A Comparative Study of Software Engineering Approaches in Mobile Application Development: Native, Web, and Hybrid

Sadeem Alyahya
Software Developer –
Ministry of justice
Riyadh, Saudi Arabia

Hadeel S. Alrewaished
King Abdullah bin Abdullaziz Public Education -
Development Project - Ministry of Education
Riyadh, Saudi Arabia

ABSTRACT

The shift from using personal computers to using smart phones to perform tasks and obtain information has motivated the software engineering community to investigate many issues related to the development of mobile applications. This paper provides an overview of the current directions in research on these issues including development of lifecycles, portability, processes, development tools, and testing.

General Terms

Software Engineering, Software Development, Algorithms, Design, Testing and Maintenance

Keywords

Mobile applications, software engineering, mobile programming environments, mobile application testing

1. INTRODUCTION

Smartphones are a handheld device integrated with an operating system capable of performing a range of different features including global positioning sensors, wireless connectivity, photo/video capabilities, built-in web browsers, voice recognition, advanced computing capability and a multitouch screen. These features support the proliferation of mobile applications [1,2]. Flora defines Mobile Application Development (MAD) as the process by which applications are developed for small low-power handheld devices [3].

The statistical study by Gartner Group shows that smartphones is a vital necessity of our lives. Nowadays almost everyone owns his/her personal phone not only for messaging and calling purposes but also for entertainment, communication over the web, complex business transactions, instant accessibility of required information and much more. The anticipation that projected no. of mobile apps to be downloaded in 2012 is 36 billion i.e. 5 app. per person over the globe. Another study exemplifies that 428 million mobile devices were sold worldwide in 2011 Q1, a 19 % increase from the previous year [4].

Mobile applications are not a new idea, and various versions and platforms have been around for some time.

Mobile application development portability is categorized into three categories: native, web and hybrid mobile applications.

There are several systems development methodologies including traditional and agile methodologies being utilized in current systems development.

To choose the suitable tool for developing a mobile application we must choose the suitable approach native, cross-platform or hybrid.

Mobile application testing is evaluating the difference between user expected and the actual results. This makes test crucial to check all the features before the application is delivered to the end users. The good user interface achieves maximum effectively, reduces the learning time of users, and improves efficiency and satisfaction. So, usability test is a necessary process to test how the application can be used. In this paper, we will shed light on the usability test in both design and test development layers. To ensure application products could meet this quality, there are several usability attributes and diminutions have been determined. Also, the paper involves some major issues that need to know before designing a mobile applications interface. The paper will describe usability life cycle, the early stage of usability in the design layer [5], usability methods and a Betio and Cybi's [6] study that compares the results of usability test of mobile interfaces using three different approaches.

The remaining sections of this paper are organized as follows. Section 2 gives an overview of history of mobile applications. Section 3 gives an overview of the portability issues of mobile application development. Section 4 gives an overview of the development lifecycle of mobile application development. Section 5 gives an overview of the mobile application development tools. Section 6 gives an overview of testing the mobile applications

2. HISTORY OF MOBILE APPLICATIONS

Various versions and platforms for mobile have been around for some time. Table 1 shows timeline for the development of applications on mobile devices.

Table 1. MOBILE DEVICE APPLICATION ERAS

Era	Timeframe	Characteristics
Candy Bar phones	1988-1998	Very limited data services to phone
Phones supporting WAP	1998-2002	WML, MMS/SMS, small screens, etc.
Smart phone	2002- present	Java-ME, Windows Mobile, Palm OS, RIM, etc.
Touch screen era	2007- present	iPhone, Android, etc.

"Candy bar" often shaped like a candy bar (and about the size of a large candy bar), applications in this era were targeted to specific phones and very rudimentary. Wireless application protocol (WAP) based phones can support wireless markup language (WML), a subset of HTML, but it was still difficult to deploy applications. Smart phones tend to have much tighter control over the applications that are provided and are focused on messaging and personal information management (PIM) features such as calendaring and contacts. Touch-screen phones are based on the application concept. Such devices are viewed as general computing devices, messaging and PIM features are just applications rather than core functionality [7]. Touch-screen phones have special markets for applications. iPhone AppStore opened in July 2008. Since then, device makers have created outlets for other mobile devices, including Android, BlackBerry, Nokia Ovi, Windows Phone, and more [8].

3. PORTABILITY ISSUES

Mobile application development platforms are categorized into three categories: native, web and hybrid mobile applications and each one have advantages and disadvantages [2,19].

3.1 Native application

Native applications have been developed with a single code for all devices and are specific to the mobile operating system. Single projects are developed for each operating system which leads to a raise in development team, cost and time. If you are working in an environment where there is no internet connectivity, native apps are the best way to go for it. Native applications work with a built-in feature and thus they are easier, faster to work on the device and platform independently. Maintenance is one of them which is a difficult task for both users and developers. Developers must develop the application according to different platforms whereas the users must upgrade it regularly. Another disadvantage is the development cost when you are making applications for different platforms [2,19].

3.2 Web application

Web applications work on desktop computers as well as on mobiles. A user does not require going to mobile application store to upgrade or download the apps. Thus, whenever the users log in, they will acquire the upgraded version. Developers also do not require to worry about the mobile platform because there will be a single ubiquitous version which can be used by any of the mobile platforms. That's why the maintenance cost of the web application is low. Internet connectivity must be there in web applications. Web applications are not coherent with smart phone features like GPS, Camera, and Phone dialing etc. Web applications even cannot be listed in play stores. If user wants to search for the application, then they must search on web. Performance wise web applications are slow compared to native apps. If user wants to use the web application on regular basis, then they have to mark it as bookmark. Without bookmarks, it is difficult to use web applications. No application icon is there on user's device because it is a web link which can be open as per requirement [2.19].

