Addis Ababa University

College of Natural Sciences

Department of Computer Science

Usability Testing Process Iteration Framework for Mobile Applications

Kiros Shiferaw Abate

A Thesis Submitted to

The Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science (in Software Engineering)

Addis Ababa, Ethiopia

April 12, 2021.

Addis Ababa University

College of Natural Sciences

Department of Computer Science

Kiros Shiferaw Abate

Advisor: Ayalew Belay (PHD)

This is to certify that the thesis prepared by Kiros Shiferaw, titled: A Comprehensive Usability Testing Framework for Mobile Applications and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science (in Software Engineering) complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Name Signature Date

Advisor: _____
Examiner:_____
Examiner:_____

Abstract

Mobile applications are playing an increasingly important role in facilitating efficient information exchange, making them a significant driving force in mobilizing businesses and improving productivity. However, the value of any application would be undermined if the user is not able to fully utilize its functionality, thus the role of usability testing process iteration frameworks plays vital role to follow usability design guidelines and conducting testing. This could result in better utilization of the developed app. Therefore, measuring usability is an essential task to ensure the application is accurate, has sufficient in speed and to ensure the safety of the user from strain injury as well.

Usability testing in mobile application aims to identify the main issues in the user interface that may lead to human error, terminate the user interaction with the mobile application and cause user frustration. Usability testing frameworks are considered as major success factor and led to higher acceptance for the mobile application since the usability testing is basic of software application development. This paper proposes a usability testing process iteration framework for mobile applications and the framework acts as a structured and general methodology for designing usable mobile apps and evaluating mobile application by collecting usability information among user interaction with mobile applications.

In addition, this work also attempts to review the existing interface design guidelines and consequently develop a usability testing process iteration framework. The framework serves as basis for usability testing consisting of guidelines, quality characteristics, goals (interface design criteria), questions, usability metrics and two evaluation instruments (task list and questionnaire).

To ensure the effectiveness and reliability of the framework and to examine its performance, it is validated by applying the suggested metrics and evaluation instruments (task list and questionnaire) in a usability study conducted on five mobile applications for mobile users. Therefore, from this usability works results gathered from usability testing proved that the framework is applicable for usability testing of mobile applications.

Keywords*: - *Mobile app, usability, usability testing, design criteria. framework

Dedication

To my mother: - You have been working so hard to raise me. You are the hero in my life for the things you did to me. God give you a long life and good health.

Acknowledgment

First of all, I would like to express my heart feeling thanks to the almighty GOD and his mom; without his assistance nothing would be happened.

I would like to convey my gratitude to numerous people who provided me with exceptional support, encouragement and wisdom throughout my MSc journey.

I would like to express my very great appreciation to my major advisor, Assistance Professor Ayalew Belay (PHD) for his resourceful comments and suggestions throughout the research period and at the time of write up process of this thesis and for his motivative and constructive guidance right from the moments of problem formulation to the completion of the work. Many thanks and appreciations go to him for the discussions with him always made me think that things are possible. His enthusiasm and encouragement has always inspired me to accelerate to the completion of the work.

I am also very thankful to my instructors and all staff members of the department of Computer Science for their contribution in one way or another for the success of my study.

I would like to forward my special thanks to test users (participants) for their commitment and their volunteers by spending their time and effort to participate in the evaluation process.

Finally, I am very grateful to my mother and all the rest of my families, friends, and peers who, in one or the other way brought me up to a success in my academic endeavor.

Table of Contents

List of Figure

List of Table

Chapter One: -Introduction

1.1. Overview

Mobile applications, also called mobile apps, are software applications, usually designed to run on smart phones and tablet computers. It also originally intended for productivity such as email, calendar, news alert, weather forecasting and contact databases, but public demand caused rapid expansion into other areas such as mobile games, factory automation, GPS and location-based services, banking, order-tracking, and ticket purchases [1]. Currently there are large numbers of new mobile applications, but many other mobile applications remain difficult to use due to lack of usability.

Usability work in software engineering is a measure of quality and it contributes to the overall acceptability of systems. According to Nielsen, usability testing is defined as a process in software development cycle and takes into consideration various factors: [2]

- ✓ Learnability: the system should be easy to learn, so that the user can rapidly start getting some work done with the system
- ✓ Efficiency: the system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible
- ✓ Memorability: the system should be easy to remember, so that the causal user is able to return to the system after some period of not having used it, without having to learn everything all over again.
- ✓ Errors: the system should have a low error rate, so that the users make few errors during the use of the system, and so if they do make errors, they can easily recover from them. Further, catastrophic errors must not occur.
- ✓ Satisfaction: the system should be pleasant to use, so that the users are subjectively satisfied when using it; they like it.

With further definition of the described factors, usability can be systematically approached, improved and evaluated. Typical measurement process includes a group of test users that uses the system to perform pre-specified set of tasks. The same system can be measured as having different usability characteristics if used by different users for different tasks.

Usability Testing (UT) consists of measuring the performance of typical users on typical tasks. In a broader sense it relies on a combination of techniques, such as observations, questionnaires, interviews and user testing. Usability helps mobile application easy to work but it cannot be directly measured [3].

Usability Testing (UT) also defined as the process of measuring usability and recognizing explicit usability problems. Usability testing in mobile application aims to identify the main issues in the user interface that may lead to human error, terminate the user interaction with the mobile application and cause user frustration. UT frameworks are considered as major success factor and

led to higher acceptance for the mobile application since the usability testing is basic of software application development.

During the testing, participants perform a series of pre-designed tasks. The Mobile developers observe the process in order to note the difficulties faced by each of the participants. Mobile devices are used by common people who don't have any technical knowledge. Most of the smart mobile phones available in the market, now-a-days, come with thousands of mobile applications. Hence, usability measure of mobile applications needs to consider these varying factors.

The main purpose of this work is therefore to design and develop a usability testing process iteration framework for mobile applications. Usability testing framework of user interaction acts as a structured and general methodology for designing usable mobile apps and testing mobile application by collecting usability information among user interaction with mobile applications. It also applies a theory for describing, explaining, and predicting usability differences.

1.2. Motivation

Usability tests identify areas where users struggle with a product and help testers make recommendations for improvement. The goal is to better understand how real users interact with the product and to improve the product based on the results. The primary purpose of a usability test is to improve a design.

In a typical usability test, real users try to accomplish typical goals, or tasks, with a product under controlled conditions. Mobile development team members watch, listen, collect data, and take notes. Now a day, mobile phone becomes the new personal computer. These mobile phones have started to become increasingly sophisticated and extremely powerful. Mean that in addition to making phone calls, they have capabilities to perform a variety function. Mobile applications are playing an increasingly important role in facilitating efficient information exchange, making them a significant driving force in mobilizing businesses and improving productivity. However, the value of any application would be undermined if the user is not able to fully utilize its functionality, thus the role of usability testing frameworks plays vital role to follow usability design guidelines and conducting testing. Which could result in better utilization of the developed app.

In addition to this, the widespread use of mobile device and the heterogeneity of the user's technical knowledge requires to have a usability testing guide lines for mobile applications. Therefore, it is important to have usability testing process iteration framework that can be used in evaluating the usability of mobile applications and this is the motivation for this research work.

1.3. Statement of the Problems

Measuring usability is an essential task to ensure the application is accurate, has sufficient in speed and to ensure the safety of the user. Recently mobile applications are coming into the market. People are using mobile apps for several purposes. The peoples are not aware of the user interface problems for these applications. The usability of an application may not necessarily be effective for the end-user. Misinterpretations of patterns and rigid ideas regarding the user interface design may adversely affect the user experience or unfavorable to the efficient use of an application. It is very expensive and risky to release the application to the market without testing it with real users. So before releasing the application to the market, tester can bring in people (real users) for the usability testing and fix the problems found from usability testing.

Even though the technical knowledge of mobile users as well as mobile applications are different mobile developers apply usability testing principles with no connection to these varying factors. For example, towards usability guidelines for mobile application and website is developed by Maria S [7]. In this scenario, the researcher is tried to identify existing guidelines for mobile apps. For this purpose, the researcher conducts a literature review focusing on the result of empirical usability experiments dealing with mobile apps. Conceptualization of the topic, literature search, literature analysis and driving agenda is the contribution of this paper. However, the proposed usability guidelines are not representing mobile application variance and also the proposed guidelines are not validated by usability testing. Similarly, for growing the understanding of the impact of contextual context of mobile device is developed by Fayaz A and Mark F [8]. This paper provides the way in which mobile device usability evaluation methods are being adapted to better reflect technology. However, the usability evaluation is only for mobile device. It doesn't serve as a method for usability evaluation of mobile application. Therefore, there are many other different mobile applications which are designed, developed and installed into a mobile device for different purpose and also users experience with mobile applications is different. Due to this, a usability testing process iteration framework that supports mobile developers for conducting usability testing needs to be developed.

In general, the questions that we are tried to answer during these studies are like: -

1. What are the major usability interface design principles and guidelines considering user as well as mobile app differences?
2. What are the major usability testing mechanisms to find and remove errors and faults in mobile application?
3. What are the major characteristics of mobile phone to identify obvious limitations for using mobile applications?
4. How can we select the important guidelines to design the testing framework?
5. What is the performance of the developed framework?

1.4. Objective

1.4.1. General Objective

The general objective of this thesis is to design and develop usability testing process iteration framework for mobile applications

1.4.2. Specific Objective

In order to achieve the general objectives, the following specific objectives were performed.

- ✓ Survey on usability as well as interface principles, guide lines
- ✓ Identify usability testing mechanisms and models
- ✓ Identify major characteristics of mobile applications
- ✓ Designing usability testing framework
- ✓ Evaluation by applying testing

1.5. Methodology

Design science research (DSR) is one of the most widely used research approaches in engineering, computer science, and information systems research. Design science research, as conceptualized by Simon (1996), focuses on the creation of innovative IT-artifacts to solve real-world problems. Hevner & Chatterjee (2010) define DSR as follows: “Design science research is a research paradigm in which a designer answers questions relevant to human problems via the creation of innovation artifacts, thereby contributing new knowledge to the body of scientific evidence. The designed artifacts are both useful and fundamental in understanding that problem”. DSR thus provides new knowledge through the design of innovative artifacts and the evaluation of performance of these artifacts (Vaishnavi and Kuechler, 2004; Hevner et al., 2004; March & Storey, 2008).

The purpose of this research project is to design and develop a usability testing process iteration framework for mobile application. As researchers, we chose to adapt Peffers et al [2008] design science research paradigm. Therefore, the research methodology is based on the process steps for DSR as described by Peffers et al (2008).

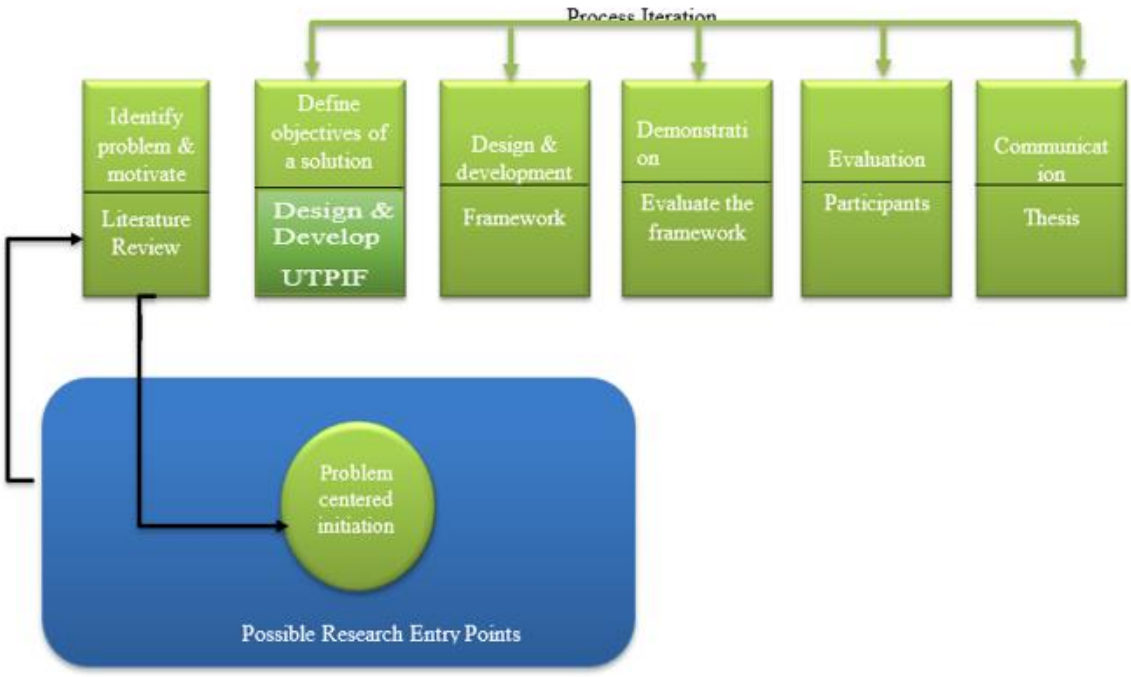


Figure 1.1 adapted design science research methodology from Peffers et al (2008).

