussed in a lighter delineated manner than waterfall model at the defining stage of needs. With the construction of prototype, the needs are started to be determined by early obtained software. While determining the needs, a small structure of waterfall model is used (Figure 3.5).

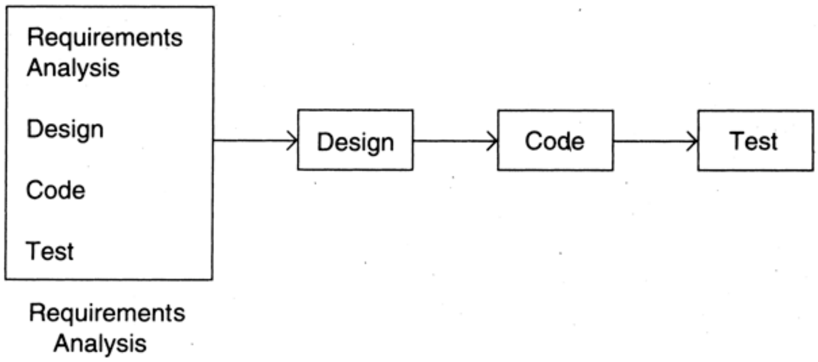


Figure 3.5 Prototyping model

It breaks down the project into small pieces and allows the modifications easily in order to reduce the risk. Users ensure permanency according to their own approval by intervening directly to the development process until the end of the application.A Mock-up undergoes an evolution through iteration for the user's needs. Prototypes are produced with the idea that they will be thrown away. However, for any prototype it is possible to evolve into a running software (Mishra & Mohanty, 2011).

3.6 Agile Software Development

Agile software development consists of a group of methodology which is based on iterations and incremental development. With its sub-methods developed in the mid-90's, it has been used effectively. In 2001, its declaration is defined by publishing the manifesto of agile software development. (Beck, Manifesto for Agile Software Development, 2001) It tries to reach the needs and solutions through teams inside itself. Agile method ensures fast adaptation for the changes to software, easy plan adaptation to change and has an evolutionary design, early product and an approach aiming to make iterations for specified periods. Agile structure always has a vision ahead of the project.

12-item content forming the base:

- Satisfied customer by rapid delivery with a convenient software
- It is welcomed even changes are applied late

- Delivering the software frequently (every week if it is possible)
- The main measure of the process is a working software
- Sustainable development can be achieved through continuous progress
- Daily face to face meetings between customers and developers
- The best form of communication is face-to-face interviews
- The project is configured around motivated reliable individuals.
- Continuous attention for technical advantages and good design
- Simplicity
- Self organized teams
- Proper adaptations for changing circumstances (Beck, Principles behind the Agile Manifesto, 2001)

Agile models, deploys the big plan to small plans by breaking apart the work. Iterations cover short periods of time extending from 1 week to 4 weeks. Each iteration include all the stages of development circle. On this basis, by reducing risks, adaptation of the changes to the project is facilitated. The aim of iterations to provide the less defective products rather than guaranteed products.

The developer teams responsible for the elimination of the needs in iterations. The teams do not apply any other public roles hierarchically. Agile methods emphasize face-to-face interviews of the teams . Thus, teams cooperate to achieve common goals within themselves and between teams. The simplicity of communicating is achieved by keeping a small number of teams between 5 and 9.

At the end of each iteration, the opinions of the customers are taken by provided process study. The new iteration is applied as a new road map in accordance with the views of customers.Due to this disciplined structure, it corresponds to the needs of customers precisely. These road maps are created via face-to-face meetings of teams with each other in which it is discussed what they have done on the previous day, what they do today and will do tomorrow.

There are various methods of Agile methodology:

- Extreme Programming (Beck, 1999b)
- Scrum (Schwaber 1995; Schwaber & Beedle 2002)
- Crystal family of methodologies (Cockburn, 2002a)
- Feature Driven Development (Palmer & Felsing, 2002)
- The Rational Unified Process (Kruchten 1996; Kruchten 2000)
- Dynamic Systems Development Method (Stapleton, 1997)
- Adaptive Software Development (Highsmith, 2000)
- Open Source Software Development (O'Reilly, 1999)

Extreme Programming and SCRUM 's features are described below.They are very well-known and within the scope of software model which is applicable to mobile devices (Abrahamsson, Salo, & Ronkainen, Agile Software Development Methods Review and Analysis, 2002).

3.6.1 Extreme Programming

Extreme programming (XP), traditional development models developed against long cycles. Method took the applicable form after successful trials. In this trials peculiar to XP, the ways of the method was determined and created based on the new methodology tests."Extreme" term comes from these extreme exercises. XP' s life cycle has 5 phase (Figure 3.6). These are:

- Research
- Planning
- Iterations to releasing
- Productionizing
- Maintenance
- Death

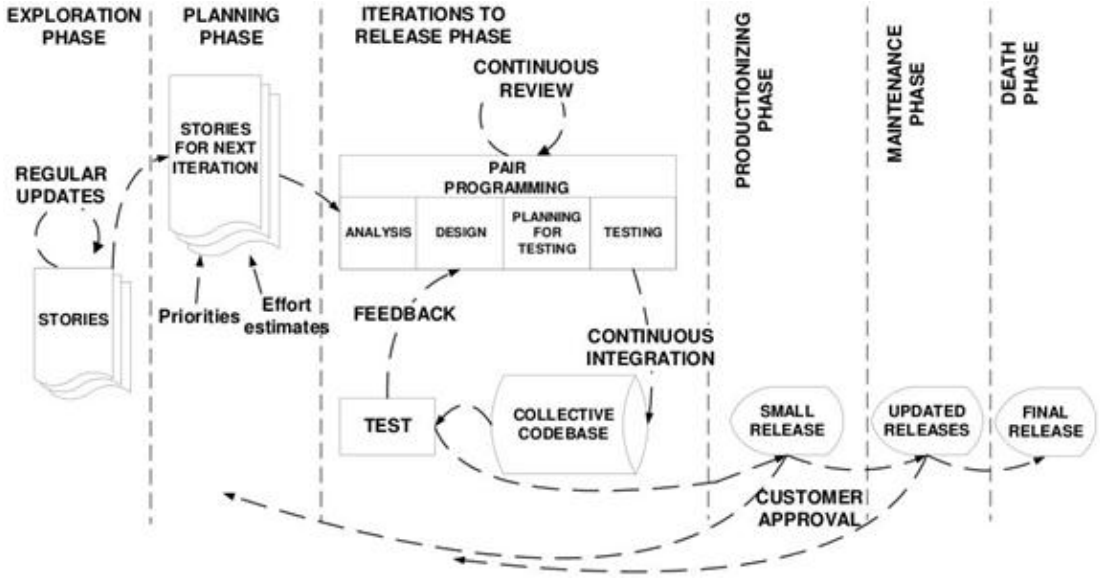


Figure 3.6 Sequence and arrangement of the life cycle of Extreme Model

According to Beck (1999), at the exploration phase the customer writes history/story cards (customer requirements) completely.The program takes shape according to specifications used in these cards. Meanwhile, related technology and supplies are prepared. It takes a few weeks to several months, and a prototype is produced. At the planning phase; first small release is determined through defining the priorities. It lasts a few days, the process of this first small release does not exceed two months. Iteration to release phase, iterations are performed based on the whole system architecture within the period from 1 week to 4 weeks. The customer decides the cards which will be used in iterations. At the end of each iteration, functional tests created by the customer will be tested with customers. Last iteration is ready to production system.

At the productionizing phase additional necessary tests and control are carried out. If any changes is required, accelerated current release is performed within the period of 1 week to weeks. Deferred ideas and suggestions are documented until the next implementation phase.

At the maintenance phase XP project protects the running product against new iterations. it demands the features of support services from the customer. Depending on this, it reduces the speed of the development after the system became product. In addition, this phase may want to add new team members or change the structure of the team.

At the death phase the customer does not have any more story which he want to be fulfilled.It is a system which satisfies the customer in terms of performance and reliability. Documentation is period in which architecture, design or code can not be changed. Death may be activated when satisfactory results are not obtained or possible future expenses would issue (Abrahamsson, Salo, & Ronkainen, Agile Software Development Methods Review and Analysis, 2002).

The core of XP entails 12 applications. These are:

- The Planning Game: The rapid perspective of next release is determined by the priorities and technical estimations.The plans shall be updated because the real time schedule will take a longer time.
- Small releases: A simple system is released rapidly and a new version is released in a short cycle.
- Metaphor: it shares the manual containing a simple basic story of development regarding the operation of the whole system.
- Simple design: The site is need to be designed as simple as possible. The complexities will be deleted when discovered.
- Testing: Software developers write tests which run flawlessly for the consistency of development.
- Refactoring: Software developers should be able to set up the system without changing its characteristics regarding the deletion of repetitions, improving communication, simplicity, and flexibility.
- Pair programming: All items of code are written in one device with two software developers
- Collective ownership: Any person can change the code at any time, at any place.
- Continuous integration: the construction of the system is integrated to every working process most of the time in a day.
- 40 hours a week: it is not permitted to work more than 40 hours a week.

- On-site customer: The developer has no time restriction for asking questions to the customer.
- Coding standards: developers writes the whole code in accordance with the highlighted code rules. (Beck, XP Principles, 1999)

3.6.2 SCRUM

SCRUM term literally means a team game strategy in rugby sport, SCRUM does not identify a specific development technique at the implementation phase. However, it provides the system flexibility for team members to handle against frequently changing environment. It takes measures against the stakeholders' pressure and the deadlines that agile manifesto can not provide.

An empirical process control is available. SCRUM does not base its process on assumptions or estimates that are covered with adequate knowledge. Instead, it creates a plan and timeline taking the advantage of the real-world processes, Scrum is divided into two or three weeks continuous and succinct works called sprint typically in the form of short runs (Figure 3.7). At the end of each sprint stakeholders and team members determine the next steps. These determinations are made according to the former works not to estimations as mentioned previously. If the consistent structure puts the team members under pressure, SCRUM's adaptation and flexibility ability makes it attractive.

There are three basic structure in the SCRUM. These are Product Owner, ScrumMaster, and team member. The product owner's duty is to ensure the communication with developers to ensure the progress of the project in a correct vision. ScrumMaster's role is to provide communication between the developer team and the product owner. Team member is the person responsible for completing the work.

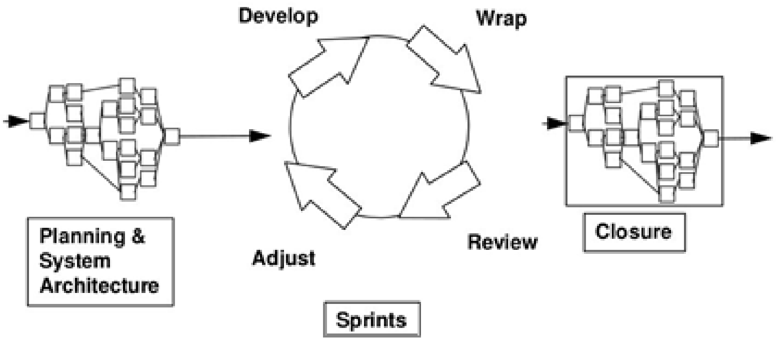


Figure 3.7 SCRUM methodology

Characteristics of SCRUM method stands. These are:

- At the first and last phases (planning and closing) processes are well-defined as input and output. The method regarding how the work shall be carried out is known.There is a flat flow.
- Sprint phase is an experimental process. Most of the process inside it are not defined or controlled. Controls are carried out as black box. Therefore, due to the risk management, flexibility of iterations in sprints are maximized and confusion is avoided.
- Sprints are non-linear elastic phases. While trial, error, and unspecified known information constitute the process information, ready and proven process information is used in sprints. Sprints are used to achieve the final product. To sum up, unproved and unrealistic information is not used.
- Project is open to change till the closure phase. At the phases of planning and sprint desired parts can be changed. Project remains open to environmental challenges, competition, time, quality and financial pressures in every respect.
- Delivery is defined according to the environment during the project (Schwaber & Sutherland, 2011).