3.3 Hybrid application

Please Hybrid application combines variant technologies from native and mobile Web application to obtain the benefits of each technology. Applications are installed from a web store and have right of entry to device specific features as available in native application, but application is developed using web application tools that's why they act correctly like a native application. Hybrid applications are not executed natively, that's why execution of the application becomes slower. Hybrid application development is new technique so there are fewer tools available for the development of hybrid applications. Applications with complex native functionality or bulky transitions, such as 3D games have sometimes performance issues [2,19].

Table 2 below describes comparison between the three [4].

Table 2. Mobile Application Development Approach Comparison

Features	Native	Hybrid	Web
Development Language	Native only	Native and Web or Web only	Web only
Device Access	Full	Full	Partial
Speed	Very fast	Native speed	Fast
Development Cost	Expensive	Reasonable	Reasonable
Advanced Graphic	High	Moderate	Moderate
Upgrade Flexibility	Low	Moderate	High
Some Example Apps	Angry Birds Instagram	eBay, PayPal	http://m.facebook.com

Choosing the optimal approach from the three above depends on the needs of the organization and can be driven by many criteria including finances, project timeline, resources, target users, application features, and IT infrastructure [2,19].

With so much interest in availability of applications for There is an apparent problem facing developers in requiring the ability to have their applications available to the widest possible audience but lack resources to build natively for each platform. To further the reach of applications, without expending the considerable amount of time needed to learn the quirks of each platform, using cross-platform development tools (CPDTs) may be the answer. However, there is no adequate measure to ensure these CPDTs are as capable as native development tools. This leaves developers unsure if they can make use of these tools to create strong applications with native functionality and performance [9].

Dhillon and Mahmoud have researched five tools as shown in Table 3 below. These tools were chosen because of their flexibility, feature support, and popularity among developers. The chosen tools provide samples of different approaches to cross-platform development with cross-compiled, runtime, and Web wrapper styles. Mobile Web browser-based applications are included for comparative purposes [9].

Table 3. Tools for Cross-Platform Application Development

CPDT	iOS	Android	Windows Phone 7
Mobile Web	√	√	√
HTML5 Working Draft	√	√	√
Adobe PhoneGap	√	√	√
Appcelerator	√	√	√
Titanium	√	√	√
Adobe Air FLEX	√	√	
MoSync	√	√	√

3.4 LIFECYCLE DEVELOPMENT OF MOBILE APPLICATION

The classification of development methodologies in this paper is based on the classification by Blum (1996), Highsmith (1999), Krutchen (2001), and Cao and Ramesh (2007), which classifies the development methodology into heavyweight and lightweight/agile. Heavy-weight methodology commonly refers to traditional plan-based methodology while lightweight or agile is an emerging methodology [11].

The heavyweight /traditional development methodologies have strengths which are [11]:

1. Predictive methodology, the methodology used can be simply utilized as the project is conducted sequentially.

2. There is not much interaction with the customers in this methodology, the process could thus move faster.

3. The project domain is predictable thus it can be completed easier.

The heavyweight /traditional development methodologies have weaknesses which are [11]: 1.

Does not accommodate the rapid changes of requirements.

2. Lots of documentation.

3. Gaps of work (Less test job at the requirement gathering phase).

4. Return on investment is at the end of the project, which can be too long.

5. Success measurement is conformation to plan which may contribute to less business value.

The lightweight/agile development methodologies have strengths which are [11]:

1. Adaptive process, the work process can be adapted according to the project domain and requirements.

2. Less documentation.

3. The applications can be delivered to the market early; thus, return on investment is early in the project.

4. Well, accommodate to the rapid change of mobile technology and variety of mobile operating platforms.

5. Promote knowledge sharing.

6. Success measurement is business value.

The lightweight/agile development methodologies have weaknesses which are [11]:

1. Unpredictable domain and methodology and thus require more work with more effort.

2. As customer feedback is essential, thus more development time is required.

3. The nature of fast development for each iteration could sometimes cause insufficient software testing, which results in usage problems on some mobile device models. Henceforth, the required applications may be delivered in too many patches/versions which are not good.

Table below differentiates between the two methodologies:

Table 4. Comparison between Traditional and Agile Lifecycle

Lifecycle Factors	Traditional development Agile development	Traditional development Agile development
Environment	Stable environment	Dynamic environment
Values	Value planning, predictability, high assurance, and control.	Value interaction, collaboration, and adaptability.
Change	Unchanged Change is unavoidable and should be	Unchanged Change is unavoidable and should be embraced
Implementation	Long process development life cycle, complete systems are implemented at the end of the process.	Incremental (small software releases with rapid development cycles)

The table above could identify the appropriate development methodology framework for mobile application development which is agile methodology. Agile methodology provides practices that facilitate communication between the developer and the customer, and undergo develop-delivering feedback cycles, to have more specific view of the requirements, and be ready for any change at any time. Abrahamsson et al. have demonstrated the traits of which reasons why agile technologies fit best in mobile software development. The various issues include high environment volatility, small development teams, identifiable customer, object-oriented development environment, non-safety critical software, application-level software, small systems and short development cycles [12].