Problem identification and motivation

The problem is initiated from literature review. Therefore, in this activity we tried to define the specific problem statement and we are also trying to justify the values of the solution which is found in chapter 1.1, 1.2 and 1.3.

Definition of the objective for a solution

The objective of the solution is actually to develop a usability testing process iteration framework. So, in this activity we are try to gather the objective of the solution from problem statement. The research was start with the literature gathering stage. This helps us to gather existing knowledge from the scientific knowledge base and to identify existing research gaps in current research on

usability of mobile applications. After literature gathering, the literature is analyzed. This analysis focuses on analyzing the outcome of the previous stages. The outcome, all relevant sources are analyzed and put into perspective with the research topic.

Design and development

In this stage after utilizing the theoretical results derived from earlier stages. We are trying to design and develop the framework. The detail is found in the coming chapter.

Demonstration and Evaluation

In this stage we are try to use real device test equipment to evaluate the proposed framework by using usability attribute and by selecting some mobile applications from different type and by including different mobile users. In addition, during evaluation we are try to use some data collection technique like tasks, questionnaire and interviews in order to gather qualitative and quantitative data from participants (test users) and the detail is found in chapter 5.

Communication

In this stage we are try to communicate our findings with different research community in order to obtain feedback as well to contribute our findings in the usability of mobile applications. In addition, the designed framework is presented by means of thesis presentation and defense and then this research will be submitted and published for master's award in the future

1.6. Scope and Limitation

The scope of this research project is limited to investigate designing and developing a usability testing process iteration framework for mobile applications which is important to guide the mobile application developers to conducting testing in mobile applications and helps to design usable mobile apps. This research project is also limited on conducting survey on usability, interface principles and guidelines. In addition, this research focused on answering the questions which are described in the above (statement of the problem). The designed framework is helps only for mobile applications it doesn't serve as a framework for others applications like windows application.

1.7. Application of Results

Usability testing is an evaluation method used to measure how well users can use a specific software system. Therefore, this study is expected to bring remarkable changes in the design and development of a usability testing process iteration framework for mobile applications. Thus, it will help mobile developers for conducting usability testing for different mobile applications considering different mobile users when the application reaches at testing phase before releasing to real users. The framework that we developed is much more significant to: -

- ✓ Mobile developers in order to avoid confusing
- ✓ Developed applications for conducting usability testing
- ✓ Improve test accuracy and increase test speed
- ✓ Testers to test easily and to fix usability faults
- ✓ Provide consistency in usability testing of mobile applications
- ✓ Have suggested usability testing language for mobile applications
- ✓ Provide general methodology for designing usable mobile apps and testing mobile applications

1.8. Organization of the Thesis

This thesis is organized as follows: -

Chapter 1 is the introduction to this work, and provides a summary of the background of this thesis.

Chapter 2 discusses literature review on different issue in mobile application and usability. In this chapter Overview of mobile apps, mobile testing approach and techniques, test levels, test automation, continuous integration, mobile application quality from different perspective, usability and different usability models and usability testing mechanisms is discussed.

Chapter 3 provides an overview of related works that deals with the usability of mobile applications, and summarizes a selection of the most relevant findings. And also, summary of related works is presented.

Chapter 4 presents an overview of user interface design, usability testing framework for mobile application. In addition, a mobile application design criterion. methods, metrics and usability testing process iteration framework and description for each framework component is the other presented topics in this chapter.

Chapter 5 discusses evaluation using mobile applications, test users, metrics and two evaluation instruments drive from the metrics. In this chapter, system usability scale for test user evaluation purpose, selected mobile application for testing purpose and result analysis is presented.

Chapter 6 states the conclusions drawn from our work and suggests and shows some possible directions and recommendation for future studies.

Chapter Two: - Literature Review

In this section a review of literatures conducted in order to understand the problem associated with the area of the objectives of this thesis and also to identify appropriate direction. Overview of mobile apps, mobile testing approach and techniques, test levels, test automation, continuous integration, mobile application quality from different perspective, usability and different usability models is presented.

2.1. Overview of Mobile Application

Mobile applications are a rapidly developing segment of the global mobile market. They consist of software that runs on a mobile device and perform certain tasks for the users of the mobile phone. Mobile apps are common on most phones including in expensive and entry level models. Their wide use is due to the many functions they perform, including providing user interface for basic telephony and messaging service as well as advanced service such as game and videos [1].

A mobile application can be an opportunity to improve interaction with customers, create brand awareness and even create additional revenue. But if the objective of the application are unclear customer can be upset and money can be lost [2]. Mobile applications, referred to software systems operating on mobile devices, are evolving rapidly, making ubiquitous information access at anytime and anywhere a true reality. For example, many mobile applications have brought Internet services to mobile. In the business area, M-Commerce (Mobile e Commerce) applications, such as mobile banking and advertising, extend electronic businesses to mobile devices. Customers can check their bank account balances and carry out business transactions through their cell phones [3].

There have been some usability studies for mobile applications. In the field of mobile education, usability studies are conducted when mobile devices are used for collaborative learning or information access. In the entertainment industry, mobile users can enjoy watching video or playing interactive games on their mobile devices. Those advanced features of mobile applications enable users to carry out a variety of activities through mobile devices. Because achieving a high level of user satisfaction is critical to the success of mobile applications, usability testing is a mandatory process to ensure that a mobile application is practical, effective, and easy to use, especially from a user's perspective.

2.1.1. Mobile Application development Option

Similar to desktop applications, there are various kinds of mobile apps. Mobile apps are classified by the programming language they used. There are three types of mobile applications, native app, web app and hybrid app. These three types of mobile apps have their own advantages and disadvantages, and these will be discussed in this section. Determination of chosen option is important from testing perspective because many tools and test cases depend on approaches used for development.

A. Native Application

These applications are often developed in platform specific SDK and languages which are commonly Java or C++ for Android, Objective-C or Swift for iOS and C#, Visual Basic or C++ for Windows Phone. Native application can use all phone peripheral and resources like GPS, NFC, Bluetooth, accelerometer and compass, etc. It can access the contact list, gallery or user media files. These applications are usually installed and updated from the platform specific application store. Native applications are limited only by OS API and application store policies. They usually provide the best performance [4].

The pros of native apps include that they provide the best user experience and interaction. Native apps offer different experiences for different platforms. And, native apps can be run offline for saving the bandwidth. Also, native apps are able to access local resources. And, for business, native apps have a clear revenue model [5].

The cons of native apps include that they are complicated to transplant to other platforms. Native apps have higher costs for development and maintenance. Also, native apps need to be submitted to App Store. And, native apps need to share profits with App Store [5].

Native apps are the best choice for applications requiring high performance, such as games and photo editing applications. And because of native apps have ability to invoke notification system, apps which require push notifications, for example chat apps, should be developed as native apps [5].

B. Web Application

This category refers to a standard web application developed using traditional web technologies (HTML, CSS, and JavaScript) and server-side code in Node.js, PHP, ASP.NET, etc. These web applications are suitable for browsing on smartphones or tablets and user needs a browser and Internet connection to use them. There are two main approaches to make the web application suitable for handheld devices. The first approach uses responsive design, which dynamically reorders page layout. This is supported with new features introduced in CSS [6].

The second approach uses information from user HTTP request to determine whether client uses mobile web browser. If so, customer is redirected to the web site tailored for handhelds devices [7]. Even the web applications can use some device resources which are provided by the web browser – in most cases, it is possible to use GPS, camera or accelerometer, but cannot use NFC or Bluetooth [8]. Communication with other applications is also restricted on browser functionality. Because of that, web application cannot use phone contact list or messages. Web applications are not installed from application stores and they do not provide such a good performance as native.

The pros of web apps include that they do not need installation. They are cross platforms. They have lower cost of development and maintenance. And, they can be updated rapidly. And compare with native apps, web apps have lower requirement for specifications [5]. The cons of web apps

include that they need the Internet. They have slower response and depend on the speed of bandwidth. Also web apps have compatibility issues with different browsers. Web apps are not available on App Store. And, web apps have security risks [5].

C. Hybrid Application

This is a mix of previous technologies. Hybrid applications are often built for multiple platforms so their main attribute and advantage is portability. There are several leading approaches how to build hybrid applications. The most often used cross-platform frameworks utilize web technologies [9]. The difference between this hybrid and web application is that the hybrid one has native application package and it is often installed from the platform specific application store.

Widely used cross-platform frameworks provide container for running web application written usually in JavaScript. This approach is used by Apache Cordova, but there are more frameworks working similarly. Using several JavaScript libraries and Cordova API, it can be implemented gesture control, battery status check and some other platform dependent features [10]. Another approach to achieve transferability of mobile application is using Mono, which is multiplatform open source implementation of Microsoft .NET framework based on ECMA standards for C# and the Common Language Runtime [11]. The code is written in C# and is compiled to platform specific binaries using Mono or distributed along with Mono runtime library. One of the frameworks build on Mono is Xamarin.

Hybrid applications are not fully platform independent, but most of them cover at least Android and iO [12]. The range of available APIs depend on used framework. In general, hybrid frameworks provide better and wider usage of device resources than web applications but less than native ones. If some new feature appears in OS update, it takes some time until hybrid framework developers implement it. Also quality of UI, security and performance are often worse than the native applications [13].

The pros of hybrid apps include that they have similar development process with Web Apps. They take advantage of Native Apps' features. Same as web apps, they can update the content without installing a new version. And, they can use a framework to help the development [5].

The cons of hybrid apps include that they use Web-based content which has slower response, which mostly depends on the bandwidth. They may not provide very high performance. They do not support all system features. And, they will be difficult to debug, if knowledge of native framework is lacking [5].

Table 2. 1 Comparison of web, hybrid and native apps [5]

Properties	Native apps	Web apps	Hybrid apps
Distribution	App store	Web	App store
Development speed	Slow	Fast	Moderate
Development cost	High	Low	Moderate
Maintenance	Hard	Easy	Moderate
Performance	Fast	Slow	Moderate
Offline mode	Yes	No	Yes
Installation	Needed	No Needed	Needed
Cross Platform	No	Yes	Yes

The development of mobile apps involves many processes. First, there should be a beginning idea, and then developers start to plan, and design the prototype. And, then they turn to development, testing and deployment. Afterwards maintenance and debugging are needed. After having the very beginning idea, the next thing to consider is which kind of an app is more suitable for the use. Native Apps have the best performance and the device access. Therefore, it is more suitable for developing games or mobile applications requiring high performance. Web Apps are cross-platforms and the development is easier and quicker, but the performance is poor and functionalities do not completely work. On the other hand, hybrid apps take advantage of native app and web app. Hybrid apps offer native performance and web-based content, and they are cross-platform, the same HTML code could be reused on different platforms. Usually, companies that want to get a presence in the App store and do not want to spend a lot of money. They will build a hybrid app as wrappers for their website. [14]

2.1.2. Mobile Operating System

Good knowledge of mobile operation systems and application environment is essential for every mobile application tester. In this section we present the major mobile operating systems which are android, iOS, Windows and Windows Phone.