CHAPTER FOUR
MOBILE DEVICE FEATURES AND LIMITATIONS

The first layer accessed by application developers for mobile devices is application framework or Application Programming Interfaces (APIs). Of course, the companies of mobile device products or the operating systems control the permission of access freedom to these mentioned products.

Application frameworks are the top layer which is used in the application development. By means of frames, how to use the resources of the mobile device are arranged. All these provide accessibility for contacts, messaging, location information, security, and so on. While frameworks provide certain standards, the same can not be said for mobile devices. Because when developing a mobile device application, it is necessary to know the version of platforms in its corresponding frame such as Java . Additionally, it is faced with constraints such as the screen size, processor power, graphics capability or use of keypad and touch method and etc. In this section the topic addressing these issues shall be open (Fling, Mobile Design and Development, 2009).

4.1 The Screen

The first element to consider in developing applications for mobile devices is to be able to have small screen. While there are dimensions range from 14 inch to 21 inch for desktop devices, the dimensions decrease ranging from1,5 inch to 8 inch for mobile devices for the same situation. Third factor comes to the fore when we consider the screen design. These are resolution, physical dimension and view ratio.

The resolution comes to the fore in mobile design. A standard screen resolution is not achieved for mobile devices. The first-term devices have resolutions up to 128×160 -128×128 pixels, medium-term devices have resolutions up to 176×220-176×208 pixels, and touch-screen devices or smart phones have resolutions up to 240×480-320×480- 360 × 480- 480×800- 480×854 - 640 × 960 pixels as shown figure 4.1. These devices have different hardware, software, and platforms. The standard resolution VGA (640 × 480 pixels) is used as QuarterVGA (QVGA) on common mobile devices with a resolution 240x320 pixels, HalfVGA (HVGA) is used for devices like iPhone. This problem has been simplified in web technology. Mobile web browsers offer a screen resolution based on the mobile device. While developing an application it should be considered that there is available all kinds of screen for devices.

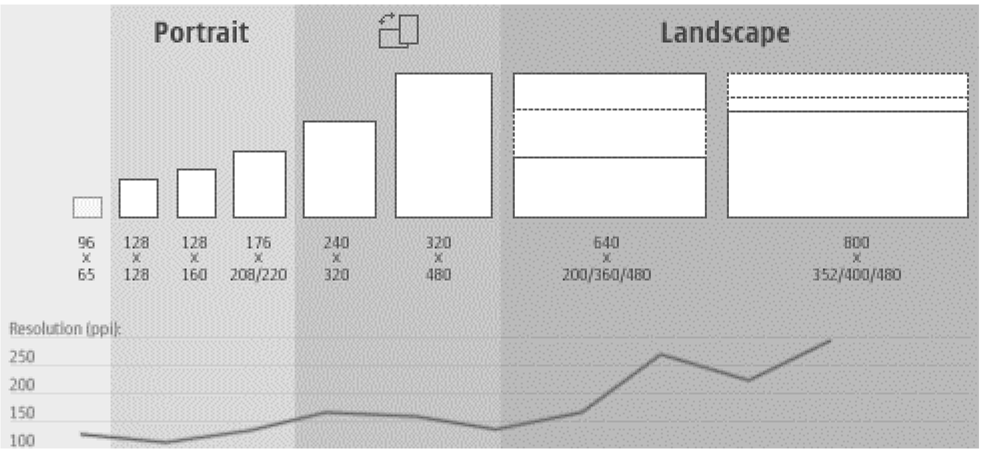


Figure 4.1 Diferent Mobile device screen sizes

Physical environment is required to be used efficiently for the mobile devices.This efficiency can be increased by the correct usage of PPI Pixels Per Inch and DPI Dots Per Inch.Thus, the clarity and quality of the pictures when zoomed image can be transferred to the user without deterioration. In addition, instead of filling the screen with unnecessary application elements, user-oriented design should be provided by leaving space for a comfortable reading via enlarged fonts. Some mobile devices offers satisfying image applications up to 300 ppi which human eye can not detect.

Mobile devices of today varies according to the outlook proportion. When we consider the height of screen, vertical devices, when we consider the width, the horizontal devices are issued. Some mobile devices (usually smart phones) with sensors which understand the position regarding horizontally or vertically provide the orientation of the screen according to the request of user. Besides, there are devices of which the height and width is equal. In this respect, the designs which meet the needs of user should be considered (Firtman, Programming the Mobile Web, 2010 , p. 12-13).

4.2 Input Methods

Input methods for mobile devices vary. Nowadays, virtual keyboards, touchscreens method has become prominent. However, Keyboards with QWERTY keys which are suitable for English language are widely-used. List of method types are as below:

- Numeric keyboard (2abc, 3def ...)
- Alphabetical keyboard (QWERTY)
- Touchscreen keyboards
- In multi-touch screens, simultaneous multi-touch detection
- Virtual keyboards on screen
- Expandable external keyboard (wired / wireless)
- Laser keyboard (in front of the mobile device)
- Handwriting recognition and processing
- Managing keyboard through customized pens
- Warning and recognition through voice

Moreover, above mentioned methods are developing day by day. Some of these methods are used together, some are used single. While determining the requirements, also input methods should be taken into account. Since touch screens has come to the fore recently, their interaction with the screen affects the input method preference (Firtman, Programming the Mobile Web, 2010, p. 14).

4.3 Network

Since most of mobile software is operating through the network, the network or internet connection take part an important role as a method. Especially, devices like smart phones that use the Internet protocols such as Wi-Fi, 3G, 4G, GPRS or EDGE technologies get connect to the network. The connection of devices through Wi-Fi confined to the areas of which diameter is 100 meters. Except the Wi-Fi, the devices using GSM technology (Global System for Mobile communication) are available. At early networking technologies Voice over IP (VoIP) was used to transfer audio. This period may also be called 1G. Then, through GPRS (General Packet Radio Services) technology, 114 Kilobytes per second (Kbps) (2.5G) data transfer bandwidth was applied. Through 384Kbps data transfer bandwith, 3G and beyond started. Therefore, it enabled to send both video and audio data. Due to the high bandwidth extent, mobile devices users able surf the internet via Wireless Application Protocol (WAP) (Prathaban, 2004).

Connecting to the network protocols varies. A mobile ad hoc network-MANE is a set of protocols connecting mobile devices to the network via wireless methods. Mobile devices to be connected are defined as nods. Because the structure is movable, The IPs of devices vary while moving to another terminal. For this reason, network topology also has to be dynamic. The abundance of devices affects the efficiency of bandwidth extent. There are problems needed to be overcame regarding the unification of existing network with other networks or the division of it to other networks. Duplicate Address Detection (DAD) procedure and IP conflict are made device-specifically. However, in the case of there is being a growth through the unification of network, it is not possible to have a single IP. Due to the other proposed Distributed Dynamic Host Configuration Protocol (DDHCP), all devices keep IP list which are reserved or available.

Dynamic Configuration and Distribution Protocol (DCDP), creates IP groups to the devices. Weak point of this approach is not being able to have unification or division of groups. for the solution of this problem, it is proposed to form IP groups (subset) under IP groups.With the described protocol and processes, a unique IP adjustment facility is provided to the devices from TCP / IP protocols automatically. A crowded multi-hop topology is come out (Ezzouhairi, Quintero, & Pierre, 2005) (Wehbi, 2005) (García Villalba, Sandoval Orozco, García Matesanz, & Márquez Díaz, 2011).

Considering the above mentioned general approach, server must be adjusted so as to meet the growing mobile device traffic. MANETs are highly heterogeneous environments due to the diverse nature of communications technologies employed, as well as the presence of the different types of nodes. Since mobile devices are used mostly during working hours, it is certain that the server has more traffic. In addition, it is important to provide network consistency while changing the terminal for critical mobile applications which need to be connected constantly to the network (Ezzouhairi, Quintero, & Pierre, 2005).

Server and client relationship should be defined on the server against disconnnection of the network. For example; the user downloads his e-mails to the mobile device because of its small bandwidth. Due to that, filtered and required content should be downloaded. These scenarios which are suitable for the needs of desktop should be prepared according to the mobile devices, also. Besides, it is required to define the setting mechanisms depending on user preferences during adaptations.the Server should allow tuning the desktop or mobile device to be able to filter the necessary information.

4.4 Memory and Processor

The most important issue of Mobile application to be considered is memory management. Memory management of the application should have a form which is easy for device and not generating redundant unnecessary data therefore filling the memory. If we examine two parts, they can be divided as macro and micro. At the macro level, the application itself is moved to the memory, such as interface, forms, buttons when the application is opened. After this stage, the user’s own entries start. User entries are at the macro level till the operation of the algorithm for the result. The screen management come to the fore because of window constraint of mobile device regarding the memory. At micro-level the application runs its algorithm for the result. The algorithm of micro-level application must erase the data which is not needed without accumulating because of the constrained memory.

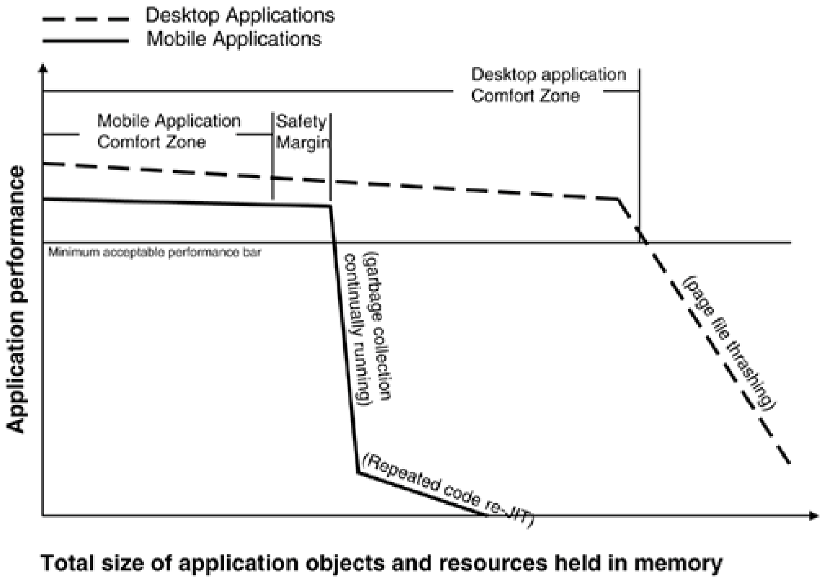


Figure 4.2 Application performance trends with increased memory consumption

As shown in figure 4.2 mobile application performance degrades due to the sources held in memory.Although desktop devices have a large memory size, at the end poor memory management reduces it after a while. The performance of the device can be upgraded with a good memory management. Mobile application should be developed by thoroughly inspecting how much application data is loaded into memory and which overall resources are loaded into memory and in which circumstances installed data / source waste removed (Salmre, 2005).

Although the memory and the processor speed of today's mobile devices are high. Application performance degrading, because of plentiful, unnecessary use of resources and unnecessary code loops.

4.5 Battery

Energy source creates a huge problem for mobile devices.Energy consumption of smart phones on standby mode reaches up to 3 days. when its all features are used (interviews, entertainment, movies, videos ... etc) it is reduced to 18 hours. It is the same for laptop, tablet PC and other mobile devices for different periods of time.