An appropriate agile method could be selected for a given project and can be tailored to a specific requirement based upon project's complexity and team size.

Table 5. Mobile Application Development Processes using

Mobile Process	Description	Year	Technique
Mobile D	Recommended use by a small co-located team of at most ten co-located developers, working in a short development cycle towards product delivery within 8 to 10 weeks of calendar time. Although this methodology being a pioneering study.	2004	XP, Crystal, RU P
RaPiD 7	Supports all software development projects, whether related or unrelated to mobile	2005	AM

	application development, used in major mobile companies such as Nokia and Philips.		
Hybrid Methodology Design	The methodology is still at a high level, and no specific tasks for the identified stages have been provided. The published material on Hybrid Engineering Methodology does not include any case study or shows that the methodology has been empirically tested on developing an actual mobile software product.	2007	ASD, NPD
MASAM	Recommended use for small companies that are focused on the development of mobile software applications	2008	XP, RUP, SP EM
SLeSS	Used in real embedded software customization development projects for mobile phones with team of 7-12 developers, in duration from 4-6 months, with average.	2011	Srum, Learn Six Sigma

The Mobile-D approach is based on Extreme Programming (development practices), Crystal methodologies (method scalability), and Rational Unified Process (life-cycle coverage). The Mobile-D approach is optimized for a team of less than ten developers working in a co-located office space aiming at delivering a fully functional mobile application in a short time frame (i.e., less than 10 weeks). Mobile-D has been developed in co-operation with three companies developing mobile software products and services. The approach has been successfully assessed against the CMMI level 2 certification.

A development project, following the Mobile-D approach, is divided into five iterations. These phrases are set-up, core, core2, stabilize, and wrap-up. Each phase consists of three different types of development days: Planning Day, Working Day, and Release Day. If multiple teams are concurrently developing different parts of the same product, an Integration Day is also needed. To ensure proper operation of the product on multiple mobile platforms as early as possible, to enhance software design, and to improve software changeability, the development is required to be testing oriented. The Mobile Test-Driven Development (MTTD) practice employed in Mobile-D involves writing tests before actual implementation, automating unit testing procedures, and acceptance testing all features with the customer. When applying the Mobile-D approach in practice the following positive observations have been made increased progress visibility, early identification and solving technical problems (such as the difficulties with the J2ME and Symbian platforms), shared responsibility, efficient information sharing, high process practice coherence, low defect density in released products, and constant development rhythm. On the other hand, the following identified challenges will guide our further research efforts:

1. High requirements for platform-specific development skills.
2. Variability and portability problems arise due to the Diversity of characteristics in different mobile terminals.

3. Weak tool support for Test-Driven Development in mobile environment.

4. Adapting rapid release cycles (management, customers). 5. Need for highly disciplined practices (process fidelity). 6. Attempts to use agile methods out-of-box lead to problems [12].

4. MOBILE APPLICATION DEVELOPMENT PROCESSES

Building a mobile application follows a similar set of four fundamental phases: planning, analysis, design, and implementation (Fig 1). Each phase is composed of a series of steps, which rely on techniques that produce deliverables.

out. Figure captions should be centered beneath the image or picture, and Table captions should be centered above the table body.

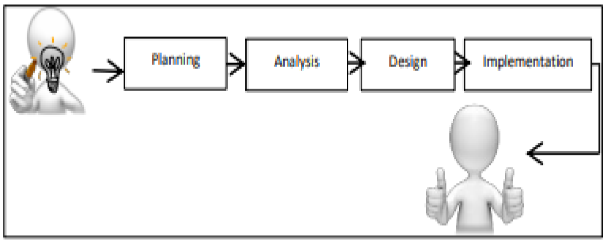

Fig 1: Fundamental phases of building information system

4.1 Planning phase

The planning phase is the fundamental process of understanding why an information system should be built and determining how the project team will go about building it. It has two steps:

4.1.1

A system request and feasibility analysis. A system request presents a summary of a business need, and it explains how a system that supports the need will create business value. The feasibility analysis examines key aspects of the proposed which are technical feasibility, economic feasibility, and organizational feasibility.

4.1.2

Once the project is approved, it enters project management. The delivery for project management is a project plan that describes how the project team will go about developing the system. The project plan consists of work plan, staffing plan, standards list and risk assessment.

4.2 Analysis phase

During this phase, the project team investigates any current system(s), identifies improvement opportunities, and develops a concept for the new system. This phase has three steps:

4.2.1

An analysis strategy is developed to guide the project team's efforts. Such a strategy usually includes a study of the current system (called the as-is system) and its problems and envisioning ways to design a new system (called the to-be system).

4.2.2

A requirement gathering (e.g., through interviews, group workshops, or questionnaires). The system concept is then used as a basis to develop a set of business analysis models that describes how the business will operate if the new system were developed. The set typically includes use case analysis and models that represent the data and processes necessary to support the underlying business process.

4.2.3

The analyses, system concepts, and models are combined into a document called the system proposal, which is presented to the project sponsor and other key decision makers who will decide whether the project should continue to move forward.

4.3 Design phase

The design phase decides how the system will operate in terms of the hardware, software, and network infrastructure that will be in place; the user interface, forms, and reports that will be used; and the specific programs, databases, and files that will be needed. The design phase has four steps:

4.3.1

The design strategy must be determined. This clarifies whether the system will be developed by the company's own programmers, whether its development will be outsourced to another firm (usually a consulting firm), or whether the company will buy an existing software package.