A. Android Operating System

Android is the fastest growing OS in terms of popularity and the most widespread mobile operating system [15]. Android is owned and managed by the Open Handset Alliance – industry consortium for creating hardware, software and telecommunication open standards for mobile devices. This consortium is led by Google [16]. Since Android is open source Linux-based project, most manufacturers and cellular carriers take advantage of that and modify the operating system to suit their hardware, increasing the complexity of the system. This fact, along with slow new version adoption, makes Android the most fragmented mobile OS, which increases test costs and complexity [17]. However, many different testing tools and frameworks target primarily Android due to its popularity.

B. iOS Operating System

iOS is mobile operating system developed and owned by Apple Inc. It is closed source, UNIX-like operation system based on Darwin (BSD) and OS X [18]. iOS was originally developed for iPhones, but now iOS runs on iPad, iPod Touch or Apple TV. Other manufacturers are not licensed for using iOS. Therefore, only Apple-made devices can run it. Most iOS users (97%) have the iOS version 7 or the latest 8 [19]. As a result, testing on all available devices is not as difficult as on Android. OS fragmentation is presented mainly by the different OS feature set (some features are not available on some devices) [20]. On the other hand, Apple devices are one of the most expensive and their purchases increase testing costs. Standard integrated development environment (IDE) for native application is Xcode which can be installed only on Apple's operating system OS X.

C. Windows and Windows Phone Operating

With Windows 8, Microsoft moved Windows OS towards to mobile devices. Windows 8.1, which is current version, is able to run on PCs and tablets with x86 or ARM11 architecture. Moreover, Microsoft has operating system Windows Phone especially for smartphones. These two platforms are about to converge in Windows 10 which is currently in beta. Microsoft has two types of application projects – Windows Store 8.1 applications for PCs or tablets and Windows Phone 8.1 applications for smartphones. These two projects share lot of APIs, tools and design principles. Therefore, they were partially merged to the Windows Universal Applications, which enables code sharing between Windows Store and Windows Phone applications projects. Microsoft is third in smartphone OS market share far beyond the Android and iOS. Furthermore, most Windows Phone users have cheap low-end handhelds, which corresponds with the fact that 71% of Windows Phone applications are installed on low memory devices [21].

2.2. Mobile Testing Approaches and Techniques

2.2.1. Test equipment

In this chapter we will discuss another part of the mobile test environment – real devices, simulators, emulators and test clouds. All of them have benefits and drawbacks. This is based on [22], [23], [24].

A. Real Device

Testing on real devices is necessary for every mobile application. It gives the most realistic results and tester can perform all required test scenarios. On the other hand, it is extremely difficult and expensive to test application on all available devices.

B. Simulators and Emulators

Simulators and emulators are type of software that enables running another computer system on the host platform. They are available for each platform and often integrated into IDE like Android Studio, Xcode or Visual Studio. There is a difference between simulator and emulator. Simulator imitates software behavior and uses all available resources from host computer like CPU, memory, network, etc. Testing on simulator can provide biased results because host computer can be much faster than the mobile device.

Emulators provide more realistic results because they replicate target device more precisely. Processor speed, used memory or memory card presence, even the network connection type or signal strength can be set to create more realistic environment. There are lot of simulators and emulators on the market with different features and limitations. However, they all have some similar advantages and disadvantages

C. Mobile Device Cloud

Successful testing of application functionality on one device does not provide assurance across all others of the same operating system [23].Mobile device test cloud is a service which enables to run automation or manual tests on hundreds of physical devices in the cloud. Providers of these services have large variety of smartphones and tablets connected by USB cable in the rack. There are two common types of mobile device clouds available on the market – public and private [22].

Devices in public cloud are always accessible over the Internet. They can be geographically spread to achieve different configurations with different mobile carriers. It is possible to rent time on the devices for automated or manual testing, but these devices are always shared with other users. This type of mobile devices cloud offers for example Perfecto Mobile (8.4) or Keynote Mobile Testing. In case of private cloud, providers usually detach requested devices for exclusive use, which increase security. Private cloud can be dedicated if the company does not have IT department, or located in-house in the customer VPN behind the corporate firewall.

The main purpose of mobile device test clouds is to find bugs which appear only on some devices or some OS versions. It can reveal bugs that can be hidden from developers if they use emulators and only few physical devices. For example, UI thread overloading (which is common mistake) may occur like a short delay on physical testing device and on simulator, it could be imperceptible. But on older or low-end devices with low computational power, it can cause highly irresponsible user interface in the better case, freeze or crash in worse case. It has also some advantage and dis advantage based on [22].

2.3. Test Level

Test levels introduced in this section are based on ISTQB (International Software Testing Qualifications Board) with respect to mobile development. Each test level is corresponding to the specific development stage of V-model.

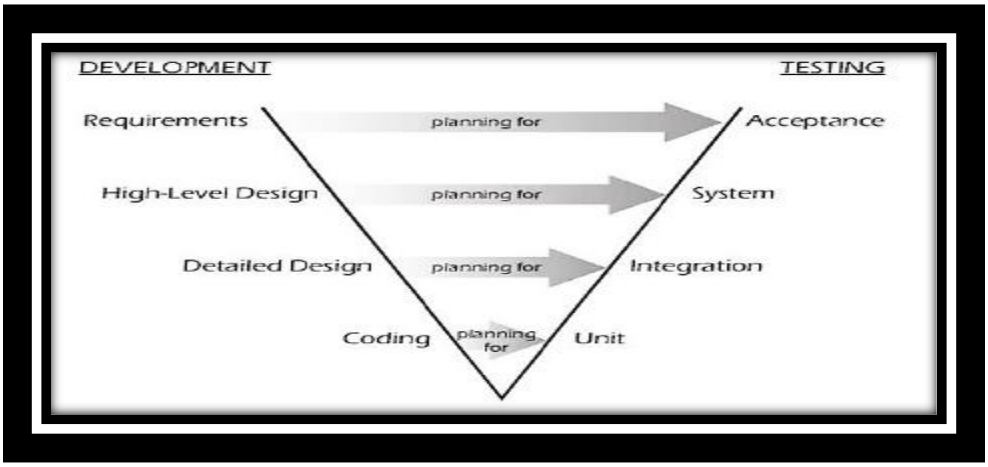


Figure 2. 1 Mobile development test level respect to specific development stage

2.3.1. Component Testing

Component testing (also known as unit, module or program testing) searches for defects in, and verifies the functioning of, software modules, programs, objects, classes, etc., that are separately testable [26]. Unit testing of a native application for Android and Windows Phone is the same as unit testing of any other application written in Java or .NET respectively. Developers can use popular unit and mocking frameworks like JUnit or Mockito for Android, or NUnit for Windows Phone. Tests can run locally (on Java virtual machine or .NET CLR) and results are shown in IDE.

In the case of Android, proper framework should be used because official Google's unit testing framework requires running the unit tests on device or emulator. This approach is much slower than running the tests on Java virtual machine [27]. Unit testing on iOS is different. Apple devices using A-series processors based on ARM architecture, but Mac computers, on which are iOS applications developed, runs on processors with x86 architecture. And because Objective-C is compiled language, iOS simulator has to be used for running ARM targeted application on x86 Mac. Second option is to run the unit test on the real device. The previous enumeration of programming languages suggests that tests on this level are done by programmers.

2.3.2. Integration Testing

Integration testing tests interfaces between components, interactions with different parts of a system, such as the operating system, file system and hardware, and interfaces between systems. [26]. ISTQB further distinguish integration between components of the system and integration between different systems. Therefore, the integration tests, as opposed to the unit tests, must run on the emulator or preferably on the real mobile device connected to the developer's computer or

the integration server. Integration tests are done by programmers or qualified quality assurance engineers.

2.3.3. System Testing

System testing should investigate functional and non-functional requirements of the system, and data quality characteristics [26]. Due to simulators and emulators limitations discussed in earlier these tests should be performed on the real devices in real environment (network conditions, targeted OS versions). Tests on this level are prepared and executed by quality assurance engineers.

2.3.4. Acceptance Testing

Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system [28]. Acceptance testing is always conducted in production environment, i.e. intended devices, OS versions and networks, by the end users or stakeholders.

Beta testing is a typical form of the acceptance testing. All three platforms support beta testing in a similar manner – an application package is uploaded to the store but is private. Access is granted only for users, who are added to the beta testing group simply by inserting their store ID or email in developer portal. If the application is intended for public use, testing through crowd can be taken into consideration.

2.4. Test Automation

Test automation is defined as: The use of special software (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes [29]. Automated tests can perform repetitive tasks which are difficult to test manually and then improve test efficiency and reduce costs. Mike Cohn introduced test automation pyramid in his book succeeding with Agile [30], which divides automated tests into three levels by their purpose and express their expected quantity. It means that the software project should contains much more unit tests than UI tests.

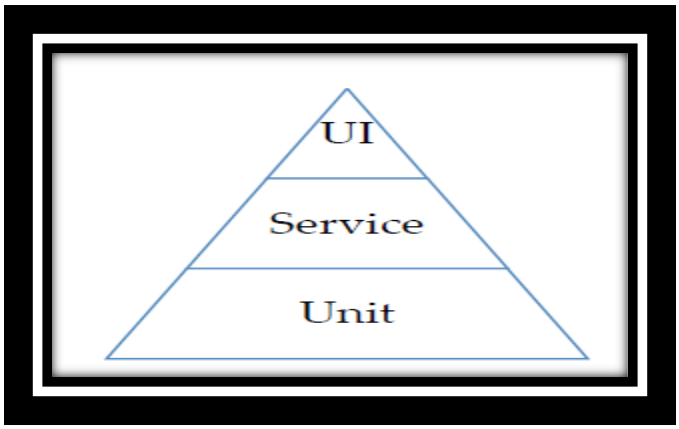


Figure 2. 2 Automation Pyramid which divides automated tests in to three levels [30]

Unit test (discussed in above) forms the base of test automation and are also the foundation of the test-driven development. Service layer represents testing of a system on integration, service or API level (without user interface). Mike Cohn especially marked service tests that call APIs which are regularly used by the UI (view) layer. An example of automated tests on this layer are tests on view-models , which can perform single tasks regardless of the application user interface layer that is bind to the view-model properties.

UI tests, which simulate user interaction with the software through the graphical user interface, are placed at the top of the pyramid. These tests are poorly maintainable and their creation is time consuming. UI test scripts can be written manually or recorded by the special software which speeds up their creation. Examples of UI automation software for mobile applications are SeeTest Automation or Ranorex.

2.5. Continues Integration

Martin Fowler [31] has described continuous integration (CI) as follow: “Continuous Integration is software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible.”

CI is practice comes with the agile software development. It gives faster feedback to the developers and leveraging the test automation. It is widely used among the organizations with matured development process, which are typically use a build server for implementation of the CI. The continuous integration poses a challenge for the mobile application development, because many automated tests have to run on real devices. Therefore, application deployment and installation on the device should be also automated. The main tasks of build server role for mobile application development (based on [32], [33]) are:

1. Download the source codes from the repository.

2. Compile and inspect the source codes, create new binaries.

3. Connect to the device or emulator.

4. Execute the unit tests.

5. Set up external dependencies (databases, services, etc.).

6. Execute higher level tests (integration, service or UI tests).

7. Create the application package.

8. Create the report.

After application passes all the tests run by the build server, application package along with the UI tests can be sent to the device cloud. Running the test on the device cloud is expensive and often

limited by maximum device testing hours and device reservation timeslots. Thus, automated tests on device cloud are not executed so often.