It is possible to develop the mobile application and the operating system in terms of less energy consumption. To do this:

- Dimming your screen at minimum level during the daylight hours.
- Turning off the display when switched to phone mode.
- When 3G connectivity is poor, EDGE is switched.
- When GSM signal is weak or broken, switching to "airplane mode".
- Displaying the remaining amount of energy according to the usage form such as talking, listening music, watching video.

The above-mentioned issues should be considered for software development regarding the energy of mobile devices. Thus, by providing an efficient and continuous operation, leaving at a critical point or in the middle of procedure of device and application is preventable (Taub, 2010).

4.6 Other Important Features

The specifications of mobile devices has increased via becoming smaller dimensionally. software which are developed through specifications that can be used in combination with each other and the user requests are becoming more complicated. Some other properties of mobile devices are compiled as following.

Most mobile devices can determine its geographical coordinates. They can manage this with a GPS, WPS (Wi-fi Positioning System) or a mobile phone follow-up technology of the base station. In addition, as mentioned earlier, mobile devices can change the direction of the screen by sensoring the position. Even they can give the number of steps through the same sensors. As we can see, all the sensors of mobile devices ensure to make a very versatile and flexible design for designer.

The most important features of mobile devices such as smart phones, SMS-Short Messaging Service, MMS-Multimedia Messaging Service features. You can send a message between 160 and 70 font or multi-faceted message unified picture-video-text add-ons. Besides, calling feature is the objective existence of any mobile phone. Of course, cell phones used to have only these features in the first place, but nowadays smart mobile phone models increasingly make more complicated the software development.

Frequent use of location detection technology in the smartphones increases the user's expectations in this aspect. Users can determine where to go by accessing map applications in a very easy way. In this respect, devices make the user dependent on themselves. Map error in software becomes very difficult to compensate. Map applications detects not only the places to visit but also the hunting areas, mining areas or snapshot follow-ups.

4.7 Surveys and Market Analysis

Keynote's 2012 Mobile User Survey is based on a smartphone and tablet. 5.388 panelists of whom 3.145 smartphone users and 1.976 tablet PC users take participate in the research panel of Keynote. More than half of the respondents are working class and 77% of them are graduated from university or an equivalent school.

Android operating system has the largest share in smartphones. Afterwards iOS comes. 57% are using 3G, 27% are using 4G. The first three most performed activity in smart phone:

- Obtaining location information (88%)
- Search information (82%)
- Social media sites (76%)

Smart phone users prefers mobile applications for mapping information, social media updates, e-mail and bank process to mobile sites. They prefer mobile sites for news, dining / entertainment (info), travel (info), and in particularly shopping info.The biggest frustration takes place when faced with slow loading sites.Every 2 smart phone users of 3 demand the loading time for a site less than 4 sec.

Apple come to the fore among other tablet brands. Amazon and HP, respectively, follow it. iOS operating system comes to the fore rather than Android. In tablets 77% Wi-Fi is preferred in order to connect to the network versus to 12% 3G and 7% 4G method. Most-outstanding activities:

- Reading News
- Entertainment
- Searching For Information
- Watching Video

The choice between mobile application or mobile site is the same like smart phones. Every 6 tablet user out of 10 ask for the download.The biggest frustration is slow loading web pages (Authority of Keynote, 2012).

In 2011 and 2012, world sale of smartphones reached 468 million units and got a %57.7 percent increase according to Gartner Inc.' s research (Table 4.1 and Figure 4.3). And Android OS will get the leading flag with %49 of the market on 2012 foresees.

OS	**2010**	**2011**	**2012**	**2015**
Symbian	111,577	89,930	32,666	661
Market Share (%)	37.6	19.2	5.2	0.1
Android	67,225	179,873	310,088	539,318
Market Share (%)	22.7	38.5	49.2	48.8
Research In Motion	47,452	62,600	79,335	122,864
Market Share (%)	16.0	13.4	12.6	11.1
iOS	46,598	90,560	118,848	189,924
Market Share (%)	15.7	19.4	18.9	17.2
Microsoft	12,378	26,346	68,156	215,998
Market Share (%)	4.2	5.6	10.8	19.5
Other Operating Systems	11,417.4	18,392.3	21,383.7	36,133.9
Market Share (%)	3.8	3.9	3.4	3.3
Total Market	296,647	467,701	630,476	1,104,898

Table 4.1 Gartner mobile OS market research (x1000)

Open OS' s devices will get their hands on %26 of the sales by the end of 2011. And at 2015, we could think an 1 billion sales that counts %47 of cellphone sales. Android' s best part is it's low price. So maufacturers can deliver cheap phones. After this prediction, Gartner says that iOS will remain as second biggest OS on the market. By this prediction, Gartne reveals their believin' in marketing strategies not only sales. And Gartner hails US and Europe's matured markets.

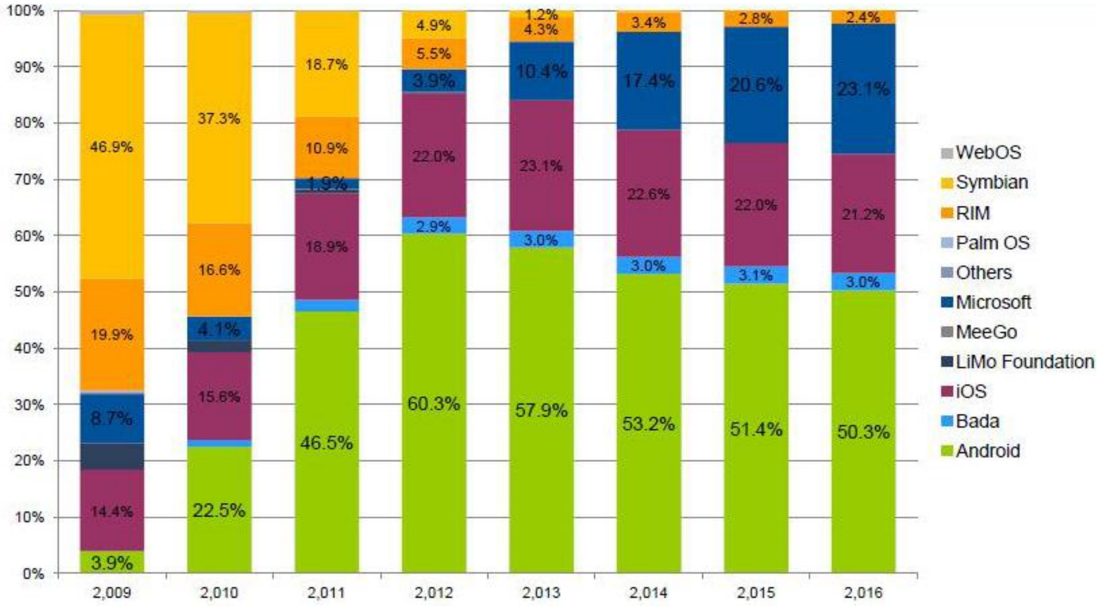


Figure 4.3 Gartner mobile devices by open operating system, worldwide, 2009-2016, 2Q2012 update

On their Nokia future, Windows Mobile will be added to their mid. Priced portfolio by 2013. Windows Mobile will be the third biggest on the scene. Although this can' t be near Symbian achieved with Nokia.

Gartner' s analist think that there will be a large mobile ecosystem. And on future years, this ecosystem will lead the sales and investors openOS devices can add will change the way. In some point , smartphone users will want same OS for their tablets. This will allow a nice share of happy usage for consumers (Pettey & Van der Meulen, 2011).

CHAPTER FIVE
MOBILE SOFTWARE ENGINEERING LITERATURE

While developing software for mobile devices, using the desired model process is in our hands. Ranging from the waterfall model to agile model there are a quite wide selection of models. However, the model needs to be developed because of unique features of mobile devices. In this aspect, mobile software engineering comes forth. The basic steps such as software needs, design, programming, testing and maintenance shall be the same in the model to be developed on the desktop or the web. But experience shows that these steps can not be transferred as the same directly to mobile software engineering.

While developing mobile software, agile modeling comes forth with the features such as providing an early prototype, having a variety of of testing and fast adaptation period. Although agile tests to be applied in mobile software engineering are not just peculiar to themselves, they need to be processed again. Due to tests that are used for mobile software, sources will be managed more easily regarding the mobile constraints.

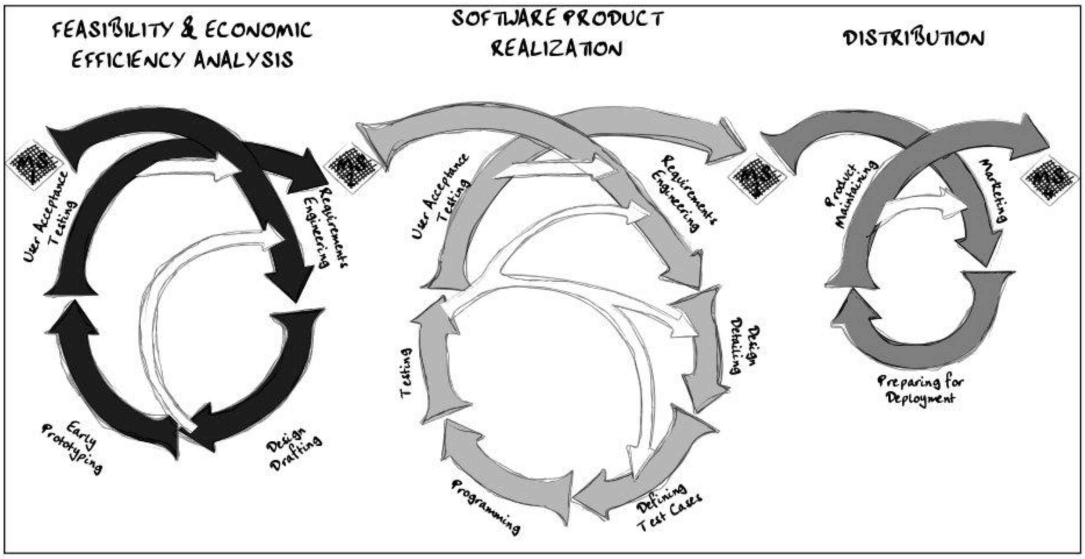


Figure 5.1 A mobile software engineering process from Introduction to Bada Application

In the book ‘Introduction to Bada Application’ mobile software engineering process consists of three main phases (Figure 5.1).These are:

- Feasibility and economic efficiency analysis phase
- Software product realisation phase
- Distribution phase

In the first phase (Feasibility and Economic Efficiency Analysis) feasibility and economic efficiency is checked basing on mobile software solutions. It consists of four phases. In Requirements Engineering, user describe the needs and begins the analysis of the requirements. A simple pre-prototype is produced according to the user stories. This prototype has a simple user interface. Thus, it helps user to think of the next program which will come out of this prototype with a more tangible approach. In Design Drafting stage user interface UI is determined and the compounds of prioritized software architecture are discussed. In Early Prototyping, a pre-prototype of the software is produced to grasp the software in general. User acceptance testing (User Acceptance Testing) is done as an option. It allows the re-determination of overlooked issues in the development environment.

In the second phase (Software Product Realisation) is built on the obtained results from the first phase. Many iterations and communications are performed in this phase and it consists of six steps. Requirements are revised. The design, programming and testing phases are examined together with stakeholders. And, the interface and sub-architecture are elaborated Then, test cases are determined out of the needs. Then, a more conservative program is obtained which able to pass through tests. Tests are usually carried out in desktop simulators. However, the testing of the screen compliance, the battery status and the light conditions of the mobile device on real equipments gain importance. As a result of these tests, small iterations occur. Later, the user's opinion is asked optionally. Then, with the approval of shareholders it is released.