4.3.2

This leads to the development of the basic architecture design for the system that describes the hardware, software, and network infrastructure that will be used.

4.3.3

The database and file specifications are developed. These define exactly what data will be stored and where they will be stored.

4.3.4

The analyst team develops the program design, which defines the programs that need to be written and exactly what each program will do.

4.4 Implementation phase

The system is built (or purchased, in the case of a packaged software design and installed). This phase has three steps:

4.4.1

System construction is the first step. Most organizations spend more time and attention on testing than on writing the programs in the first place.

4.4.2

The system is installed. Installation is the process by which the old system is turned off and the new one is turned on. One of the most important aspects of conversion is the training plan, used to teach users how to use the new system and help manage the changes caused by the new system.

4.4.3

The analyst team established a support plan for the system. This plan usually includes a formal or informal post-implementation review, as well as a systematic way for identifying major and minor changes needed for the system.

5. MOBILE APPLICATION DEVELOPMENT TOOLS

There are three development approaches it must take care of which is native, cross-platform and hybrid. The discuss her three examples of developing the application using native speakers which are iOS, Android and Windows Phone. The other approach is cross-platform and the given overview about PhoneGap which is one of the popular cross-platform frameworks. Hybrid development combines some features of native and cross platform applications.

5.1 Native development

5.1.1 iOS

To develop an application for iPhone, ipad, and iPod touch, you will require Mac computer running OS X 10.5.8 or later is needed. iPhone SDK furnished by Apple includes IDE of Xcode, iPhone simulator and a suite of excerpt tools for the development of an application. App Store supply user facility to search and download applications which are developed by iOS SDK [7,2,19].

5.1.2 Android

Android OS is released by open-source Apache license and built on Linux. Android applications can be built using Linux, windows, or Mac and java is primary language for development of Android applications. Android does not support J2ME, and its most ordinarily used is Eclipse and Android Studio. Developers can create native libraries in C or C++. Google play store is the official site and portal for searching and downloading android application [7,2,19].

5.1.3 Windows Mobile

Windows mobile OS gives more desktop like user experience. In addition to C# or C++, Silverlight and Microsoft XNA game studio are used for game and application development. For the development of windows, mobile applications required tools are windows mobile SDK, visual studio.NET framework and developer toolkit [7,2,19].

5.2 Cross-platform development

5.2.1 PhoneGap

It is essential that developers choose software that provides an ordinary development track across variant platforms when application is developed with cross-platform mobile frameworks. In this title, It talk over briefly on some of the popular cross-platform frameworks. Adobe's PhoneGap platform enables a developer to create an app that runs on a variety of mobile devices. The developer accomplishes this largely by writing the user interface portion of their application with Web technologies such as HTML, CSS, and JavaScript. PhoneGap's development tools then bundle the HTML, CSS, and JavaScript files into platform-specific deployment packages. PhoneGap supports a wide variety of platforms such as iOS, Android, Windows 8, Blackberry, WebOS, Symbian, and Tizen. PhoneGap essentially wraps a Web view of your HTML, CSS, and JavaScript in a native application. This is required because the Web view in an application does not inherently support many device features, such as access to the file system or the camera. PhoneGap has a bridging mechanism that allows JavaScript running in the Web view to invoke native code contained in the application [7,2,8,19].

5.3 Hybrid development

Essentially, they are cross-platform applications but inside a native app shell. They render the UI by using an embedded web browser just like cross-platform apps do, and at the same time, elements that require responsiveness and high performance can be developed using native capabilities. A user is unlikely to notice the difference between a native and hybrid application. These applications can be downloaded through app stores

5.4 Choosing the optimal development approach

The table below differentiates between native and cross platform:

Table 6. Comparison between Native and Cross-Platform

	Native Platform	Cross Platform
Languages	Objectives-C Java C++ C# Vb.Net	HTML CSS JavaScript Web programming language
Installation on device	Yes	Mobile web application: No Hybrid app: Yes
Cost	Costly	Inexpensive
Development time	Long time for complex apps	Drastically reduced.
Usage	Highly graphical	Limited graphics, and performance issues

Choose native development if the application needs full access to all device hardware and functionality, the application processes video, audio or images, the application need high responsive user interface, the user interface should feel familiar to users on each platform and the application performs complex calculations or works with large sets of data, the application includes encrypted data.

Some drawbacks of native development are:

5.4.1

Development speed, supporting multiple devices requires more time to build an application (two or more completely separate apps need to be developed).Testing time is also affected.

5.4.2

Development costs, separate development for each individual platform requires more employees to be involved thus leading to more expenses.

5.4.3

Maintaining and supporting a native application for finding and fixing bugs, updating and any cosmetic change takes on average twice as much time and resources.

Choosing cross-platform development if want to develop a quick prototype to test and validate a simple concept, need a temporary 'stopgap' or promotional application which has a limited lifespan, the application has a simple user interface and has limited user interaction such as listing and showing news content, the user interface has limited scrolling and swiping and users will not be affected by a sluggish screen response, the application does not interact with any device hardware, camera, microphone, geolocation, accelerometer etc., the application with low budget and does not process complex data or work with audio or video.

Some drawbacks of cross-platform development are:

1. Poor performance. The most notable issues or sometimes even show-stoppers that impact performance happen with animation, clicks, and scrolling.