2.6. Mobile Application Qualities

Every application in any application store or marketplace has tens of direct competitors. To find an application that provides really unique functionality is almost impossible. Therefore, most applications fight for their users with quality. If the application does not match with user expectations on quality, it can be simply replaced by another application in matter of seconds. Testing increases software quality by finding defects, which are consequently fixed. Software quality is defined by ISO/IEC 25010 [34] as: Software quality is the degree to which the software product satisfies stated and implied needs when used under specified conditions. ISO/IEC 25010 also describe quality model which defines eight software internal and external quality characteristics.

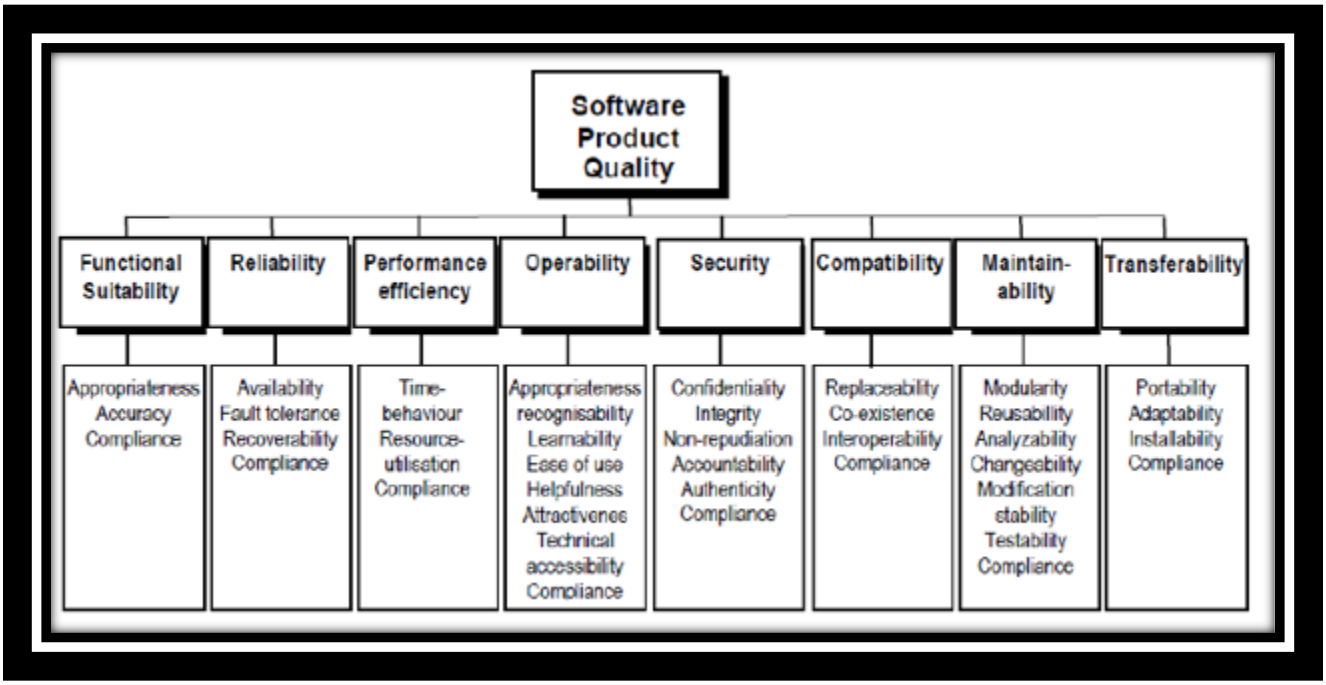


Figure 2. 3 Software Product and Quality Model (Eight internal and external quality characteristics of software)

This section mentions five quality characteristics: reliability, transferability, security, usability and performance. These characteristics were chosen because they require specific testing approach for mobile applications [22]. Moreover, they are considered as the most important for mobile application success as implied from several publications [35] [36] [37].

2.6.1. Reliability

A. Interruption

Mobile application lifecycle differs from classic desktop one where operating with several open applications at the same time is normal (as well as minimizing, maximizing, switching between applications or running in background). Same operations are executed on mobile applications, but they are handled differently due to performance restrictions (such as low memory or battery power). Managing mobile application lifecycle in code is difficult and switching between application states (e.g. active, inactive, suspended, background) is potential source of faults. Moreover, mobile applications can be forced to immediately change its state (e.g. from active to background due to incoming call). To achieve the best reliability of the product, application under the test must be tested on various types of interruptions.

- ✓ Incoming call, SMS or MMS.
- ✓ Alerts or notifications from other applications or services.
- ✓ Loss of power, alerts on attaching or detaching charger.
- ✓ Cable insertion or removal for data transfer, Bluetooth triggered events.
- ✓ Network outage and recovery alerts.
- ✓ Media player interruptions.
- ✓ Lock, home or back button pressed.
- ✓ Auto lock.
- ✓ Memory card or SIM card insertion or removal.

Testing application on every interruption in any time or any application state is impossible and unnecessary. Preferable procedure is to identify areas or situations where the application may crash after unpredictable interruption. These situations could happen when the application under the test:

- ✓ Handles high-performance task.
- ✓ Operates with sensors or camera.
- ✓ Switches between pages.
- ✓ Communicates over the network.

Test automation using mobile device cloud can accelerate this testing technique and increase the probability of bug exposing.

B. Networks

A lot of mobile applications depend on the Internet connectivity. Moreover, some of them are only clients consuming web services. Week signal strength or switching between connection types frequently cause failures. Network based testing investigate misbehaving of an application that uses network connection and then, ensure the reliability of a product.

2.6.2. Transferability

Transferability issues are the reasons why device test clouds exist. Almost every application tries to support large variety of devices to reach as many users as possible. However, platform and

device fragmentation makes this task harder. If the user lost the data from application after OS update, if will certainly appear on the store as negative review.

A. Installation

Installation testing checks if the application is installed correctly on device under specific conditions. Scenarios to test

- ✓ Installation on internal device memory.
- ✓ Installation on SD card with different settings (e.g. mass storage mode).

Related testing

- ✓ Moving application from internal memory to SD card and vice versa.
- ✓ Uninstall application from internal memory and from SD card.

These testing scenarios are suitable for automation on the device cloud. Simple smoke test can assure that the installation of the application on many different devices with different OS versions will be successful. Note, that there are scenarios in which the application should not be installed. For example when the device does not have needed capability (e.g. missing NFC for contactless payment application).

B. Operating System Update

Mobile application developers get new versions of OS in advance to have a time to prepare their applications. OS updates are common cause of application crashes [38]. Test application after OS updates can be convenient on mobile test cloud, but many of them do not support test flow when application is installed on device with specific OS version and OS is consequently updated on higher version.

2.6.3. Security

A. Data Storage

According to Hewlett Packard research, 75% of applications do not use proper encryption techniques when storing data or leave them unencrypted [39]. OWASP (Open Web Application Security Project) describes three common reasons for data leakage [40]:

- ✓ Developers often assume that no one will have access to application local data.
- ✓ Developers are not fully conversant with the operating system and unintentionally save the data to shared locations.
- ✓ Usage of week cryptographic algorithms.

Basic test for data leakage is application footprint analysis, which contains these steps [41]:

1. Acquire rooted device (to be able to access all folders and files).

2. Create a file system backup for future reference.

3. Setup directory monitoring tool to monitor mobile application folders.

4. Install the application.

5. Check created files, compare modified files with pre-installation backup in CompareDiff or TotalCommander.

6. Use the application or run the automated UI tests.

7. Check created files, compare modified files with post-installation backup.

Folder monitoring tools can show all newly created or changed files. Subsequent analysis of these files verifies that all data was stored in proper locations and no sensitive information was stored unencrypted.

B. Transport Security

Cyber security company FireEye conducted a research on 1,000 most downloaded applications on Google Play. They found out that 73% of applications, which use SSL/TSL to communicate with remote server, do not check certificates and 77% of an applications, which use WebKit, ignore SSL errors. Disrespect of basic security principles makes the application vulnerable to man-in-the-middle attack [42]. Previously mentioned study from Hewlett Packard states that 18% tested applications send user names and passwords over HTTP. It is important to note that lot of security issues of this type could be produced from third party libraries (e.g. analytical or advertising) [42]. Transport security testing should assure that the data are transported securely with proper use of cryptographic protocols.

C. Binary Protection

Binary protection is one of the most important and most underestimated part of mobile application security [39]. Android and Windows packages (APK, APPX) are actually ZIP archives and can be very easily downloaded from the stores and unpacked. Apple protects applications on the App Store with an encryption (Apple FairPlay Digital Rights Management) but this security feature could be very easily bypassed if the attacker has jail-broken device.

Moreover, most of the Android and Windows applications have been developed in Java or C# respectively. These languages are translated to the intermediate languages which are higher abstraction than machine code with metadata, making it easier to decompile. iOS source codes are compiled to native machine code with many optimizations. However, there are many tools developed specially for decompiling iOS applications binaries. If the attackers manage to decompile binaries to readable source codes, they can find there vulnerabilities or information for another attacks. Applications, which do not have properly secured binaries, can be modified and published in the unofficial applications stores. However, protect binaries from reverse engineering can never be perfect, but techniques like code obfuscation, jail-break detection or checksum control can make reverse engineering complicated [43].

2.6.4. Usability

About 26% of users abandon the application after first use and one of the main reason why users tend to immediately delete them is poor user experience [22]. Usability is quality in use defined by the ISO 25010 [38] as: The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments. ISO 25010 also describes usability sub-characteristics likability and pleasure, which represent cognitive and emotional satisfaction. On the mobile application market, these attributes are more important than we are used to on desktop applications, because users very often leave reviews on usability on application store.

2.6.5. Performance

Performance requirements are also very important for mobile application market success. The most important key performance indicators are application launch time, UI responsiveness, memory footprint and battery life [44].

A. Launch Time

Launch time refers to the time delay when application starts. According to the consumer expectations survey [37], the median of expected mobile application launch time is two seconds and 80% of respondents expect that the launch time will be three seconds or less. The App Quality Alliance (AQuA) required to show message or progress bar if the launch time is longer than five seconds. The launch time should be measured by automated test script [44].

B. Responsiveness

Responsiveness means that the application GUI is able to react quickly without delays or freezes. Mobile users are always in hurry and nobody wants to wait for slow application. Application responsiveness is measured in milliseconds as time delay which takes rendering different application pages or controls. Delays between switching pages can be measured in code, by GUI automation tools or by the tools dedicated specially for this purpose.

C. Battery Life

Application should not overload CPU or network because high CPU or network load decrease the battery life. The best way how to measure battery life is to measure CPU, GPU and network usage with a special tools which can count how much the application drain the battery.

D. Network Usage Efficiency

Network usage efficiency is one of the main performance criterion for mobile applications by the App Quality Alliance (AQuA). AQuA defines three basic network usage tests:

- ✓ **Duplicated content** – application should cache frequently used downloadable content.
- ✓ **Periodic transfers** – advertisement or analytics data may not overuse internet connection.
- ✓ **Closing connections** – network transfers should be closed when the application is deactivated.

2.6.6. Application Store Compliance

Certification testing is the final procedure before the application is uploaded on the application store. Tester have to assure that the application comply with all platform specific guidelines and will be approved to application store. This is very important mainly in the case of iOS where approval process takes the longest time and any guidelines breach can significantly protract application release.

2.7. Usability

Usability is a measurable characteristics of product user interface and a measurement for the quality system. It mainly focusses on motor capabilities, cultural usage and mental model for users and how well and how easily a user, without formal training can interact with an information system of interface. The element in usability are learnability and efficiency, aesthetics and navigation, content and functionality, accuracy and consistency, technical adequacy, help and documentation and error removal.

Usability is the extent to which the product can be used by a specified user to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use (ISO (9241-11)). From this stand point we can incorporate three critical points: -

Specific users: - specific users not just any users for whom the product is designed.

Specified goals: - these specific users have to share the goal for the product meaning that the products goals represent their goal.

A specific context of use: - the product has to be designed to work in the environment in which these users will use it.

Usability is a quality that many products possess, but many more lacks. Usable refers to the usability of a given product it is more than useful. It examines the way that the product will be used and whether it enables the user to do so in a satisfy able, simple and effective manner.