In the third and last phase of the distribution, the software that come out as product is introduced to the market. This phase contains less iteration than the other two phases. it is tailored according to market entries of software products . Then, the software content and the parts which will be downloaded are created and released to the physical environment. Maintenance services are provided thorough bug fixing, and support and feedback. The maintenance level can be also determined by the approach of provider. In the last stage, termination of project is identified by the project managers and stakeholders (Somerville, Luo, Lansdell, Morris, & Bortenschlager, 2010).

Kyle Lutes who is the faculty member at Purdue University decides to develop mobile software of PDA or cellular phones for mobile computers with students in semester. Kyle Lutes states that there are two stages of mobile software. These are:

- To learn the mobile platform and the related works which will be used in the development of the software.
- The use of appropriate tools İn the development of mobile software platforms.

To be able to provide the above steps, he gave assignments, quizzes and surveys to students. 6 items are presented with that educational process:

- The different input / output (I / O) operations and the differences in the structure of the main file.
- Graphics programming challenges that arise because of the lack of keyboard as issued at PDA devices.
- The design of the user interface is different from desktop programs.
- To overcome the challenges of connecting to a server or to the web as online.
- The decision of regarding whether running a motor such as Microsoft's SQL Server CE on the device for data base management or not.
- Management of network traffic with the direct use of protocol that is used for web.

These items for the projects have been revealed to be overcome. All in all, these are today's main problems of modeling.The only difference is now there are platforms and approaches which are appropriate and more practiced for mobile devices. However, by the end of the project, quite successful works are presented even under circumstances of that date. In an other approach it is not set up constraints to the project teams regarding the number. However, the problems have been experienced when more than one or two-person development teams are involved in the studies regarding management, communication and timeline. In addition, with the leadership of a faculty member or assistant, student or teams are guided and the aims are raised to the upper levels through a 20-minute interview in a week (Lutes, 2004).

To create a mobile software methodology characteristics of mobile devices are need to be composed by analyzing each direction. In addition, a very rapid growth of the market for this development should be considered. Rapidly developing features which are used to generate a mobile modeling:

- Profits and capabilities of the open source mobile operating systems need to be evaluated in detailed way.
- The appearance and the feeling of user interface create charm in terms of design.
- To connect a terminal online is high demand.
- It is required to use the resources with proper mobile software engineering method for the development of software and hardware not to cause high efforts and a lot of money spent.
- A wide range of applications based on the needs are developed by both the manufacturer and the client.

The above-mentioned features highlight the need for the development of mobile software engineering method regarding the future evolution and strategy of mobile devices. There is a need for a mobile methodology which can realize predictable objectives of future by examining the main methodologies and major software platforms (Hammershøj, Sapuppo, & Tadayoni, 2010).

Some requirements came out as a result of research conducted on five distributed releases which examine different areas from each other on software engineering. The reason of these requirements is that the existing models of software engineering does not able to correspond to complicated environment of mobile software field while meeting the needs of existing traditional environment.

Firstly, the reliability, performance and security of mobile software development need to be increased by appropriate mobile modeling. Secondly, reusable components and standard approach should be developed in mobile software. Thirdly, the design and analysis principles in theory should be reinforced to perform the case which is introduced in the first two states. And fourth, the changing and evolving mobile environment should be coordinated within itself. And fifth, the evolving dynamic elements including the application topologies should be developed. Finally, mobile software models and tools should be developed for implementation of the requirements of fourth and fifth phases. The main five releases which have been used to identify these determinations which emerged as a result of the experience basing on mobile needs in various fields [Introduction to Special Issue on Distributed and Mobile Software Engineering; Automated Software Engineering; *Carlo Ghezzi, Paola Inverardi;* 11, 203–205, 2004 Kluwer Academic Publishers. Manufactured in The Netherlands.] Adaptation problems arise since a lot of complex mobile software has been developed. The relationship between the development of supermarket and the demand of customers leads the increase of complexity of applications gradually (Roman & Murphy, 2002).

There are a couple of methods which have reached success in mobile development process regarding past experience and expertise. These methods have produced solutions by approaching events in their logical frameworks. One of them is Pattern-Oriented Mobile Web Engineering Methodology (POMWEM) by Pankaj Kamthan. it has been examined and aimed a mobile approach to improve quality. At the end of research, the necessary definitions and patterns revealed:

- Regarding a device connected to the network or not, the quality determined.

- Wide variety of ways basing on the purposes through the social network connection availability.
- Personnel training and resource planning phases emerge in accordance with the field of mobile technology when viewed in an organizational way.
- The obligation for development regarding the constraints of mobile devices have introduced

In order to create a high-quality mobile product systematic opinions, successful development, release, maintenance, science, engineering, principle management should be applied with a high discipline. Existing mobile software models can not be used directly on mobile softwares. Reliability, availability, and accessibility are discussed under the scope of the existing models. However, how to improve the quality of these mentioned and unreviewed issues are not stated. Some released principles leave missing and unclear points regarding screen management rules. A linear process can not answer to flexibility and timing for required mobile modeling. In Mobile-D model obtained from XP, macro architectural details are not explained. In mobile software of spiral model, the design phases are available but it is not discussed how to improve the quality (Abrahamsson et al., 2004). The Oregon Software Development Process (PSSP) (Schummer & Lukosch, 2007) can be used in a moderate mobile application. The Pattern-Oriented Analysis and Design (POAD) (Yacoub & Ammar, 2003) approach fails to provide the process between the customer and the user. In a model regarding the security issue, the patterns except safety are not explained (Fernandez et al., 2007). In the model abbreviated as OBMP, usability is taken into account while other quality modelings are not referred (Ocampo et al., 2003). In a persona-driven, pattern-oriented approaches, user patterns of experienced staff are skipped and limited approaches are not discussed.

POMWEM produced by benefiting from past experiences. The purpose is to provide a systematic addressing of the quality for mobile applications (Kamthan, 2008a; Kamthan,2008b). According to Hypothesis 1 the improved development processes should increase product quality (Nelson & Monarch, 2007). According to Hypothesis 2, to be able to identify the quality of mobile applications from the perspective of the stakeholders' qualities should be segregated into manageable parts. According to Hypothesis 3, preventive approaches in quality addressing should provide significant improvements at least (Dromey, 2003). POMWEM including this hypothesis covers the following three steps:

- Selection of development process model.
- Road map from a certain point of view (Semiotic Viewpoint, Shanks, 1999) viewpoint, quality, defining and organizing their interest.
- To achieve the appropriate pattern, select, and implement.

A particular development model is not enforced in pattern-centered approach. However, mobile application development process should be flexible, people-centered and caring for quality. In this sense, the process should be a model which is progressing slowly and with iteration or incremental. XP is a model based on agile methodology and suitable for small mobile applications. RUP, Open UP, OpenUP / Basic (open source) and The Unified Process (UP) (Jacobson, Booch, & Rumbaugh, 1999) are incremental models following over the first model. RUP and OpenUP are based on heavy modeling and documentation. OpenUP / Basic is in structure of the agile method can be used from small to medium-sized application development. XP and OpenUP / Basic methods have similar properties for professional developers. A model can be put forward by complementing each other. This model needs improvement though additions. Three hypotheses must be adhered while selecting the process model to be used in mobile applications. The selected process model should follow the timeline, use qualified personnel, be supported by handy gadgets and include the budget (Kamthan, 2010).

One of the greatest difficulties of developing mobile devices is the rapid growth of technology. This difficulty arises in the adaptation of the software to device. As a result of the rapid growth of digital media platform, The Helix DNA, the addition of new features are appeared. In the software development process to be used for growth, reliability and bug-free issues emerged in the platform which has more than 150 000 members. In addition, during the meeting of multimedia security needs in devices, safety-critical software defects grows. In automotive mobile software, life critical situations arise. The explosion of the growing market in mobile devices puts the foreground the time and market relationship in the firms of software market. Thus, the software development processes utilized in the products are not able to approach the software in global standards and leads to poor quality short development cycles. Variety of software platforms increase the complexity and size of software (Bouchard, 2010).

The rapid growth of mobile devices leads to more than 250.000 applications. Application of different devices and models is possible to find in supermarkets and department stores. In fact, the multiple platforms of these applications are available. As a result of the conducted research the following points are reached for the mobile software:

- Mobile applications consist of almost a few thousand lines of code and are designed and complemented by one or two developers.
- There is a sharp division between native application which is running on the device and the web application which a small part of it locates on device.
- Developers rarely use the appropriate development models.
- Developers do not record and bring together the ways they used for development and the metric information.

A variety of platforms are used for mobile devices in software development. These platforms have their own development tools. In addition, there are available best exercise samples in offered development samples. Due to these materials, small and medium-level mobile software which are appropriate for platform architecture can be improved and updated easily by one person. However, today, software development is shifting to the side of business-critical field which emphasis on high quality and safety from the entertainment-centered software.The distinctive features of mobile devices:

- Mobile devices have numerous applications while most of embedded devices are loaded with factory software.

- They have many abilities through their sensors.
- Network connections can receive data from the ınternet.
- Due to mobile software platforms, there is no need for embedded software for hardware. Platforms can be installed and applied to the most of devices.
- While the embedded devices are closed to the data transmission, in mobile devices the data can be transferred to the outside. Safety precautions must be enabled.
- While the user interface is controlled by the user in the embedded software, the general principles are used in mobile applications. The criteria of developing interface is set by the user.
- The telephone network and gateway include a variety of tests.
- Mobile applications need to keep the battery consumption at minimum level.

Types of agile modeling are used in mobile application development. However, for the usage of these models, stated guidelines of mobile platforms and best technical practices provide guidance.Technical overview can not provide analysis to large-scale software issues (Wasserman, 2010).

Business-critical situations of mobile devices are increasing continuously. Bandwidth and security challenges are the most important needs of the development. Health sector, army members and administrators use smart phones or tablets for their personnel. In case of weak network connection, communication problems lead to crises. Some questions issue such as if an administrator lost his device, the information in the device will be able to be protected.

Today, to be able to test the mobile applications, traditional test methodologies are used in mobile applications by being reinterpreted. Many developers think that emulator-based tests would be enough. However, how much it decreases the risk? Is there a need to test the device in real environmental conditions? In addition, in these tests, composite applications that are brought the different units (database, services, Guise ..) together by developers should be united not only as a one piece but also testing part by part. Due to low performance of the testing, the rupture and colapse occur in network connections of devices. Especially web pages should be optimized such as mobile applications.

HP is using its own Quick Test Professional (QTP) testing platform by benefiting from agile and continuous development processes regarding test issue. It implements its own scripts to both device and emulator. Updates take place over network. Security is based on experience of developers due to the scripts like real user. However, mobile networks increase excessively over time.Thus, a mobile network connection which is inconsistent, error-making or conflicting is established. Mobile applications need to be designed according to these inconsistent network connection (HP, 2012).

Mobile softwares which are developed for learning provide an interactive learning. With these applications learning can continue at every stage of our daily lives. Especially for students, a consistent learning is ensured in daily life through mobile learning applications. While learning tools can be used only in classrooms, the review of notes, calculations, access to school databases and such similar processes will be more comfortable and easy by means of mobile learning software. In these applications, the most important problem is the small screens of devices. Screen designs of applications come to the forefront for successful learning. are more emerging display designs to the forefront. Successful application screen design for learning come to the fore. To do this, tests in accordance with electronic learning are used. Three types of tests are performed and measured:

- Psychometric Tests
- Psycho Physiological Measuring - Biofeedback Test
- Eluat (E-Learning Usability Attributes Testing)

After passing the appropriate tests described above, the appropriate interface and screen design of application are made. Application interface is being developed by taking measurements such as stress, attention and concentration in tests for the appropriate screen design (Fetaji & Fetaji, 2009).