2. Limited access to phone's native APIs.

3. User experience for different platforms needs to be combined into one which could be quite difficult. Otherwise, an application built following the iOS Human Interface Guidelines will make Android users uncomfortable and vice versa.

6. MOBILE APPLICATION DEVELOPMENT TESTING

The increase in mobile applications in the marketplace makes it a challenge to develop mobile applications in good quality in order to compete. There are many aspects to mobile applications quality and test is one of them. Test is to measure the user satisfaction before the mobile application is delivered to the end users. Test is an efficient strategy, and it is required to check if the application would cover customer expectations and objectives or not. If the application often crashes or cannot be used, it will get poor ratings and bad comments by the user. Consequently, this will lead to low downloads that will result in fewer revenues. Testing, in general, is the process of executing a program to find errors before it is delivered to the users.

Mobile applications test is more complicated than the software or web test. Bakhtiar M. Amen, Sardasht M. Mahmood and Joan Lu [1] have managed to create a mobile

application test matrix that consists of four aspects. The aspects are test techniques, testing environment, test level and test scopes

6.1 Test Techniques

There are two kinds of manual and automated test techniques. The automated test is based on tools. Tools such as Monkey Talk, Test Plant and other tools [1]. On the other hand, manual tests are more relying on human interaction [1] [3]. Manual test techniques are based on human participants and developers. The developers - experts- observed the experiment and then they reported it. This kind is manually executing and there is no programming knowledge required. The testers must have some skills such as mobile experience usage. Automated test technique, which is non-human, required a bit executed by tools. This kind saves time and there is no need to have a participant to do the test.

6.2 Test Environment

To test an application, it has to determine the environment that will face in the test. One of the environments is the different types of devices:

 Real mobile

 Emulator device: "virtual mobile device in computers".

6.3 Test Scopes

Generally, test scopes are categorized in two major parts, functional (black box) and structural (white box).Black box is to detect functionality errors in the application. White box is to detect the logical errors in the application code. By comparing between the black and white box, the first is to test the application functional and non-functional requirements without knowledge of it source code by independent team. The second type is to test the application with good knowledge of the source code by the developers.

6.4 Test Levels

Test level is one of the fundamental crucial parts of mobile application development. Bakhtiar M. Amen, Sardasht M. Mahmood and Joan Lu [1] described the different test Levels as follows:

• Unit Testing: To test the application in independent parts or units.

• Functionality Testing: To check the application functionality and compare the user's requirement.

• Usability Testing: To check the application interface and how it can be used in an easy way.

•Compatibility Testing: To verify and validate apps compatibility

• Regressions Testing: To ensure the correctness of application operation Security Testing: To ensure the protection of the information

• Acceptance Testing: To Deliver and evaluate the application in aspect of end user view

• Network Testing: To check application connection strength and weakness.

The high demand and fast growth of mobile applications have attracted extensive research. The success of mobile application developing depends on an easy-to-use interface which is critical for adopting and using the applications. One of the important research issues in mobile applications development is usability test. This paper is to review existing studies on usability tests since usability test is an evaluation method used to measure how the users can use a specific application.

6.5 Usability Testing Definitions

In the past, usability was subjectively evaluated by informal processes. Year after year the usability becomes more important, and the definition of the usability testing has developed [4]. In (1995) Bevan said usability test is "The capability to be used by humans easily and effectively, quality in use." The ISO - international stander organization - In 1997 " Usability refers to the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. " ISO/IEC 9126-1 "The capability of the software product to be understood, learned, used and be attractive to use when used under specified conditions.". Hornbeak and Effie in (2007) gave the usability "The effectiveness, efficiency and satisfaction whit which specified users can achieve goals in particular environments ".

6.6 Usability Attributes and Dimensions

Usability is defined as the capability of a product to be understood, learned and operated. The usability has attributes and dimensions. As a summary, several usability dimensions and attributes were introduced by previous studies to measure usability of mobile-related applications. Each model measures different aspects of usability and emphasizes different levels of priority. In order to ensure holistic usability design and evaluation of mobile applications that can be executed to determine which one the papers are focused on.

• Effectiveness: The completeness and accuracy with which users achieve their goals.

• Efficiency: The speed and accuracy with which users can complete their tasks. • Satisfaction: How pleasant is to use the application?

• Learn ability: The application is easy to learn.

• Simplicity: The application is simple.

• Understandable: Determines the ease of which the applications functions can be understood, relates to user mental models in Human Computer Interaction methods.

• Memo ability: The application can be remembered.

• Flexibility: The application is flexible.

• Consistency: The elements have the same style in all of the applications pages.

• Adaptability: Characterizes the ability of the application to change to new specifications or operating environments.

• Operability: The application is easily operated by a given user in each environment

All the usability attributes are important, but effectiveness, efficiency and satisfaction are the common use of usability attributes. Different dimensions [4] of usability should be highlighted in designing and evaluating a product. There are 4 contextual factors that should be considered within the usability dimensions: user, environment, technology and task/activity

• User: studies discover the user profiles as an important requirement for user evaluation such as culture, knowledge, experience.

• Task: A task can be visualized as a use case scenario based on product operation or application task. Realism (outcome and user outcome), description (open and understand).

• Technology: Device profile can be categorized by device features, hardware, software, network connectivity

• Environment: The environment can be viewed as device type, geographic location, environmental data (temperature, noise); social conditions (group of users or event that take place).