David McQuillen in Darwin magazine says “usability is about human behavior, it recognizes that humans are lazy, get emotional, are not interested in putting a lot of effort into, say, getting a credit card and generally prefer thing that are easy to do versus those that are hard to do”.

Many “useful” products fails to be “usable”. A usable system: -

- ✓ Is accessible to all the people who may want to use it
- ✓ Always lets us to know where we are, what is going on and where we can go next
- ✓ Is well-behaved logical, consistent and predicable
- ✓ Helps us to avoid errors and give ways to recover gracefully from our mistake

2.7.1. Usability Testing

When we talk about usability testing, we mean that the activity that focuses on observing users working with a product, performing tasks that are real and meaningful to them. So, usability testing can be: -

Formative testing: -while the product is in development with a goal of diagnosing and fixing problems typically based on small studies repeated during development.

Summative testing: - after the product is finished with a goal of establishing a baseline of metrics or validating that the product meets requirements. Generally, requires large numbers of statistical studies.

Why does usability test is important?

How well a product looks usually determines how well it works. There are many software applications which miserably fail, once launched due to the following reason: -

- ✓ Where do I click next?
- ✓ Which page needs to be navigated?
- ✓ Which icon represents what?
- ✓ Error messages are not consistent or effectively displayed
- ✓ Session time not sufficient

Usability testing identifies usability errors in the system early into the development cycle. Generally, usability testing is important because of the following reason: -

- ✓ It helps to find usability issue before the product is marketed.
- ✓ It helps to improve end user's satisfaction.
- ✓ It makes the system highly effective and efficient.
- ✓ It helps to gather true feedback from target audience who actually use the system during usability test.

Important guidelines for the developer when the application reaches at testing stage before released to market which is used for native application, web application and hybrid application is: -

- ✓ check each and every action gets the responses from the application within a second
- ✓ check exit functionalities of the application while it is being run either in foreground or background
- ✓ check the menu and the sub menu response in the screen of the app
- ✓ check the navigation is easy across the applications
- ✓ check the application representation with respect to screen resolution
- ✓ check single selection and multiple selection dropdowns are responsive
- ✓ check checkbox and radio button are clickable and responsive
- ✓ check textbox and text area are clickable and allow the user to be manually type the text within the area
- ✓ check all buttons have clear and readable across the application
- ✓ check the button size is minimum and suitable for fingers
- ✓ check the same button is placed at the common location in all the related screen of the application
- ✓ check the color of the button is the same in all related screen of the apps to ensure consistency of the button
- ✓ check the active and in-active buttons are distinguishable from each other

- ✓ check the error and warning messages are prompted properly
- ✓ check zoom-in and zoom-out features in the application
- ✓ check the text is simple, visible and clear across the application
- ✓ check enough spacing is present between the elements (paragraph, sentence) in all the screen of the apps
- ✓ check the opening and closing of the applications in all possible present stage
- ✓ check the language is correct across the applications. Spelling and grammatical mistake should not be present
- ✓ check the application synchronize according to user's action taken
- ✓ check the content is precise and clear
- ✓ check the application is displayed in user chosen language
- ✓ check the navigation of the app is through notification
- ✓ check the user data is stored and accessible next time if the apps has stopped due to unexpected errors

2.7.2. Usability Testing Mechanism for Mobile Applications

In order to conduct usability testing of mobile application the developer can use either: -

A. Laboratory based usability testing

This is the widely popular method to perform usability testing for mobile applications. Here the individuals whose profile matches with the target users are selected to perform usability testing. This is the controlled activity and the facilitator would be in-charge of the entire phase. In this testing method the selected individuals are bounded with: - **Recording software: -** To record the moves within the applications which helps in analyzing the ease-of-use of the apps. **Webcam: -** To record the facial expression while using the apps. This helps in analyzing which features of the apps is user-engaging. **Test-Location: -** A proper ecosystem where the test cans be conducted successfully. It has to be equipped with proper infrastructure, complementing tools and resources to carry out the tests effectively, accurately and reliably. **Test-Facilitator: -** To facilitate the testing process and to ensure that the test run smoothly by addressing any issues that the test participants may have with the task being assigned or device they are using. He/she should be able to handle extreme situation and adverse out comes during the entire phase.

B. Field Based Usability Testing

Usability testing for mobile application can be also performed by using field-based technique. However, this activity most likely time consuming than laboratory testing. It may also require extra effort from test users and the test leader/tester.

According to ISO 9126-1, usability is defined as the capability of the software product to be understood, learned, operated, attractive to the user, and compliant to conditions" [45]. Some researchers has combine ISO 9126 and ISO 9241 attributes and develop new model that has effectiveness, efficiency, satisfaction, learnability and security as attributes [46]. Usability

standards provided by (ISO) the International Standards Organization can be broadly classified into two categories first, product-oriented standards (ISO 9126, 2001; ISO 14598, 2001) and second, process-oriented standards (ISO 9241, 1992/2001; ISO 13407, 1999).

Usability is a product attributes that give impact or influences on the quality of a software system. Usability models are conceptual view and not only states the characteristics but also indicates how those characteristics fit together. There are several usability models such as:-

- ✓ Eason Model (1984)
- ✓ Shackel Model (1991)
- ✓ Nielsen Model (1993)
- ✓ ISO 9241-11(1998)
- ✓ ISO 9126 (2001)

Eason Model (1984)

This model is proposed by Kenneth Eason (1984) and published in an early issue of Behavior and Information Technology. Eason Model has 3 aspect, **task**, **user** and **system. For task** it has 2 sub attribute that is **frequency** and **openness**. **User** has 3 sub attributes that is **knowledge**, **motivation** and **discretion**. **System** has 3 sub attribute that is **ease of learning**, **ease of use** and **task match**. Eason Model cannot measure usability without considering users and their target task. **Based on Eason's model**, it has two parts that is **input** and **outcome**. The input for Eason's model is user, system, task characteristic and can define as independent variables. The outcome is user reaction and it dependent variable. Eason model is causal type of model because it has input that is independent variable and outcome or result that is dependent variable. A causal model is one that makes prediction about causality. Eason model sees usability as the result of several interacting variables or "multi - variety". [47] [48].

Shackel Model (1986)

Shackel Model was developed by Brian Shackel. It has 4 attributes that is **effectiveness**, **learnability**, **flexibility** and **attitude**. Shackel Model does not weight the dimension, recognizing that the importance of each of these may different from project to project. Shackel model emphasizes measurement of a number of human factors, relating to human performance and attitude [47] [48]. Booth Modified Shackel model and adapted the model into **usefulness**, **effectiveness**, **learnability** (or ease of use) and **attitude** (or likeability).

Nielson Model (1993)

Nielson Model was developed by Jakob Nielson. It has 5 attributes that is **easy to learn** (learnability), **efficient to use** (efficiency), **easy to remember** (memorability), **few error** and **subjectively pleasing** (satisfaction). Nielson Model focus on acceptability that mean if the system

is not useful such as did not meet the user requirement, it will not accept it either it usable or not. The model is based on user interface usability in the context of a software engineering project. Nielsen emphasizes usability as part of a larger set of system characteristics. Same with Shackel Model, Nielson Model also does not weight the dimension, recognizing that the importance of each of these may different from project to project [47] [48].

ISO 9241 – 11 (1998)

ISO 9241 is an international standard for guidance on usability based on process oriented. ISO 9241-11 explains the benefits of measuring usability in terms of user performance and satisfaction. It emphasizes that visual display terminal usability is dependent on the context of use and that the level of usability achieved will depend on the specific circumstances in which a product is used. The disadvantages of this model is too abstract and does not tackle the learnability metric, as recommended by majority of standards and experts [49]. Has three attribute that are:-

1. Effectiveness
2. Efficiency
3. Satisfaction

ISO 9126 (2001)

ISO 9126 is an international standard for the evolution of software quality model from the product perspective. ISO 9126 is an extension of previous work done by McCall (1977), Boehm (1978), FURPS and others in defining a set of software quality characteristics [50]. ISO 9126 divided into 4 parts which address respectively to the quality model, external metrics, internal metrics and quality in use metric. The internal and external metrics are functionalities, reliability, usability, effectiveness, maintainability and portability [51]. ISO 9126 has 5 attributes such as understandability, learnability, operability, attractiveness and usability compliance [50]. The advantage of ISO 9126 model is it provide a framework for making trade-offs between software product capabilities and the attribute are applicable to any kind of software including computer programs and provide consistent terminology for software product quality. The disadvantage of ISO 9126 was unclear architecture at the detail level of the measures, overlapping concepts, lack of a quality requirement standard, lack of guidance in assessing the results of measurement and ambiguous choice of measures [52].

OTHER USABILITY MODEL

Quality in Use Integrated Measurement (QUIM) (2006)

QUIM or Quality in Use Integrated Measurement developed by Ahmed Seffah et al in 2006. QUIM is a consolidated model for usability measurement and metrics. QUIM model brings together usability factors, criteria, metrics and data mentioned in various standards or model for software quality and defines them and their relations with one another in a consistent way [10]. QUIM model used 4 attribute that are Effectiveness, Efficiency, Learnability, Satisfaction and also include Accessibility. This model also based on ISO standards and previous research in the area

usability and quality in use. It combines various standard and model such as ISO 9241 and ISO 9126 and unified into a single consolidated, hierarchical model. It outlines methods for establishing quality requirements as well as identifying, implementing, analyzing, and validating both process and product quality metrics. This model appropriate for novice users that have little knowledge of usability and can be applied by usability experts and non-experts. QUIM model consists of 10 factors and subdivided into 26 criteria or measurable criteria, and finally into specific metrics consists 127 specific metrics. The 10 factors consists Efficiency, Effectiveness, Satisfaction, Learnability, Productivity, Safety, Trustfulness Accessibility, Usefulness and Universality. The model is used to measure the actual use of working software and identifying the problem. The limitation of this model, it is not optimal yet and needs to be validated [53] [54]

Booth Model

Booth model has 4 attribute that is usefulness, effectiveness, learnability (or ease of use) and attitude (or likeability). Booth did not include flexibility because he thought difficult to specify and measure the flexibility of a system and include useful to be consider as fundamental to usability. J. Rubin and D. Chisnell said that definition with one or more of four criteria in Booth model are generally accepted by usability community. Likeability also important to usability which constitutes user's perception, feelings and opinion of product [48].

McCall's Model

McCall's model (also known as McCall's triangle of quality) is one of the software evaluation models from the early 70's. It provides three different perspectives of software quality or property according to the major three processes in software life cycle: product operation (basic functionalities), product revision (ability to change), product transition (ability to adopt new environment) [47]. The usability characteristic in this model, is under product operation processes that contain sub-factors such as operability, training, and communicativeness.

Chapter Three: - Related Works

In this section we discuss literature that deals with the usability of mobile applications, and we summarize a selection of the most relevant findings. To start, in the ISO/IEC 9241-11: 1998 (E) standard, usability is defined as "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". List of related literatures to our work which are done before, is reviewed and presented as follows: -

Towards usability guidelines for mobile applications and website is developed by Maria S [7]. In this scenario, the researcher is tried to identify existing usability guidelines for mobile applications. For this purpose, the researcher conducts a literature review focusing on the results of empirical usability experiments dealing with mobile application and mobile website. Definition of the review scope, conceptualization of the topic, literature search, literature analysis and driving agenda is the contribution of his work. However, the proposed usability guideline is not representing mobile applications variance and also the guidelines is not examined by conducting testing.

For growing the understanding of the impact of contextual component of mobile device usability, methods for usability evaluation of mobile device is developed by Fayaz A and Mark F [8]. Their work provides the way in which mobile device usability evaluation methods are being adapted to better reflect technology. However, the usability evaluation is only for mobile device it doesn't serves as a method for usability evaluation of mobile application.