With rapid developments of mobile devices smooth operation and security of the software came to the fore. The competition and opportunity-creating movements between software companies have directed the mobile software devices that they have developed to the test platforms which have been automated in a cheaper way.Thus, the market have been filled by quickly tested, cost-oriented softwares. Softwares are full of bug errors. In addition, although they have the same platform they are not able to work sometimes beceuse of different hardware systems. 6 basic questions must be answered for the test criteria to be used:

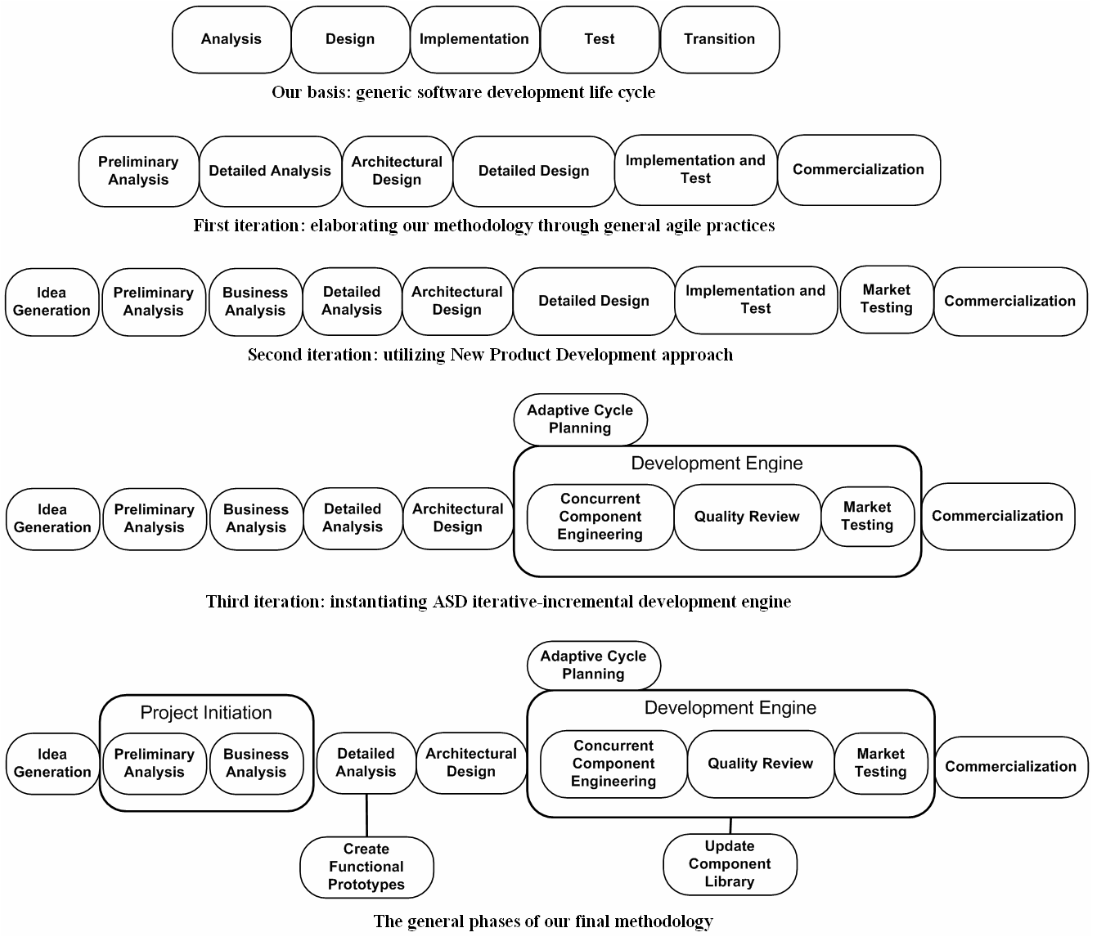
- Which mobile device and platform will software support?
- How will we test the applications to make sure they work on platforms?
- What changes should be applied to ensure cross-platform compatibility?
- How frequently performed product innovations will be supported in the future?
- How development and testing processes will adapt to the different wireless network protocols and mobile service providers?
- How will be decided how much test is enough?

If the mobile software test to be applied is performed as emulator, it should meet the interfaces, tests types and capabilities of mobile device. However, the actual device should be used if the network connection is tested in different places. The difference between these two cases is that the cost of the emulator is cheap and real device is expensive. Regarding emulator, required popular handsets are at our hands. As to the real devices, they must be purchased. The result to be achieved in real devices is much accurate.

Mobile software test creating revolves around the four factors; the first one is the choice between device and emulator. As a general judgment it should be initiated by emulator basically and followed by real environment. The second factor called as test levels covers the user, integrity or system tests. The third is the content test factor in which security, availability and performance tests are available. Testing techniques factor is planned and applied for the adaptation of the software manually, automatically or with script. These factors can be improved and extended (Selvam & Karthikeyani, 2011).

In developed mobile software applications development methods are mostly performance-centered. However, the mobile issues need to be properly set forth software methodology oriented. For this purpose, Hybrid Methodology Design (combined methodology design) is proposed. This methodology is output by taking into account the mobile design software problems of existing methodologies. Mobile D (Abrahamsson, et al., 2004) are examined, agile methods are proposed for development. However, this suggestion is described unscrutinized. It could not propose a specific method for mobile software. It has built its own approachn on the XP method. It takes the life cycle from (Rational Unified Process) and sustainability of the Crystal family of methodologies. Mobile-D presented the components of classic approaches (planning, testing, architectural lines, user-centered approach, the use of agile models ... etc) under one roof.

Hybrid Methodology Design contains 4 approaches. It uses Instantiation , Artifact-oriented, Composition and Integration approaches. It benefits from an existing method. Afterwards, it surrounds the components according to the needs of that model. and continues to benefit from existing process patterns of the method. It completes the development by adding its own ideas, needs and technique. Hybrid Methodology Design 's process is depicted below on figure 5.2.



Our basis: generic software development life cycle

First iteration: elaborating our methodology through general agile practices

Second iteration: utilizing New Product Development approach

Third iteration: instantiating ASD iterative-incremental development engine

The general phases of our final methodology

Figure 5.2 During the iteration of Hybrid Design Methodology process

In the first iteration a risk-based and architect-centered prototype is produced by using the general pattern. The second iteration consists of ideas developed for the market and market tests. In the third iteration the quality based agile methodology takes place. The estimates, cooperation, learning control the whole cycle. Thus, the software lean to the incrementals that can be used again. In the last iteration, composition and integration are performed by taking into account the potential mobile risks. By using Adaptive Software Development (ASD) methodology, the re-usable ideas are transferred to iteration (Rahimian & Ramsin, 2007).

Software engineering differentiates from other engineering fields due to the market needs and rapid technological change. However, these changes have the qualifications repeating themselves. Because the development environments are integrated with each other. In order to overcome this situation through education, Mobile Solutions

Laboratory is recommended. Features such as mobile programming, mobile technologies and mobile commerce are gained to students in universities, nowadays. This need arose because of the gap between the business world and universities. Mobile commerce part is examined as two sided; hardware technology solution (maintenance, security ... etc) and server providers (connected, disconnected and half-connected) solutions. The student's creative and productive and administrative features require additional capabilities when the commerce get involved. They consists of three steps: software, hardware such as mobile network, and social-communicative skills. For this purpose mentioned issues below figure 5.3 are processed in trainings .

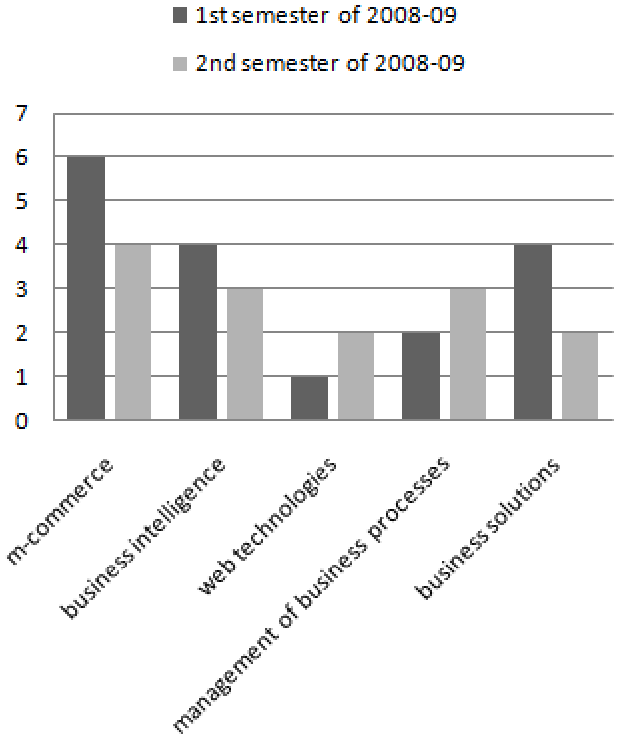


Figure 5.3 The motivation diagram of students in courses

The experience and the information of the students are improved with applications in the laboratory without getting lost in details. By providing financial support to these applications, students were able to study on mobile software project activities which are chosen by the university or student themselves.The solutions regarding mobile software and technology are output. Work discipline reached by the team work. A closer training to the real world of mobile software was ensured. The evaluation criteria were based on as individual and team. Proper management of the project, on-time completion, management of requirements, scientific and technological research, market studies, and software quality are criterias which are taken into consideration. These training are realized in contact with the students by separating the periods. In these periods mobile devices and softwares are examined in all aspects (constraints, capabilities, ... etc) (Hashemi & Vakil, 2009).

mobile software world has become more complex with the development of the open source software. Open source-based mobile software development has to overcome the commercial and environmental problems. Therefore, 3-dimensional viewing angle is introduced (Figure 5.4). Common problems faced by open source development are:

Technical issues

- Dependence on unclear solutions
- Intensive maintenance in roadmap
- To demand different license terms for source code control.
- Changes in the source code
- Different compilations according to the code and version
- Guaranteed quality

Legal issues

- License allowance
- Patent right
- Export rules

The main source of the problems mentioned above, due to the beta nature of open source software. License rights in open source software should be carried by legal experts. Open source software development increases these problems because of the unique pattern of each code and having embedded software. However, in the three-dimensional model some resolutions are proposed to be able to control these issues.

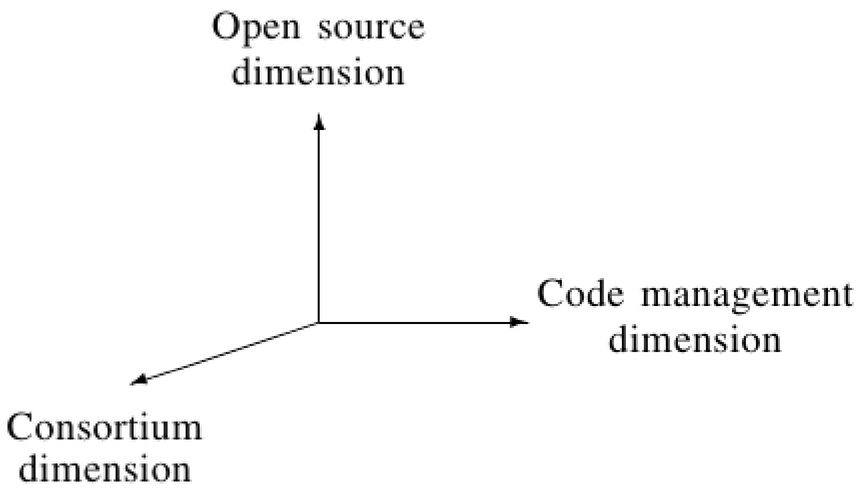


Figure 5.4 A three-dimensional view of the open source model-aware development of large-scale mobile software platforms

In the three-dimensional development model presented in Fig. consortium size take into consideration the business community of mobile software, ecosystem, project management and legal factors. Code management dimension deal with the factors of the source code and the technical infrastructure of the code except source code. The open source dimension social handle social factors and open source management (which version will use the open source? if the version changes will be able to pass the new one? ... etc).