6.7 Issues Of Usability Test

Determine the issues important to avoid it:

1. Display: The display capability of mobile devices supports less display resolution

2. Screen size: Small screen size, can significantly affect the usability of mobile applications

3. Connectivity: This problem largely affects data downloading time and quality of streaming media

4. Constraint Power and Processing Capability: applications that require a large amount of memory for graphic support or

fast processing speed, such as an application of 3D city maps for PDAs

5. Unclear meanings of labels

6. Icon/symbol/graphical representation

7. Order of the interface elements

8. Mental model mismatch—unexpected feedback of the product

9. Appearance/look of the product

6.8 Usability Lifecycle

6.8.1 EARLY USABILITY TEST

As viewed before in the development phases, design is one of the developments of life cycle. In the design phase of the application development life cycle apply an early usability test. Early usability test is to test the application before moving to the implementation phase. Testing the usability in the design phase will reduce the risk and the cost of the development. Quality and customer satisfaction will be ensured in this layer. Also, the participants may give new ideas to the design or the interface. This will lead to the final usability requirements. First, the early usability test will begin with a report which comes from the analysis layer, and it is called "initial usability requirements". In this step, It must report the target usability attributes and diminutions which users focus on. The data gathering techniques are also reported in the section. Data gathering techniques refer to how to collect the data by observe the participants and how to record their interactions with the application such as (observe, report, interview, questioners and using camera).

Second, Production design which is paper prototyping Fig 2. This is a sketching process and construction of low fidelity prototypes. The developer can choose a group of users or participants to test the application on paper as a virtual mobile use. Several problems of the usability can be seen in this process in an early stage. In this type It can find the interface problems and fix it with least cost possible.

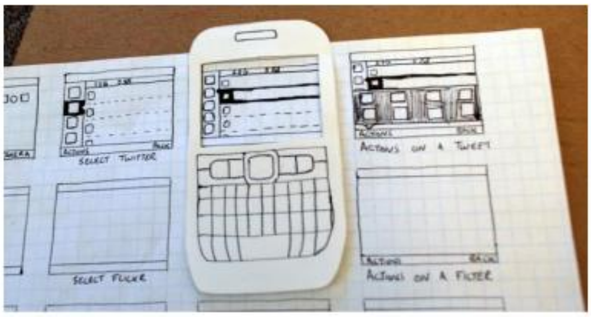

Fig 2: Paper prototype

Third, Early usability test as it is mentioned before this step will evaluate the usability test in early stage on papers prototype. Finally, we got the final usability requirements report based on the paper prototype test and the participants comments. Also, the participants give some ideas about the interface and what they wish to be in the application (Fig 3)

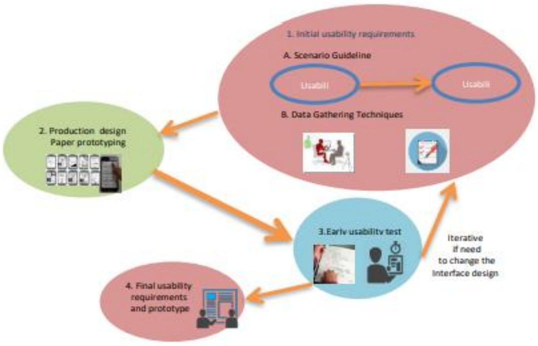

Fig 3: The early usability test in the design process

6.8.2 USABILITY TEST

After the implementation phase of the development, there is a test phase which is testing the application. In this paper, we present the usability test. Usability test has two main techniques and methods. Choosing between them based on many questions:

- What is the test technique like to use?
- What are the tasks in the application?
- What are the environments will use in the test?

The above three questions lead us to have a good choice and have a good approach. The approaches determine based on which techniques, methods and environments are used in the test. Therefore, it is important to choose an approach because it will make a difference in the usability test evaluation. If the application has errors, interaction mistakes or there is a need to change tasks there will be a need to go back to the implementation phase an iterative step. If there is no need, the application will be delivered to the end users (Fig 4).

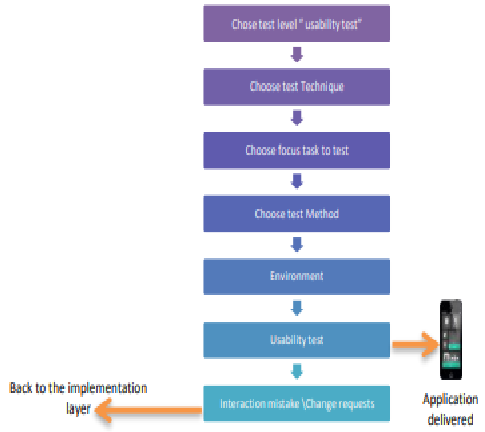
Chose test level " usability test"
Choose test Technique
Choose focus task to test
Choose test Method
Environment
Usability test
Interaction mistake \Change requests
Back to the implementation layer
Application delivered