Works presented by Junnu P [68] evaluates the usability of the developed apps (AchSo). Also try to identify usability issue within the apps and study the suitability of the apps. However, types of usability attribute he used for evaluating usability of AchSo app and UX during evaluation is missed. Usability Evaluation tool for android based mobile application also developed by Babita S and Meena S [68]. Both of them are tried to develop usability analyzer tool that helps in software development organization to calculate the usability factor and measure the software usability from the real users view point and produce the accurate result of software usefulness based on the feedbacks given by the real users. However, Usability evaluation is only for android based mobile apps and the usability analyzer tool considers only web based android apps it doesn't serves as usability analyzer tool for many other android mobile applications like native based and hybrid based android mobile apps.

Usability testing on android based applications is developed by Aruna A [5]. This research project finds existing usability faults and propose solutions to enhance their usability. However, Testing process is including only technical background participants not include non-technical background users. Testing process focuses only basic features of mobile apps Advanced features of mobile apps are missed. In this work users are given tasks to perform using the product and observed to see if they have any problems performing the tasks. The mediator is the person who conducts the usability test session. During the test session the mediator will give some tasks to the users and asked them to perform the tasks on the application he/she want to test. While user is performing the task, mediator will observe the user how he/she is navigating in the application and the facial expression of the user to find out how he/she is reacting to the application. While mediator is conducting the test session other persons (called observers) can also observe the user. Facial expressions of the user and the phone screen will be video recorded as mediator or the observers cannot catch all the expression during the test session. The mediator and/or the observers will go through the videos thoroughly after each test session and note down the observations. Depending on the observations, moderator/usability team will suggest design solutions. The following applications are selected by the researcher for doing their study like: -

- ✓ Calculator application
- ✓ Weather application
- ✓ Calendar application
- ✓ Maps application
- ✓ Gallery application
- ✓ Dolphin browser application

In general, much work has been done to assess the quality of mobile applications, many researchers are still attempting to figure out the most important reasons why some applications fail. Therefore, usability of mobile applications is still in the initial stages of development, and there are many issues that need to be resolved to reach a high level of usability—in terms of efficiency, effectiveness and satisfaction. With a significant increase in the use of mobile application by individuals, the level of usability and its related issues must be addressed thoroughly. Typically, people usually carry their mobile phone with them and frequently interact with the apps. Therefore, the fact that individuals are so connected to their mobile phone can also be a factor that increases the interaction between apps and users— which can in turn increase the quality of the using

experience for users. This forces us to pay a lot more attention to the usability issues of mobile application. In general, in this work we tried to address: -

- ✓ Significant of testing mobile applications from usability perspective
- ✓ Mobile application design guidelines
- ✓ Icon design consideration and process
- ✓ UID principles and guidelines considering users as well mobile apps difference
- ✓ Characteristics of mobile phone that helps to identify obvious limitations for using mobile apps
- ✓ Procedures during usability testing of mobile apps
- ✓ Significant guidelines for the developer when the app reaches at testing stage
- ✓ Usability testing mechanisms
- ✓ Core usability measurement attribute for measuring usability of mobile apps including metrics for each attribute
- ✓ Developed usability testing process iteration framework which is significant for designing and developing usable mobile apps for all types of users and mobile apps

Chapter Four:

4.1. A Usability Testing Process Iteration Framework for Mobile Applications

4.1.1. User Interface Design (UID)

The system must adapt to the user this is why design principles are so important. Therefore, before we talk about mobile application testing, mobile application design guidelines are important for the developer. Therefore, the developer should keep in mind when designing the mobile apps. If the application is not usable, it has no real value for users and no one has any reason to use it. Even if the app is useful but also requires a lot of time and effort, as we describe in the above useful product may fail due to its usability case. Therefore, users would not bother learning how to use it because users prefer things that are easy to do and to learn.

Interface design principles represent high level concepts and belief that should be used to guide the software developer. Therefore, the developer should determine which principles are most important and most applicable for his/her system and then use those principles to establish the apps and also determine design decisions. Well-designed software interface should build a close relationship that guides users to learn and enjoy what they are doing. Good interface can even challenge users to explore beyond their normal boundaries and stretch their understanding of the user interface and the mobile apps. The developer should have an understanding and awareness of the user's abilities. Users have come to know and love their mobile applications. They expect the apps that runs on them to look a certain way and behave a certain way. Follow the guidelines presented in this section and the apps will look and feel like a usable application. Users will learn it more quickly and the developers will be well on their way to designing an application that the users love and recommend to their friends and colleagues.

Actually, principles are not mean to be follow blindly rather; they are serves as a guiding light for sensible interface design. User interface design concerned with the design of visual tools that users need to interact with a product. This aims to create a visual language. This visual language is made up of a user interface element like: - input element, navigational element, sharing element and information element. User interface also encompasses the design of interactivity. Mean that what a user interface element may do when a user clicks, taps, swipes or interacts with it in any way.

In order to design the apps first the developer should keep in mind three things. First, he/she must place users in control of the interface. Second, he/she must reduce user's memory load. Third, he/she must mind interface consistency. We are tried to put some design criteria (which considers user as well as mobile app differences from usability perspective) for the mobile app developer when he/she is designing an app. Those criteria are: -

1. Designing Screens

As a UI developer, he/she will be in charge of designing each screen a user will interact with. This means designing everything from a screen's layout and the different UI elements like: - input, navigational, sharing and information elements (toggles, lists, buttons and patterns) used on each screen.

2. Designing Interaction

The developer should give careful consideration to the interactivity of an interface. He/she will need to design what elements do when users interact with them.

3. Designing Icons

Icons have more popular and important in interface design. Icons should not only be beautiful; they should also communicate their meaning clearly. It is a graphics symbol which convey meaning and essential pictures of things. We see them everywhere not only in UI design. They can communicate vast amount of information in short time. According to **Marina Y**. there are different icons from its functionalities and visual performance perspective like: -

- ✓ **Clarifying icons: -** the developer can use these icons to explaining particular features or marking out categories of content.
- ✓ **Interactive icons: -** as the name indicates it is directly involved into interactions. So, the developer can use this icon for navigation because they are clickable or tap able and respond to the users request during the action symbolized by them. He/she also use this icon when designing the apps in order to inform users about the functions or features behind the buttons, controls and any other elements of interaction.
- ✓ **Decorative and entertaining icons:** - the developer can use this icon for considering the style and appearance corresponding to the target audience preference and expectation.
- ✓ **App icons:** - the developer should use these icons to represent interactive brand signs that present the apps on different platforms supporting the original identity of the digital product.
- ✓ **Favicons: -** the developer can use these icons to represent the product or brand in the URL-line of the browser as well as in the bookmark tab. Because, it helps the user to get a quick visual connection with it when they are browsing.

- ✓ **Glyph icons:** - the developer can use this icon to apply simplified and universal shapes plus to give recognizable and flexible images in terms of responsive design. It helps the developer to give emphasis for navigation for a digital product.
- ✓ **Flat and semi-flat icons: -** the developer can apply color combination, filling of elements and present a bit more complicated images by using these icons. He/she can also use this icon for simpler and recognizable visual metaphors for quickly transfer the required meaning.
- ✓ **Skellomorphic icons:** - the developer can use these icons to reflect images in 3D look for the purpose of game design.
- ✓ **SVG icons: -** the developer can use these icons to build XML based 2D vector images.

In general, when the developer used all of the described icons in above those icons are: -

- ✓ Must be understandable and accessible to the end users
- ✓ Must be transfers informative value
- ✓ Must be presented in the form of recognizable and interpreted correctly by the user
- ✓ Must be quickly perceived and understood without too much effort
- ✓ Must be stands out among other similar elements of the interface and real for the app icon
- ✓ Must be saves its unity, integrity and legibility in different sizes and resolution
- ✓ Doesn't have hidden message or meaning which could feel offensive or rude for any parts of end user
- ✓ Must have the same stylistic concept of the layout it is applied for
- ✓ Must have known when the end users see

Icon design has a process that the developer will follow. We are trying to brake icon design process into three main stage: -

- ✓ **Exploration: -** before the developer start drawing out the icon, he/she must give some time to explore. Spend some time thinking about the idea or the object that he/she is designing for and all the images or symbols directly associated with what he/she need to communicate.
- ✓ **Design: -** once he/she has selected an idea or object to work with, begin sketching pencil and paper work wonders and are a quick way to explore alternatives fast.
- ✓ **Testing: -** finally, he/she will need to test the icons and make sure that they are scalable and read clearly at different sizes. They should also align with the interface in which they will be used. Pop them into the UI and actively look for issues. He/she also go back to the drawing board as needed. Revise the icons based on his/her findings and test again until they are just right.

4. Designing UI Style Guide

The developer should follow sets of standards for the design of an apps, websites or products. He/she ensures consistency and the proper implementation of a visual language across an entire product.

5. Simplicity

The developer should apply simplicity. Because, simple interface helps the different mobile application users to work easily, to learn easily and less likely to create opportunities for error than complex ones. The developer can reduce complexity by applying simplicity. There are different complexities like: -

- ✓ **Visual complexity**: - The developer can reduce this complexity by putting only the most important controls in buttons or control panels and hiding the rest in menus. The menu controls are still available but they don't clutter up the display.
- ✓ **Verbal complexity**: -The developer can be reduced this complexity by studying and employing the words and concepts that the users of the application use. Use words common in the user's vocabulary. Positive, active words are most appropriate.
- ✓ **Task complexity**: -The developer can reduce this complexity by grouping related tasks together and by reducing the number of different tasks presented in a single window. Mean that provide only a few alternatives at a time and provide only choices that are valid - appropriate actions should be constrained. Provide clearly marked exits for the user.
- ✓ **Conceptual complexity**: -The developer can reduce this complexity by helping the user form a clear model of how the application is organized and how the different components can be accessed.

6. Conceptual consistency

The interface should be based on a clear and consistent conceptual structure; that structure being revealed to the user through the design and behavior of the interface components. A well-structured interface organizes and helps the user's activities. It achieves this by presenting objects and actions appropriate to the tasks being performed and by indicating the flow of activities required to carry out a task. In some cases, the interface may explicitly restrict the user to performing certain tasks in a certain sequence. In other cases, the design will permit the user to perform tasks in parallel or in a number of different sequences.

If the developer apply consistency in his/her interface design therefore, it supports positive transfer of training and reduces the time required to learn a new application. It also supports users to know how to perform a certain task or operation in one application and they should be able to perform a similar task in a different application without retraining. Some basic principles of consistent design which helps for the developer that he/she will apply in his/her interface design are: -

- ✓ Similar interface components should have similar appearance and behavior
- ✓ An action should always have the same result regardless of context
- ✓ Standard functions should be reused across tasks and should be presented in the same way in each task

7. Design for error

The best treatment for errors is prevention. However, errors will happen even in the best interfaces and with the most skilled and experienced users. Therefore, the developer should minimize the

consequences of errors and helping the user to recover from them by applying the following technique in order to assist the users. These techniques are: -

- ✓ **Provide clear error messages**: - Error messages should state the problem in the user's terms, not the systems. Telling users that Error- occurred, telling them that the network is down and the like. Error messages should also indicate appropriate actions for the user to take to recover from the error or correct the problem that caused it.
- ✓ **Provide undo mechanisms**: - This allows the user to reverse an unintended operation before proceeding to the next one.
- ✓ **Force the use of explicit destruction**: **-** apply this when an operation can't be undone. Means asking the user to take an extra step to perform operations that result in permanent alterations to data or other important system assets. The most common technique is asking the user to confirm operations.

8. Provide good feedback

Since, feedback is the application's way of communicating with the user. When the developer applies these principles for interface design it will help users to confirm that the application has received their input. It also informs the user about the status of a task and the state of objects in the interface.

9. Support different interaction style

There's usually more than one way to perform a task, and different users are likely to prefer different techniques. Mean that the developer can apply these principles to provide many ways to select objects and actions for novice users, provide shortcuts for expert users and minimize the number of keystrokes.