The purpose of the model is to eliminate the open source gap of mobile software emerging from the technical and social aspects through the dimensional approach used. The stated problems are eliminated by using the model between consortium and a roadmap is provided in that way. Of course, after this stage, 3-stage enterprise combination should be provided on the consortium . As can be seen three-dimensional development model primarily aims the management of fast-growing open-source mobile software development platforms before development methodology (Yamakami, 2010).

Performance measurements of mobile software should not be such as traditional software. Mobile software are examined in their own context. Therefore, Scenario-Oriented Performance Evaluation in S.M. Pieper, J.M. Paul, M.J. Schulte, "A New Era of Performance Evaluation", IEEE Computer, Vol. 40, no. 9, September 2007, pp. 23

-30 is proposed. The creation of these scripts should be performed taking into account the characteristics of the mobile world. Scenarios should ensure to meet the criterias including the element comparison of varied applications in terms of usability, versatility, the ability to face critical situations and having visual presentations.

The scenarios should be used not only in virtual, but also in real circumstances. Achieving this will start together by the first prototype. Measurement of the prototype performance should be created as end user-based. Performance measurement criterions are presented in the table below.

Table 5.1 Performance evaluation metrics for mobile software sytems

Evaluation Category	**Performance Evaluation Metrics**
Software Structure	Size of Executable Code Size of Program Memory
Resource Utilization	Throughput Processor Utilization Network Utilization Memory Utilization Battery Utilization
Responsiveness	Waiting Time Processing Time
Dependability	Mean Time to Failure Mean Time to Repair
Productivity	User Friendliness Learnability Ease of Use Maintainability Success Level in User Scenarios Operation Delay Penalty

The criteria specified in the table regards the user-sided software , code size, maintenance sustainability, the software loading time and taking response from server. Two measurements are proposed in this study for the assessment of these criterias. Through measurements the success levels of the scripts and the criterias of errors can be detected. Thus, the effectiveness of next stage for the mobile software can continue being determined with these criterias. (Rahimian & Habibi, 2008)

Users demand for reliable and smooth operation of smart mobile devices. The lost data on smart mobile devices are not responded such as the lost data on PC. Because the data on the mobile device is personal gateway for user which is connected directly to his everyday life. Increasing the required reliability and stability can be achieved by the test which is during the phase of application development. Testing for mobile devices are not easy. Versatile features of the device complicates the tests. Wireless signals, user-end usage and operating systems are primarily examples of the diversity. Thus, the diversity reduce the reusability of tests. Also the limitation of hardware specifications on devices can cause testings not giving accurate results.

Mobile device applications are usually carried out by the manufacturer on their own as one-sided. However, independently developed tests are available. Digia AppTest and TestQuest Pro are two examples. However, these tests are not efficient due to requirements such as user-controlled continuity or environmental conditions that added to the device. In another study, Ichiro Satoh, collected datum which are closer to the actual results when the tests are carried out both by realistic and virtual environments.

MobileTest approach proposed by article is based on the scripts by determining the needs of an independent test. Test conditions are for the cases which are connected to each other and where space and versatility are defined. For example:

- Filling the inbox for volume test
- Receiving SMS text messages while emptying the inbox for the dependent test
- Performing procedures when the battery power is low.

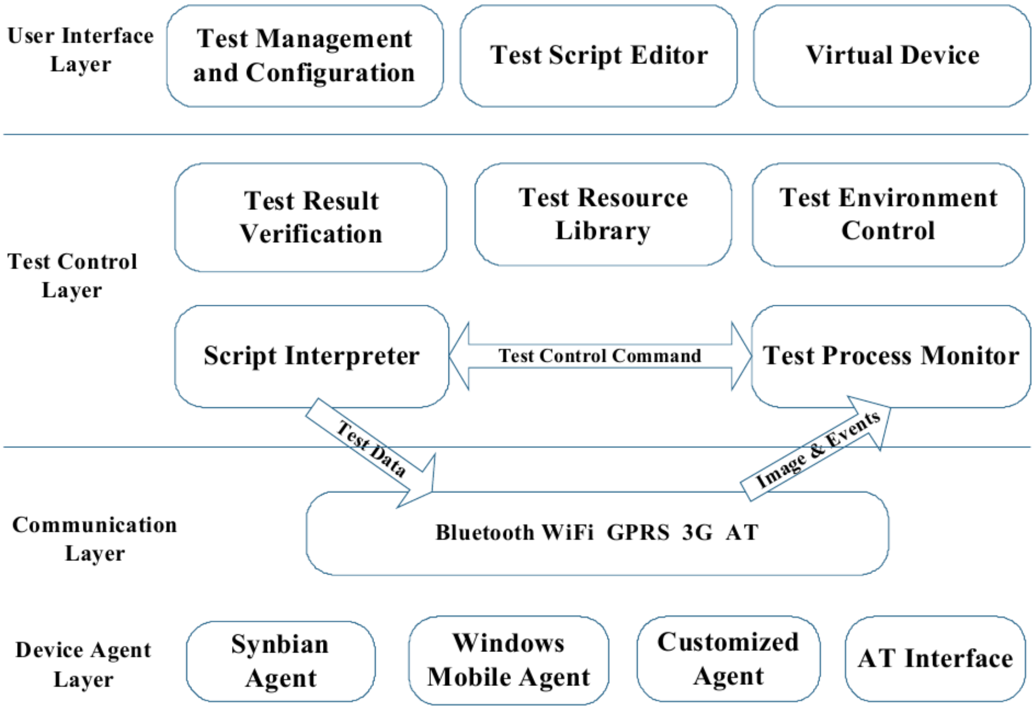


Figure 5.5 MobileTest architecture

MobileTest architecture is defined as 4 layers in figure 5.5.These layers are operated according to the scripts. First, tester determines the target device and the environment. He plan and prepare scripts or utilize virtual devices. Executes them according to the schedule.Outcome results are send to Device Agent. Agent detects critical points and keys. It automatically determines sensitive events and provides the transmission to the following script. Presented test method is sensitive-event based approach. Approach is need to be developed while the complexity of device is getting increasing.The extension packages should be added against the diversity of the wireless signal for the future (Xiaopeng, Xiang, & Bo, 2007).

ADL Con Moto is an approach basing on the simulation of accepted necessary features by modeling functional and non-functional features based on simulation modeling approach. Architecture description language (ADL) approach is taken into account in distributed mobile systems. By means of this language mobile difficulties are addressed during the early times of design so that satisfying designs come out. There is a need for quite a mixed model for non-functional requirements. These are:

- The model should reflect the physical structure completely. Full compatibility of model with the physical connectors and their bandwidth, the relationship with the server ensure compatibility mobility.
- The logical structure should be modeled in detail. Model should take into all components of software and demonstrate the physical dependencies with these components.
- The model should fully reflect the dynamics such as the behavior of logical components, their interactions with each other and exchange of information.
- Finally, the user interaction with the system should be expressed, the number of users and their interactions with the system should be specified.

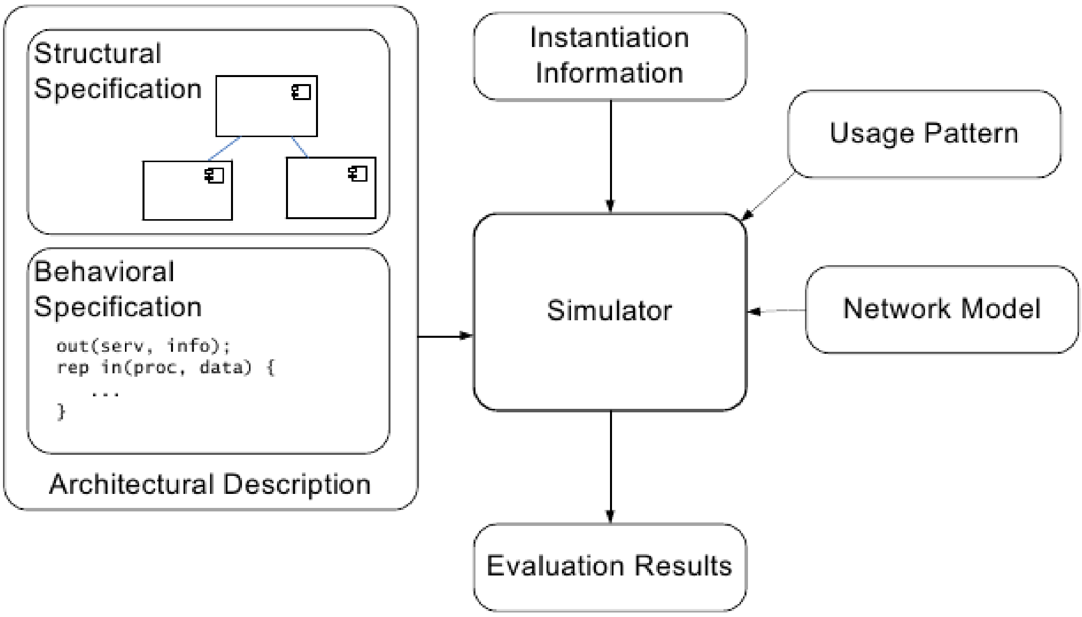


Figure 5.6 Con Moto constituents

The composition of Con Moto shown in the figure 5.6. The simulation of modeling is carried out with all these components. In Behavioral Models listing of dynamic input and outputs of mobile device are ensured. In Structural Model the separation of physical and logical compounds are defined.The problems of resources such as CPU, memory are identified. The same procedure is also applied for connectors. The relationship between the server and client because of mobile nature, the remote changes, mobile agents are tackled. Connecting or not connecting of the components with each other examined and reflected to the model. Finally the components used via architectural connection are embedded in the model in a compatible way with each other. simulation make run modeling through a given scenario. Efficiency is increased with early measures which are obtained from the results in the design phase (Schafer, 2006).

International software engineering training is not created in a real sense. The most crucial requirement is to ensure infrastructure which is capable to achieve it. Schools provide education services to meet the needs of software engineering according to their environment. Thus, the technological education is based on market expectations of that area. Guide to the Software Engineering Body of Knowledge (SWEBOK), an authorized software community, developed the universal principles which served as model for software engineering education. This structure was named Ladder Globalization (GL) structure. It consists of 6 levels. At first, the stages start without curriculum and then, more weight is given to local education. After that universal structure is moved with national documentation. Students who have gone through the stages develop themselves by choosing their universal educators as a free programmer. Constantly changing curriculums are updated by feedbacks obtained from students. Thus, this cycle reaches to a sustainable form with universal educators (Kajko-Mattsson, 2010).

CHAPTER SIX
A MOBILE SOFTWARE ENGINEERING DEVELOPMENT PROCESS

Mobile software development has became more complex because the features of the devices and user interaction has increased a lot. The works and releases up to the present for mobile software show that problems need to be overcame with permanent solutions. The first way to achieve this is to examine the relationship between users and mobile devices thoroughly. Mobile software development should include design which is completely unique to its nature, construction and documentation.

Mobile software development map that we offer will be aimed to the shortages of mobile nature. The published articles seek remedies for mobile solution or put forward a new approach regarding elements such as quality, methods, modeling, training and testing. When we think of all the problems, method come to the fore first.