Fig 4: Usability test

6.8.3 USABILITY METHODOLGY

Different techniques and methods describe different ways to evaluate usability tests. Selecting different approaches to evaluate usability depends on the different choics of usability test attributes and diminutions (tasks, environment, user and technology) which are used in the current test. Two major methodologies that have been applied to usability test of mobile applications are laboratory- lab - and field test. In a laboratory, human participants are required to accomplish specific tasks using a mobile application or emulator device in a controlled laboratory task. Laboratory is more costly because it requires a special room which has a window to let the experts observe the test experiment. The experts can control tasks relevant to the experiment. Interruptions, movement, noise, etc. are not found in laboratory methods which could affect the users' performance. Not all mobile sets are tests such as cannot do the test when the application has location functionality. The field test is the second methodology which is testing the application in real environments. This methodology can test all application functionality. In the past, field tests were difficult because there was no video recorder but nowadays it has become easier to attach a small camera to record the screen of the mobile device. By way of comparison, users behave less positively and more negatively in the field than in the laboratory. Laboratory takes more time from the expert. Field tests require extra effort from the participants than the laboratory. The participants also take longer time in the field to perform certain tasks and present more negative feelings, such as dissatisfaction and difficulty to use, so it consumes more time than in laboratory. If more time is spent in application development, then the budget also will increase. The risk in the field test, the result of the test depends on the network signals which can be weak sometimes (Fig5).

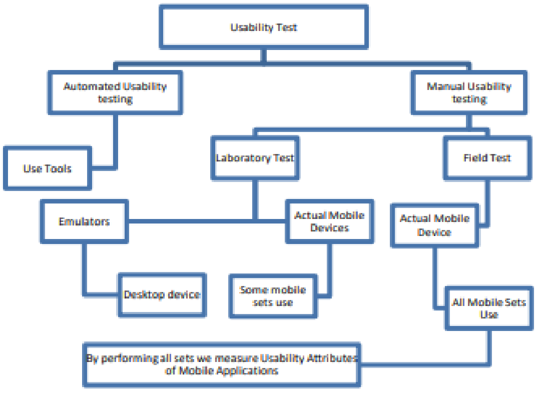

Fig 4: Usability test Methodologies

6.9 CASE STUDY Cycle

In 2005 Bettiol and Cybis [9] compared and evaluate the usability test methods by three approaches by using ANOVA test. ANOVA test is used to determine whether there are any statistically significant differences between the means of three or more independent groups.

Three distinct approaches were defined to conduct the usability tests based on the evaluation contexts. These contexts differed only in three aspects: (mobile phone or emulator), the location for conducting the tests (usability lab or field) and the equipment used to register the interaction (computer, document camera or wireless camera).

The first approach, laboratory test method using an emulator device (desktop). This is called a" Emulator context" approach. There were no cameras filming the participants, the image recorded on tape was only that of the computer screen. There were no cameras filming the participants. The evaluator was in the same room as the participants observing the interaction on the TV screen.

The second approach, laboratory test method using actual mobile phone device with document camera. This is called a "Document camera context" approach. There were no cameras filming the participants. The tripod could not move from its initial position, so the mobile phone could not leave the visualization area of the camera.

The third approach is testing methods using actual mobile phone devices with wireless camera. This is called a "Wireless camera context " approach. There were no cameras filming the participants. The camera was fixed to a support so its movements would accompany the movements of the mobile phone. The evaluator was seated close to the participants observing the interaction through the video camera screen. The support had several degrees of freedom of movement, which enabled the camera several different adjustments to obtain the image of the phone without hampering the users' view and quality of filming.

The participants were requested to fill out a questionnaire. Then participants were asked to accomplish some tasks. After the participants finished each task there was a quick interview with them to clarify doubts, they faced. At the end the participants were also requested to fill out the satisfaction and workload questionnaires.

It is important to say that some studies have found that doing tasks while participants were walking or standing requires more attention and can lead to slowing down the tasks' performance as compared to and sitting.

Laboratory does not simulate the real context and lacks the desired ecological validity. As a result, it is assumed that more attention should be paid to situational considerations in evaluating mobile devices such as interaction style, privacy, mobility, context, lighting conditions, varying levels of noise, multitasking as well as unexpected external interruptions.

In this experiment the results were better in the first approach which was emulator device in the laboratory method was the lower usability problems and effects which found in laboratory. The field test method was the second best.

The real mobile phone and document camera at the laboratory method record the highest risk and problems. But this is not fair to compare the two methodologies because there were different environments and cameras.

Finally, based on its results, Bettiol and Cybis suggest repeating this experiment using the wireless camera and real phone in the laboratory and in the field, with the user standing or walking, for better testing the effects of the mobility aspect and of these two different contexts over usability evaluations.

7. CONCLUSION

Various versions and platforms for mobile have been around for some time. Mobile application eras were discussed in detail. Mobile application development portability are categorized into three categories: native, web and hybrid mobile applications. The native app development approach is the best choice for excellent performance, but the development cost is expensive. The web app development approach is inexpensive but limited in functionality and cannot offer exceptional user experience. The hybrid approach is a middle way for many situations, especially if the developers want to create a single app that targets various platforms. There are several systems development methodologies including traditional and agile methodologies being utilized in current systems development, the appropriate development methodology framework for mobile application development is agile methodology. An appropriate agile method could be selected for a given project and can be tailored to a specific requirement based upon project's complexity and team size. Building a mobile application follows a similar set of four fundamental phases: planning, analysis, design, and implementation.

There are three development approaches: native, cross-platform and hybrid. If the application is not a game and aimed at long term development, requires smooth performance and needs to be as responsive as possible – native development is the best choice. Cross-platform implementation can be also an optimal solution for B2B and business process automation applications where deployment time is more vital than good look and feel. When you need to quickly enter the market to conduct some tests for your business idea, or when you have a website that you want to be turned into an application for a minimal price.