4.1.2. Usability Testing Framework for Mobile Application

According to Neilson and other usability experts, usability testing evaluates the application based on usability model attributes for the target audience like learnability, efficiency, effectiveness, satisfaction. There are also obvious limitations for using mobile applications due to the characteristics of mobile phone like: - small screens, different display resolutions, mobile context, multimodality, connectivity, limited processing capability and power and restrictive data entry method.

Therefore, measuring mobile application is a more important issue for mobile phone and it is very important to get usability testing in place from the early stage of the application design and it should not be done only when the application is completed. Usability testing requires involvement of users and the output may affect the application design which is very difficult to change in the later stage of the project.

Therefore, usability testing process iteration framework for mobile application is all about validating the user interaction with the mobile application and ensuring the great user experience of it. It is all about testing the sensitivity and comfort of the user of the application.

4.1.3. Methods, Metrics & Proposed Usability Testing Framework

The usability test must set up in a way that is a close to the normal context as possible. Testing in a usability lab offers some benefit because it can be designed to create the ideal testing environment by using specific equipment for recording the session, controlled logging software, special equipment for testing the mobile apps. On field and remote testing is much closer to the normal context of used and requires special apps that have the ability to automatically collect user interface events as the user interacts with the apps, to capture screen shoots and eventually make an audio video recording of the user. Even if closer to the real-life use of the final product, testing out of the lab has specific disadvantages.

A. Method

User based evaluating (testing) method are mostly used in a mobile application usability testing. This method involves collecting quantitatively and qualitatively from users while or after running through well prepared scenarios. Users are invited to do typical tasks within a product or simply asked to explore it freely. While their behavior is observed and recorded in order to identify design flaws that cause user errors or difficulties.

During this observation the time required to complete a task, task completion rate, number and types of error are recorded. Users are also asked to complete questionaries' (surveys) that provide useful qualitative data from users. Although the data collected is subjective it provides valuable information on what the users wants.

B. Metrics

Upon review of the measure's relative appearance in the review literature the core constructs for the measurement of the usability appearance to be: -

Efficiency: - the degree to which the apps enables the users to perform tasks in a quick effective and controlled manner or hindering performance.

Effectiveness: - accuracy and completeness with which specified goals in particular environment.

Satisfaction: - the degree to which a product is giving contentment or making user satisfied.

Goal Question Metrics (GQM) has three levels the first level is conceptual level in which goals are identified based on guidelines. The second level is operational level in which questions are formulated in order to asses each goal and the third level is quantitative level in which a set of metrics are developed to provide information in order to answer the question formulated in the second level. The guidelines will used for metrics development. The developed metrics are used to develop two measurement instruments that are task list and questionnaire. These tools are used for usability evaluation of mobile applications in order to obtain qualitative and quantitative data. Therefore, based on GQM model, we construct table that provides a list of usability characteristics, goals, guidelines and metrics that can help to collect quantitative or qualitative data during the usability testing/evaluation process.

Table 4. 1 List of usability quality characteristics, goal, guidelines and metrics for selected usability attribute

quality characteristics	Goal	Guideline	Metrics
Effectiveness	Accessibility	Ease of understanding apps and its content	Time taken to understand apps and the content
	Help	Ease of navigation of help information	Is/not easy to learn how to navigate help topics
	Interactivity	Ease of interaction Ease of use Ease of customization Provide interaction collaboration or sharing option	Amount of interaction required Is/not easy to use Provide/not easy navigation Number of mistakes during interaction Number of collaboration or sharing option
	Time taken effort required	Loading application Time to learn Time to respond Amount of task effort	Time taken to load the application Time taken to learn the application Amount of time taken to respond Amount of task effort required Time taken to complete the task Number of times the user follows the wrong path when attempting a task
	Features	Provision of task related clues Provision of error recovery Provision of help Allow personalization Organize content appropriately Coherent multimedia usage Use of appropriate control Narratives to structure the content	Provide/not task related clues Provide/not error recovery assistance Provide/not help Allow/not cater for personalization Rating scale of content Organization Rating scale of multimedia usage Use/not appropriate control Use/not narrative for content structure Rating scale of narratives appropriately
	Simplicity	Easy to input the data Easy to use the output Easy to install Easy to learn	Time taken to input the data Time taken to display out put Time taken to install the app Time taken to learn the app
	Accuracy	Accurate	

		Should be no error Successful	
Efficiency	Operability		Time taken to start transaction Time taken to select
	Loading time		Time taken to load/initialize apps Time taken to response Time taken to display Time taken to complete tasks
	Compatibility		Performance speed Loading the application in any device
	Features	Support/help Touch screen facilitation Voice guidance System resource information Automatic update	
Satisfaction	Familiarity	Mental model	Use/not familiar mental model Rating scale of user familiarity with user interface
	Consistency	Navigation	Rating scale of consistency navigation across system Easy/not to navigate
	Attractiveness	Use appropriate font style, size and colors	Rating scale on whether the system is attractive or not
	Help	Sufficient help information Organization of help topic	Provide/not sufficient help information Rating scale of usefulness of help info
	Preciseness	Messages precision	Provide/not precise message Rating scale of message precision
	Feedback	Helpful messages Suitability for all users	Provide/not helpful feedback messages Provide/not suitable feedback message for all categories of users
	User guide		Satisfaction with help menu provide Alert message if error occurred
	Structured task		Satisfaction with app design and format Satisfaction with content design and format Satisfaction with logical design of menu buttons Satisfaction with application

	Content		Satisfaction with logical presentation of content Satisfaction with transaction task provided Update requests
	Safety	Safe while using the application	Provide/not comfort for users while using the application
	Attractive	Eye-catching user interface	Is/not users happy with interface Is/not users familiar with interface

4.1.4. Usability testing process iteration framework

Based on testing methods and metrics a usability testing process iteration framework for designing and developing a usable mobile application are designed. Figure [4.2] shows an overview of our suggested usability testing process iteration framework for mobile apps. This framework includes all types of mobiles by considering user's difference. A developer who builds mobile apps or a usability tester who validates mobile apps for usability testing purpose can use our instantiated framework.

The design of the testing framework starts from identifying questions and objectives. This is the initial activity that usability testers need to perform to do a good usability testing. What question do he/she want to answer with the usability test? What hypothesis do he/she want to test with the usability test? Identify what he/she want to accomplish with this test, what he/she is looking for and what he/she want to demonstrate to stakeholders? This basically involves asking a number of questions to the application stakeholders to explore important areas. Therefore, based on the answers given to the question it is now easier to start identifying the goals and what usability metrics one should use to measure them. A survey should be done before choosing the mobile apps to be used for the test. Because, the selected mobile apps should be representative for the majority of the user of the system.

Then selecting the experimented test users is the next stage for testing their own usability issues. Selecting users is a key activity of this initial step. Selecting can be performed according to their demographics (age, where they live, etc.) or psychographics (cognitive background: if they're used to perform the proposed scenarios). This enables the testers to know how the user is interacted with the application. Users are different preference and characteristics and have different abilities by application usages. The user may novice mean that it depends on system features to assist, perform simple tasks and needs learning time. In addition, the user may expert mean that he/she have leverage experience, use recall rather than recognition, fast, needs less help and feedback and like to reduce keystrokes. Therefore, in order for validate whether or not the application accommodates such variances it is necessary to select experimented test users and study what user's characteristics are, in which application users are interested and what a preferred way to present the applications.

After selecting experimented test users' task is given which is performed by test users in order to perform some tasks using the products (mobile apps which is selected for testing purpose) and it helps for tester in order to see if they have any problem performing the tasks.

The tester is the person who conducts the usability test session. During the test session the tester will give some tasks to the users and asked them to perform the tasks on the application he/she want to test. While user is performing the task, the tester will observe the user how he/she is navigating in the application and the facial expression of the user to find out how he/she is reacting to the application. End user should be considered for usability testing, but not the developer. Programmer/tester already knows how the system/product works, because they will use their knowledge acquired from the experience/education during testing, may not result in finding usability defects. The end user of the system/product doesn't know the internal details of the system. He/she uses the knowledge perceived from the environment without learning while interfacing with the system/product. After giving tasks for end users the next process will be decide whether laboratory or field test is required (simulator, emulator and real device is test equipment for laboratory-based usability testing and actual real device is also test equipment for field-based usability testing).

After deciding if laboratory or filed test is required, then it is time/ the next important step is using the suggested usability attribute to be measured, methods and metrics for content-based user interface usability testing of mobile apps that are best for answering the questions and objectives from the user interface point of views. Metrics give a common fact-based description of user/task performance upon which to make informed design decisions. They are important because: make usability recommendations concrete, influence and change option, provide objectivity to design arguments and help repeat and validate design concepts.

The guidelines for the main usability test the setup of the environment and equipment, the preparing of questionaries and all necessary preparation can be done based on the result of the usability test. Data analysis and interpretation is the last step followed by the required changes of the application platforms. Once testers were done all testing sessions, it is time to sit, analyze the information and jump into conclusion. Actually, there is no rule to do this. However, when testers have all the information in place, they need to look for the trends that emerge, make notes of the possible problem and potential solutions.

Teste report should be included in data analysis and interpretation. It enables testers to create each and every time they perform in a usability testing to be stored with any other testing documentation. The test report may be:

Background summary: - brief summary of what they were tested, the material that was utilized and brief description of all their findings and the goal of the session.

Methodology: - should explain how the session were done, the task or scenarios that were tested, the metrices they used and brief descriptions of each segment of users.

Test Result: - summarize all the results from the metrics they have used.

Findings and recommendation: - list all findings it may positive or negative. Positive finding may help to know they were on the right track and negative findings may help to provide proposals to solve them.

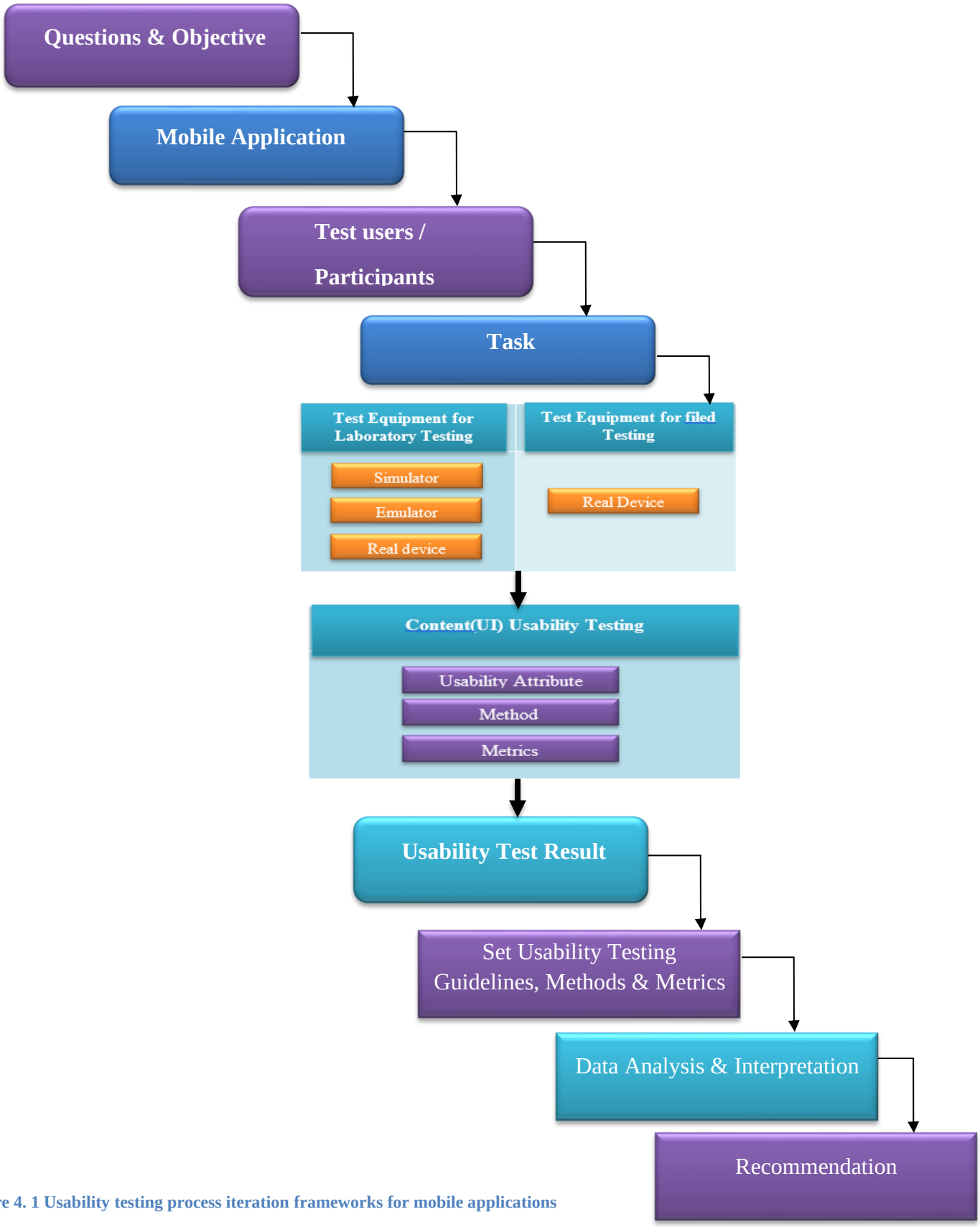


Figure 4. 1 Usability testing process iteration frameworks for mobile applications

Chapter Five: - Evaluation

The overall objective of this work is to improve the usability of mobile applications. Therefore, the objective of the evaluation is to examine whether or not the suggested usability testing process iteration framework helps to improve effectiveness, efficiency and satisfaction of the developed and the developing mobile applications. Therefore, by considering the factors of time and effort 50 participants are invited into the usability evaluation. The participants include a mix of male and female with different level of experience from novice to expert.