When we examine the released work, introduced methodologies presents themselves as the copies of existing methodologies. Mobile software development has been headed by taking agile methods in general as a starting point. However, this method can not meet the needs of the mobile software world completely. Mobile applications due to its own nature has more criteria. The use of agile methods has been recommended without explaining the details and step by step. Our methodology was developed considering the criterion of mobile nature. The properties of mobile software methodology we offer are listed as follows:

1. Analysis, design, development, testing and maintenance are indispensable elements.
2. During the analysis phase, user-centered structure should be established while determining the needs. The priorities of needs should be listed within themselves. The physical structure of the mobile device is the most effective element in the determination of these needs. The use of screen comes to the forefront with developed mobile map applications. It starts by determining the sizes of screen. If it is an application that needs to make the safe connection

advantages/disadvantages should be considered. Whether there will a single or multi-application operating system should be determined. As another advantage of this, the license and copyright issues emerge legal arrangement.

3. During the design phase a user-sided development should applied. Due to the constraint of screen, the simplicity of the interface will increase the availability. The designs especially aimed at energy consumption will increase the efficiency.
4. As we mentioned in the development because there are many types of operating system which languages to use should be decided. Due to the screen size, there is an interaction with a single window. For that reason unneeded background programs should be closed. Thus, the memory usage will be saved from unnecessary storage.
5. Tests also should be performed in accordance with real circumstances so that the nature of application are not limited to virtual environment only.
6. the product in the maintenance phase should be ensured not to remain behind the fast-growing mobile world.

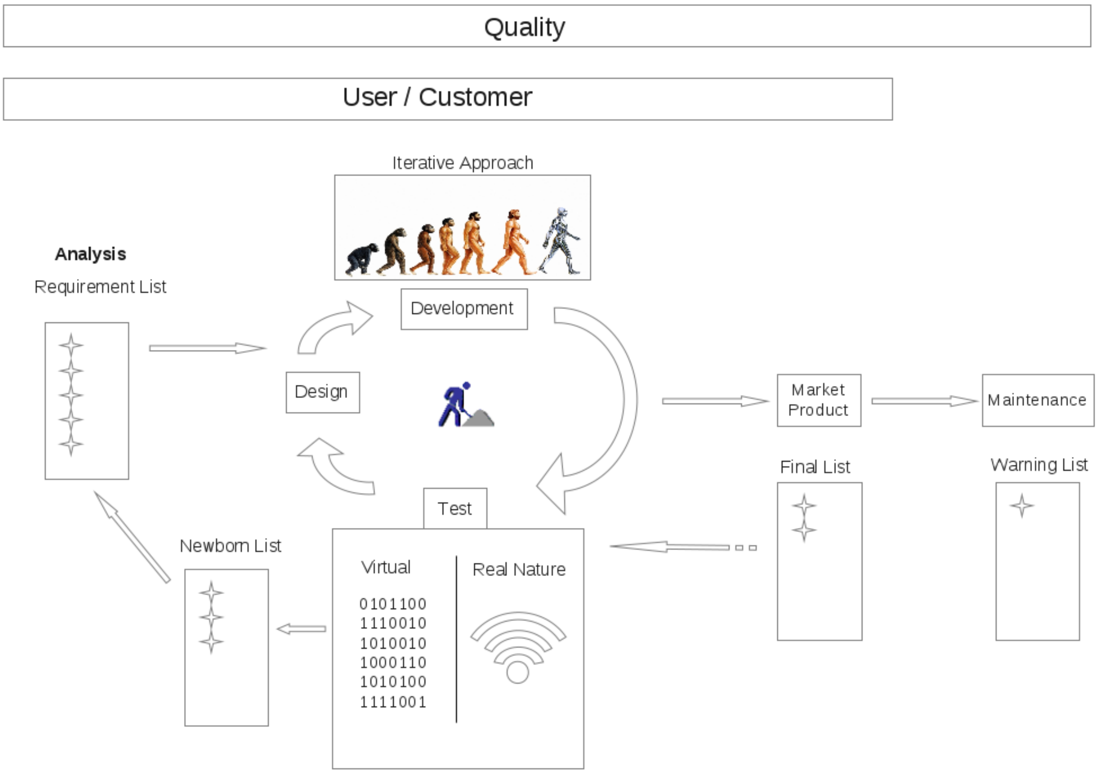


Figure 6.1 Proposed metodoloyg

On the basis of the general elements proposed mobile methodology is depicted in the figure 6.1. As to explain in detail:

After determining the system requirements during analysis, design, development and test cycle begins. In order to have a less re-cycle, it needs to be well analyzed. Needs that can not be determined ahead of time will result in an increase in the consumption of time, labor and money resources. Early period prototype application emerging in design and development stages is tested. In the design and development stages the core structure of the application must be processed. Thus, the basic problems of the process that the application want to achieve is directly dealt with. Therefore, efficient use of resources is ensured. The user interface design is skipped with a more emphasis on kernel code. In the test phase first testings begin in virtual environment. till obtaining first working prototype, virtual tests are continued to apply. Because this is a basic prototype, it should be prepared carefully so that possible technical risks can be reduced. Applied iterative approach ensures the desired results faster. The results of the tests are sent to the list of newborn and a new list of needs is created according to priorities and are continued improving. The virtual and real environments of test are applied during the phases of network connection.

If newborn list can not be formed at the end of tests, the productization is started after completing the list of needs. In the productization developers discuss the compliance of the application to the target audience regarding the views of customers and managers. This stage is the final determination of needs. The final list should not be big to be able to change the size of the kernel code. If this is the case we can conclude that software resources are ill-used the efficient analysis are not realized. After the end of productization stage application is launched to the market. The maintenance activities are carried out according to the feed-backs obtained from the marketed product. The maintenance activities create solutions for the difficulties and provide an example list and serves them to the experience of application developers.

The quality control stage covers the entire application development process including the maintenance. Documentation must be made without delay in order to ensure the permanency of quality, each result out of cycles should be recorded not to make the same mistakes. At the end of day through the meetings between developer teams, the project manager and customer, the required changes are provided with rapid response so that the quality is increased. The customer or the representative carry out the role of a consultant by taking place at every stage of the mobile application development process. Although we stated above standard widely used methodologies are not appropriate to use in mobile applications, advanced approaches such as attention and detail tracking which are applied for the quality should be also developed for mobile methodologies.

Mobile software development process must be unique and have a long-term use. Methodologies for mobile applications such as our proposed methodology should be continued. The pieces of code that are used should be provided as modular and archived. Thus, in the future by re-using of the same modules, resource / time efficiency is increased. The number of code digits in the mobile applications are less than the soft-wares of desktop. The consumption of resources should be reduced by taking into account this feature. The development process is limited to a maximum of 2 weeks. A smaller application means to reach the end in a short period of time. Thus, the basic operating prototypes will be achieved early. In the process of development teams should be classified by a maximum of 2 people. Because decisions within team will be held faster and also the application changes will be faster. When complied with specified periods which is presented in the methodology, the conclusion should be reached within 6 months. Increases in the number of team and schedule of development phases will lead to the unnecessary repetitions, the increase of resource consumption and the deterioration of coordination.

Choosing the right platform to provide ease of code comes to the fore. The point to be aware of in platform is that the manuals should point to the problems which may arise from administrative factors. With these components that need to be considered, the mobile applications which are compatible with multi-platform face with less problem. In today's software development platforms are compatible with all the existing operating systems. Featured operating systems, e is able to develop codes. It is possible to develop software through closed, semi-open or open-source codes in the proposed operating system. In open-source software development platforms coding and compliance issues could create difficulty in the process of developing because of the constant updates. While native applications are increasing the burden of code, web applications lessen the burden of the code. For closed or semi-open sources, legal issues emerge such as the certification. Development platform, operating system or coding language must be selected carefully against the problems.

The advantages and disadvantages of single or multiple operating system of the application developed according to this selection should be analyzed well. Demand for commercial, institutional, or social networking is bound to effect the analysis a lot. For example; the developed applications for versatile platforms directly increase the cost of tests. An application developed for social network should work on all operating systems. The basic factor to be able to overcome the problems is to have a very experienced developer in that environment which increases the efficiency of resources. In addition, the dynamic structures such as redundancy of development platforms, replacement of prototypes, sustainability, speed of teams, trainings, predictions will result in complexity and intensity in implementation of the methodology. However, the use of existing platforms for mobile applications to be developed should be encouraged. Thus, a sustainable, grounded, reusable development environment is created.

Being able to have more experience as developers is based on the education which entails continuous and quality basic training. Mobile application developers should be provided to follow the up to date changes to keep their continuous education. Software firms should sent them to the certificate programs to catch the speed of rapid growing mobile software technologies. In addition, the training of developers should not be formed by the region of their schools where they have graduated. Schools should gather the universal mobile software engineering rules under one roof. Because the software engineering practices based on the regional needs leads standardization problems. The universal features that mobile application developers should have are as follows:

- To develop the project in group dynamics.
- To have knowledge of different mobile application development platforms, and development tools.
- To use the resources effectively in the commercial sense.
- To be able use web services in accordance with mobile structure and combine creative applications in architectural design.
- To be able to design user interface convenient for mobile devices.
- To be able to fulfill the rules and format of the mobile methodology to be used. (to be able to adhere to the methodology.)
- To comply with the delivery time specified in the planning of the project.

Digital Living Network Alliance (DLNA) is a structure established by the industry's leading consumer electronics, mobile and personal computer companies. With this structure, the products manufactured by different companies are aimed to work in harmony with each other within the home network.The same structure should be established for the mobile software engineering. Right of access to documents should be facilitated by providing continuing education system. Because the procedures are documented in mobile application development process, the resources will be composed for learning processes. After this step, the most important issue is to be able to obtain an universal education from these experiences which ensures not to perform the same mobile software engineering mistakes.

The tests used for mobile application are carried out by the virtual platforms offered by operating system or platforms. The mobile applications only running in the desktop environment are difficult to determine the actual performance. Because, there are no restrictions of the desktop devices regarding hardware. The applications which are developed similarly to the ones that we proposed in our testing phase should also be carried out in real environment. The test scenario should be created by installing a mobile device. The most crucial expectation from our personal mobile devices is to have constant network connection scenarios. The majority of mobile applications are operating in connected way to a network. This network can be home network, Internet or corporate network. To do this, the application of the mobile device should be subjected to the scenarios such as having network connection, no network connection or multiple network connection methods. The target is to ensure the fulfillment of task for the related application in accordance with the continuity of the connection. Depending on the criticality of developed application the ease of use for the interface of device (such as social networks), battery depletion status (map apps, etc.) and contstantly developing sensors (GPS, pedometer, etc.) complicate and increase the number of emerging scenarios. The test features in the development process of mobile applications are:

- Interaction of application components need to be good with each other. (such as receiving massage while writing text in smart phones)
- A time scale and sequence need to be established for test scenarios.
- Tests should be developed or adapted in a way to include all mobile devices.
- Memory management of tests in devices should progress in stages in a flexible way.
- It should be continued until the test results become accurate.
- Tests should be presented in operation, and related test script should be shown on the screen.
- The tests need to be applied in different environments, definitely.

As we stated before, the test criteria between native, J2ME and web applications should have differences (Table 6.1). The mobile application ecosystem is as shown in the table. Native applications use a lot of more in-depth software and hardware resources of device during operation. The mobile applications running on J2ME use also the resources of device, but the life cycle is short. Because it is a non-preferred platform for smart phones. Web application adds the server element that makes most of the work to these components. The server issue for mobile applications to be developed in the scenarios should not be missed. The interface designs of web application sites should be designed according to the features of mobile device. So, easy to use and simple interfaces which relieves unnecessary site installation should be presented. All desired mobile application features shall not be able to develop efficient applications without proceeding from their true nature. To have early tests which are ready to apply increase the accuracy. So catching the errors beforehand will make the changes easier to perform.