8. REFERENCES

[1] J. Dehlinger and J. Dixon, "Mobile application software engineering: challenges and research directions". Workshop paper. Oct 2011.

[2] PHYO, Min Tun, "Choosing a mobile application development approach", ASEAN Journal of Management & Innovation, June 2014.

[3] Anthony I. Wasserman, "Software engineering issues for mobile application development", Proceedings of the FSE/SDP workshop on future of software engineering research, 11/2010.

[4] Sobia Zahra, Asra Khalid, Ali Javed, "An Efficient and Effective New Generation Objective Quality Model for Mobile Applications", I.J. Modern Education and Computer Science, 2013.

[5] Marco de Sá and Luís Carriço LaSIGE & Department of Informatics," Lessons from Early Stages Design of Mobile Applications ", Amsterdam, the Netherlands, 2008.

[6] Adriana Holtz Betiol and Walter de Abreu Cybis, " Usability Testing of Mobile Devices: A Comparison of Three Approaches ", 2005.

[7] Kim W. Tracy, " Mobile application development experiences on Apple's iOS and Android OS", IEEE.

[8] Jigna Solanky and Kajal Patil and Gayatri Patel, "Resemblance of PhoneGap and Titanium for Mobile Application Development", International Journal of Computer Applications, 2016.

[9] Sunny Dhillon and Qusay H. Mahmoud, "An evaluation framework for cross-platform mobile application development tools", Software: Practice and Experience, Volume 45, Issue 10,10/2015.

[10] Muazzan Binsaleh and Shahizan Hassan, "Systems Development Methodology for Mobile Commerce Applications", International Journal of Online Marketing, 2011.

[11] Harleen K. Flora and Dr. Swati V. Chande, "A review and analysis on mobile application development processes using agile methodologies", International Journal of Research in Computer Science, 2013.

[12] Pekka Abrahamsson and Antti Hanhineva and Hanna Hulkko and Tuomas Ihme and Juho Jäälinoja and Mikko Korkala and Juha Koskela and Pekka Kyllönen and Outi Salo, "Mobile-D: An Agile Approach for Mobile Application Development", Proceeding OOPSLA '04 Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications, 2004

[13] Bakhtiar M. Amen and Sardasht M. Mahmood and Joan Lu, "Mobile Application Testing Matrix and Challenges".

[14] Pallavi Bhuarya, Shruti Nupur, Anuradha Chatterjee and Rajesh Singh Thakur," Mobile Application Testing: Tools & Challenges", International Journal of Engineering and Computer, 2016.

[15] Rosnita Baharuddin, Dalbir Singh and Rozilawati Razali," Usability Dimensions for Mobile Applications-A Review ", Research Journal of Applied Sciences, Engineering and Technology, 2013.

[16] Harleen K Flora, Xiaofeng Wang, Swati V Chande, "An investigation into mobile application development processes: challenges and best practices", International Journal of Modern Education and Computer Science, Volume 6, Issue 6, 06/2014.

[17] Bakhtiar M. Amen and Sardasht M. Mahmood and Joan Lu, "Mobile Application Testing Matrix and Challenges".

[18] B.A.L. Madhushani1 and P.H.A.M. De Silva2 and W.A.L. Madushanka3 and M.G.T.H. Malalagama4 and Dilhan Manawadu5," Challenges in Mobile Application Testing", 2014.

[19] Tom Seymour and Jasmine Zakir Hussain and Sharon Reynolds, "How to Create an App", International Journal of Management & Information Systems, 2014.

[20] [B.A.L. Madhushani, P.H.A.M. De Silva, W.A.L. Madushanka and M.G.T.H. Malalagama, Dilhan Manawadu, " Challenges in Mobile Application Testing: Sri Lankan Perspective" , COMPUSOFT, 2014.

[21] Eeva Kangas and Tomi Kinnunen, "Applying User-Centered Design to Mobile Application Development ", COMMUNICATIONS OF THE ACM, 2005.

[22] Fatih Nayebi, Jean-Marc Desharnais and Alain Abran, " The stat of the art of mobile application usability evaluation ", Canadian Conference on Electrical and Computer Engineering ,2012.

[23] Naseer Ahmad, Muhammad Waqas Boota and Abdul Hye Masoom, " Smart Phone Application Evaluation with Usability Testing Approach ", Journal of Software Engineering and Applications,2014.

[24] Yao Wang," What is Good Enough for Mobile System Evaluation: A Comparison of Laboratory and Field Setting ", Rochester Institute of Technology RIT Scholar Works,2012.

[25] Dongsong Zhang and Boonlit Adipat, " Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications", INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION, 2005.

[26] Naveen Singh and Ravi Shanker Yadav, " Field Testing of Mobile Applications: An Effective Approach", International Journal of Computer Applications, 2013.

[27] Corne´ P. J. M. van Elzakker, Ioannis Delikostidis1 and Peter J. M. van Oosterom, " Field-Based Usability Evaluation Methodology for Mobile Geo-Applications", The Cartographic Journal,2008.

[28] Caspar Ryan and Atish Gonsalves, " The Effect of Context and Application Type on Mobile Usability: An Empirical Study", School of Computer Science and Information Technology.

[29] Youn-kyung Lim, Apurva Pangam, Subashini Periyasami and Shweta Aneja, " Comparative Analysis of High- and Low-fidelity Prototypes for More Valid Usability Evaluations of Mobile Devices", NordiCHI, 2006. ISO definitions: http://www.sqa.net/iso9126.html.