Table 6.1 Mobile Ecosystem

Native	Web	J2ME
• Processes running on back stage • Develop for OS • Complex dispersion/test • Deepest Using mobile device sources	• Limited use mobile device sources • Run on servers • Working on almost every devices • Unadapted web site versions	• Using mobile device sources • Coherent Run • Poor Lifecycle • Works on most non-smart phones

The development in mobile applications according to the user criteria and requirements comes to the fore. The presence of customer at every stage of development ensures not to miss human factor during the development phase. However, the customer is not the only factor to consider. Because mobile applications surround us in terms of information-sharing, sensors and social links. Humans should be used in the development processes. Users' interactions with existing applications should be observed and learned. Research questions should be defined by interacting with the user for whom mobile application is developed. The desired design will be reached more easily through the obtained responses. The areas of interest can be more easily understood by the observations. Thus, inspired designs will be able to be developed to the new applications. This characteristic of mobile software engineering also can be ensured by a mobile developer. Documentation of the observations should not be forgotten via taking notes. Even developers can increase the diversity of ideas that comes to the mind by keeping notes in everywhere mobility can reach.

Today's mobile devices have the most anticipated feedbacks regarding updates. By gathering the demands of user in the maintenance phase for a smart phone or enterprise, the high expectations should be satisfied. The fast and flexible updates should be performed.They should not cause a lot of work by providing automatizing. Since critical updates can push away the user, updates should be focused on quality. Having a critical update should not make the consumers thought that applied development process of the product in marketing stage is inefficient. The changes to be brought as a result of updates should not change the standards. Harmonization of the standards should be provided within the application. The standardization should be evaluated by administration in terms of resources and decided whether an update or a new mobile application is required. For example, in multi-faceted platforms, the maintenance costs of software application will be relatively more regarding single platform. However, the classification of the returns will be easy due to the large number of users. Thus, corporate decisions are obtained easily

Number of mobile applications is increasing rapidly. The main reason for this increase is the commercial demand. The increase in commercial demand lowers the quality of the mobile applications . The reason of this poor quality is introducing the product to the market quickly with missing test and careless analysis. The majority of mobile applications are composed of entertainment and game sectors. Hundreds or thousands of applications are accessible on the mobile application shops that are developed through operating systems of smart phones, tablet computers. Applications in the shops are premium or free of charge. In this case reliability problems arise. The rigidity of the certificate is increasing the reliability in closed-source software development, on the other hand safety control problems emerge in semi-open or open-source software development. The majority of mobile applications are far from professional mobile application development. As we said before, commercial concerns control the development processes. In addition, these concerns create a competition among companies. Thus, rather than project-based, product-based mobile application development should be started. The emerging commercial concerns force the mobile development processes to be product-oriented structure. It is certain that the winners of this competition are the companies or developers implementing the methodology which can comprehend the mobile world with an approach preserving the quality and speed.

CHAPTER SEVEN
CONCLUSION & FUTURE WORK

Variety of mobile devices are increasing day by day. Today's mobile software engineering is developing with the market concerns. Although there are mobile software development methodologies in the market, they are imitations of the traditional methods. The details are not described in the mobile methods.

In our methodology, the special structure of mobile software engineering stands out. The quality is improved with the generated lists. Tests which carried out both in real and virtual environments provide early detection of problems. The early prototype test scenarios can be performed quickly. Also, at the productization phase, the market concern is reduced to a minimum level. It is made no concessions on the quality of the product. In the maintenance phase, the experiences transferred to the future mobile softwares by the list generated again.

The future of mobile applications is obvious to become complex due to the complexity of the devices. There is a rapid technological development in terms of technical equipment. Particularly the sensors attached to mobile devices enhance their abilities. A variety of methods for the network connection for today's smart phones and tablet computers are presented. The abundance of these methods are indicative of a variety of methods for the network connection in the future. Devices are getting smaller. Smart phone type products of which screens operated through glasses are started to be developed and marketed. Even mobile device scan transfer the users' health information (blood sugar, blood pressure, etc ...) to their doctor. The mobile devices with touchscreen feature have come to the fore, recently. This feature has been studied further to provide touch feature to the near objects. In addition, Touch screens are being developed to detect two different user.

The complexity in terms of software is more prevailing by the developments of hardware. The commercial concerns of applications in development processes are not too much. In the future applications these concerns will increase further. Large application developers go to law against each other such as patent lawsuits. When we look at the future of forthcoming mobile software technology, cloud computing starts to get into the mobile applications. Mobile platforms meet the needs of hardware and software of the period with rapid updates.

Abundant products are being released on the market in terms of variety of mobile devices. Mobile operating systems have control over the recently released cameras. Thus, the devices gain the ability to make processing at the time the photo taken. Besides, the products such as laptop are tried to be implemented with tablet computer features by designing attachable screens and touch screens. So it is obvious that the mobility dominates the market and users a lot.

A challenging future awaits mobile software engineering. Regarding observations and future predictions, the number of mobile device user is growing increasingly. The devices such as desktop or laptop are preferred only at home or office. Future mobile software methodologies to be developed will achieve to keep up this speed through the collaborative organization of developers and educators. If organized approaches can not be established, the quality will not be able to sustained for the demands of mobility ruling our everyday affairs. In our model we deal with the issues which are passed without given enough details to improve the quality. Thus it introduces a distinctive approach to the mobile software engineering.

REFERENCES

Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme, T., Jäälinoja, J., Korkala, M., Salo, O. (2004). Mobile-D: An Agile Approach for Mobile Application Development. *19th Annual ACM Conference on Object-Oriented Programming, Systems, Languages and Applications* (pp. 174-175). Association for Computing Machinery.

Abrahamsson, P., Salo, O., & Ronkainen, J. (2002). Agile Software Development Methods Review and Analysis. Finland: VTT Electronics.

Android. (2013). *Android Developer*. Retrieved from Android: http://developer.android.com

Apple. (2013). *Apple Developer*. Retrieved from Apple: http://developer.apple.com/

Authority, K. T. (2012). *2012 Mobile User Survey*. Keynote Competitive Research.

Beck, K. (1999). XP Principles. In K. ..-W. [Beck, & 1.], *Extreme Programming Explained Embrace Change* (pp. 47-48). Addison Wesley.

Beck, K. (2001). *Manifesto for Agile Software Development*. Retrieved from http://agilemanifesto.org/iso/en/

Beck, K. (2001). *Principles behind the Agile Manifesto*. Retrieved from http://agilemanifesto.org/principles.html

Bouchard, L. (2010). Multimedia Software for Mobile Phones. *IEEE*.

Burnette, E. (2010). *Hello, Android Introducing Google's Mobile Development Platform, 3rd Edition.* The Pragmatic Bookshelf.

Ezzouhairi, A., Quintero, A., & Pierre, S. (2005). IP configuration in Ad Hoc Networks. *IEEE* .

Fetaji, B., & Fetaji, M. (2009). Mobile Learning Software Solution Using Software Engineering Task Based Learning Approach, . *World Academy of Science, Engineering and Technology*, 54 .

Firtman, M. (2010). Mobile Device Categories. In *Programming the Mobile Web* (pp. 6-11). O'Reilly Media.

Firtman, M. (2010). Programming the Mobile Web. Sebastopol: O'Reilly Media.

Fling, B. (2009). Mobile Design and Development. Sebastopol: O'Reilly Media.

Fling, B. (2009). The Mobile Ecosystem. In *Mobile Design and Development* (pp. 20-27). O'Reilly Media.

García Villalba, L., Sandoval Orozco, A., García Matesanz, J., & Márquez Díaz, J. (2011). Auto-Configuration Protocols in Mobile Ad Hoc Networks Sensors].

Hammershøj, A., Sapuppo, A., & Tadayoni, R. (2010). Challenges for Mobile Application Development. *Center for Communication, Media and Information Technologies (CMI) IEEE.*

Hashemi, A., & Vakil, N. (2009). A Practical Course on Mobile Software Engineering, Mobile Solutions Laboratory. *Fourth IEEE International Conference on Software Engineering Advances.* IEEE.

HP. (2012). *Five Mobile Application Challenges for IT Teams.* HP Business White Paper.

Kajko-Mattsson, M. (2010). Towards a Mobile Software Engineering Education. *Information Society (i-Society), 2010 International Conference on* (pp. 529-535). IEEE.

Kamthan, P. (2010). Using Patterns for Engineering High-Quality Mobile Applications. In *Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization* (pp. 174 - 195). Canada: Concordia University.

Lutes, K. (2004). Software Development for Mobile Computers. *The IEEE CS and IEEE ComSoc*.

Mishra, J., & Mohanty, A. (2011). *Software Engineering.* India: Pearson Education.

Pettey, C., & Van der Meulen, R. (2011). *Gartner Says Android to Command Nearly Half of Worldwide Smartphone Operating System Market by Year-End 2012*. Retrieved from Gartner Technology Research and Advisory Company: http://www.gartner.com/newsroom/id/1622614

Prathaban, V. (2004). 3G. *Malaysian Business.*

Rahimian, V., & Habibi, J. (2008). *Performance Evaluation of Mobile Software Systems: Challenges for a Software Engineer.* 5th International Conference on Electrical Engineering Computing Science and Automatic Control IEEE.

Rahimian, V., & Ramsin, R. (2007). Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach. *The Research Vice-Presidency of Sharif University of Technology*.

Roman, G.-C., & Murphy, A. (2002). Guest Editorial: Software Engineering for Mobility. In *Automated Software Engineering* (pp. 9,135-136). Kluwer Academic Publishers.

Salmre, I. (2005). *Writing Mobile Code Essential Software Engineering for Building Mobile Applications.* Addison Wesley Professional.

Schafer, C. (2006). Modeling and Analyzing Mobile Software Architectures. In *EWSA Software Architecture: Third European Workshop* (pp. 175-188). Springer.

Schwaber, K., & Sutherland, J. (2011). The Scrum Guide Developed.

Selvam , R., & Karthikeyani, V. (2011). Mobile Software Testing - Automated Test Case Design Strategies. *International Journal on Computer Science and Engineering (IJCSE)*.

Somerville, M., Luo, C., Lansdell, J., Morris, B., & Bortenschlager, M. (2010). A Software Engineering Model for Mobile App Development. In *Introduction to bada A Developer's Guide* (pp. 463-470). Wiley Publishing.

Taub, E. (2010). Hints on how to curb power drain: Great features can mean much shorter operating span for your new toy. *International Herald Tribune*.

Wagner, R. (2008). *Professional iPhone and iPod touch Programming:Building Applications for Mobile Safari.* Wiley Publishing.

Wasserman, A., & Mellon, C. (2010). Software Engineering Issues for Mobile Application Development. *Silicon Valley*.

Wehbi, B. (2005). Address Autoconfiguration in Ad Hoc Networks ;]. *Institut National des Telecommunications INT Departement Logiciels Reseaux LOR Internal Report*.

Xiaopeng, G., Xiang, L., & Bo, J. (2007). MobileTest: A Tool Supporting Automatic Black Box Test for Software on Smart Mobile Devices. *Second International Workshop on Automation of Software Test (AST'07.* IEEE.

Yamakami, T. (2010). A Three-dimensional View Model of Open Source-aware Software Development for Large-scale Mobile Software Platforms. *4th IEEE International Conference on on Digital Ecosystems and Technologies.* IEEE.