In order to conduct testing this study used 5 mobile applications like: Telegram, Messenger, Xender, Amharic dictionary, Bubble shooter. Before the usability evaluation process all the participants are informed of the purpose and procedure of this work and signed a consent form which is present in **Appendix 1**. Their privacy is protected and the data collected are used only in this work. The evaluation was conducted in a quiet and comfortable room in MTU campus and all participants were completed all the given tasks.

In order to obtain qualitative and quantitative data from the participant the questionnaire and interviews are applied after the task is performing. The questionnaire lets the participants evaluate the usability attribute metrics towards different perspectives and angles. Therefore, the data collected from the questionnaire is easy to analyze and ensures the result of the usability evaluation is convincing (definite). As mentioned earlier the metrics are used to develop two evaluation mechanisms that are task list and questionnaire as shown below. Therefore, the overall design of the questionnaire in this work is based on forced-choice questions as shown in Figure [5.1], where a statement is made and the respondent then indicates the degree of agreement or disagreement with the statement on a 5-point scale.

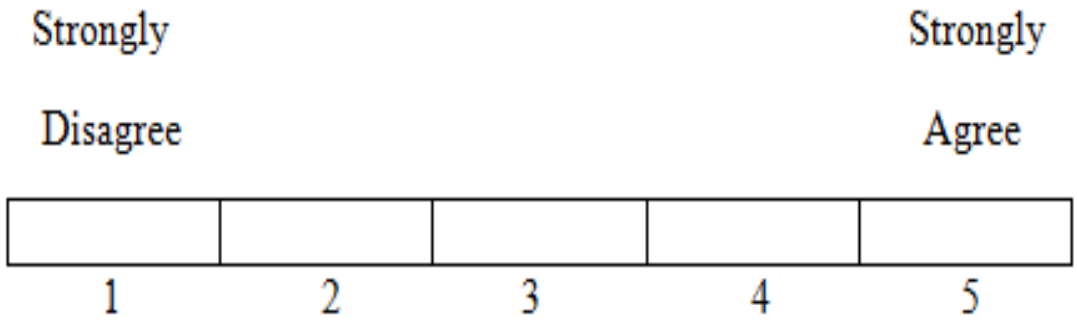


Figure 5. 1 System usability scale used for questionnaire to indicate agreement or dis agreement on a 5-point scale

The statement in the questionnaire is corresponding to the usability attribute of effectiveness, efficiency and satisfaction. The statements and their related usability attributes are shown below in table [5.2] and the questionnaire used in the testing purpose is present in **Appendix 2.**

Table 5. 1 List of tasks which is performed by test users

Task List
❖ **Check for interactivity** ✓ Check of user interaction with application ✓ Check of availability of communication tools ❖ **Navigation activity** ✓ Check of main menu presence ✓ Check for scrolling ✓ Check for hierarchical menu ✓ Check for navigation keys ❖ **Check for adequacy feedback** ✓ Response to input ✓ Check proper response ✓ Check of supportive response ❖ **Check for time** ✓ Loading application ✓ Task: - time taken to complete ✓ Learn: - time taken to learn ✓ Check for time to input the data ✓ Check for time to display output ✓ Check for time to install the app ❖ **Check input/output availability** ✓ Virtual keyboard ❖ **Check for adequacy of help** ✓ Check for task related clue ✓ Check of help icon ✓ Check for navigation of help topic ✓ Check of sufficient help information ✓ Check of organization of help topic ❖ **Check for cognitive load** ✓ Check for suitability of language ✓ Check for suitability of content ❖ **Check for personalization/customization** ✓ Check for availability of setting option ❖ **Check for user control** ❖ **Check for message precision** ❖ **Check of safety while using** ❖ **Check for eye-catching user interface**

Table 5. 2 Statements and their related usability attribute used for the questionnaire

Usability attribute	Statements
Effectiveness	I found that it was easy to understand the apps and its content
	I found that it was easy to learn the app and its content
	The app was easy to navigate and easy to know help information
	The app provides easy way for interaction
	The app was easy to use and easy to customize
	The app took a lot of time to load
	The app is too slow and I had to wait for response to continue
	It was easy to complete the task without too much effort
	I found that it was easy to know what to do in this app
	The app took a lot of time to complete a given task
	The app provides task related clue
	The app provides error recovery
	The app provides useful help information
	It was difficult to find help information
	The app has simple multimedia usage
	The app lacks narrative for content structure
	I found that it was easy to input the data in this app
	I found that it was easy to use the output in this app
	It was difficult to install this app
	The app provides accurate information
	The app gives error message that clearly tell me how to fix problem
Efficiency	The app provides a visual display to show the loading process
	I found that the app was compatible for any device
	The app provides easy touch screen facilitation
	The app provides useful voice instruction
	The app notifies automatic update while updates available

Satisfaction	It was difficult to understand the interface in this app
	The app uses appropriate font style, size and color
	The app lacks message precision
	The app provides appropriate feedback for my action
	I found that the app was suitable to use
	The app provides alert while error happens
	I was satisfied by the app design and format
	The logical design of menu and button is confusing
	The app provides comfort while using
	I found that the app has eye-catching user interface
	I was comfortable with screen orientation of this application
	The app provides clear and understandable navigation keys
	It was difficult to understand the language used in the app
	The app provides different difficult level that I could easily change according to my choice
	The app provides progress report /result for my performance in every activity
	Overall, I was satisfied by this app

Interview

After finishing the tasks and questionnaire we are also tried to invite the participants for an interview. The participants are asked pre-designed questions. The question is shown in table [5.3] below. Q1, Q2, Q3, Q4, Q5 and Q6 are designed to collect data from satisfaction perspective and Q7 and Q8 are aims to collect data from evaluation and comments perspective.

Table 5. 3 Interview questions for test users

Number	Interview questions
Q1	What do you like most in the application
Q2	What do you dislike most in the application
Q3	Would you like to have a feature or user interface improved in the app? If yes, what are they?
Q4	Would you like to have other features which are not available in this app? If, yes what are they?
Q5	What design makes you feeling comfortable when you use this application?
Q6	What design makes you feeling uncomfortable when you use this application?
Q7	Is there any problem when you use this application? If, yes what are they?
Q8	Do you have any suggestion?

Discussion

This section begins with presenting the result of the usability evaluation questionnaire only. In addition to this comprehensive discussion the result is followed including the analysis and potential improvement.

Participant's / test user's information

Users	Gender	Age Group	Occupation	Types of Phone
5	Male	35 to 44	Self employed	Huawei
5	Male	25 to 34	Teacher at high school	Techno
5	Female	25 to 34	Government employee	Techno
5	Female	18 to 24	Student at university	iPhone
5	Male	25 to 34	Teacher at university	Samsung
5	Male	25 to 34	Teacher at collage	Samsung
5	Female	18 to 24	Student at collage	Techno
5	Female	25 to 34	Accountant	Samsung
5	Male	35 to 44	Teacher at collage	Techno
5	Male	35 to 44	Banker	Techno

Result

The result of usability evaluation consists of the analysis of answer to the questionnaire. These are presented as follows.

Result of questionnaire

The result of the questionnaire towards the selected mobile applications according to **fifty** participants are shown below in table [5.4]. S1-S21, S22-S26 and S27-S42 are corresponding to usability attribute of effectiveness, efficiency and satisfaction respectively.

Table 5. 4 Result of the questionnaire collected from test users

Keys:

SD = Strongly dis agree (1) N = Neutral (3) SA = Strongly agree (5)

D = Dis agree (2) A = Agree (4) P = Participant

No	Statements	Telegram					Messenger					Xender					Amharic Dictionary					Bubble Shooter				
		SD	D	N	A	SA	SD	D	N	A	SA	SD	D	N	A	SA	SD	D	N	A	SA	SD	D	N	A	SA
S1	I found that it was easy to understand the apps and its content				30p (4)	20p (5)	2p (1)	3p (2)	5p (3)	25p (4)	15p (5)			10p (3)	20p (4)	20p (5)			10P (3)	15p (4)	25p (5)			10p (3)	24p (4)	16p (5)
S2	I found that it was easy to learn the app and its content				35p (4)	15p (5)	3p (1)	7p (2)	10p (3)	20p (4)	10p (5)			5p (3)	35p (4)	10p (5)			5p (4)	32p (4)	7p (5)			5p (3)	25p (4)	20p (5)
S3	The app was easy to navigate and easy to know help information	2p (1)	3p (2)	4p (3)	24p (4)	17p (5)	5p (1)	5p (2)	15p (3)	15p (4)	10p (5)		10p (2)	20p (3)	15p (4)	5p (5)		9p (2)	20p (3)	12p (4)	9p (5)	3p (1)	4p (2)	5p (3)	26p (4)	12p (5)
S4	The app provides easy way for interaction	4p (1)	3p (2)	4p (3)	23p (4)	16p (5)	4p (1)	3p (2)	2p (3)	34p (4)	7p (5)			10p (3)	25p (4)	15p (5)	7p (1)	7p (2)	17p (3)	5p (4)	14p (5)	5p (1)	4p (2)	5p (3)	24p (4)	12p (5)
S5	The app was easy to use and easy to customize	3p (1)	3p (2)	3p (3)	23p (4)	18p (5)	5p (1)	3p (2)	4p (3)	24p (4)	14p (5)	2p (1)	3p (2)	5p (3)	30p (4)	10p (5)	4p (1)	5p (2)	9p (3)	13p (4)	19p (5)	5p (1)	4p (2)	5p (3)	25p (4)	11p (5)
S6	The app took a lot of time to load	23p (1)	21p (2)	3p (3)	2p (4)	1p (5)	10p (1)	34p (2)	1p (3)	1p (4)	4p (5)	15p (1)	20p (2)	10p (3)	3p (4)	2p (5)	20p (1)	17p (2)	5p (3)	4p (4)	4p (5)	23p (1)	23p (1)	1p (3)	2p (4)	1p (5)

S7	The app is too slow and I had to wait for response to continue	22p (1)	22p (2)	1p (3)	2p (4)	3p (5)	20p (1)	22p (2)	2p (3)	3p (4)	3p (5)	10p (1)	30p (2)	7p (3)	2p (4)	1p (5)	17p (1)	22p (2)	4p (3)	4p (4)	3p (5)	21p (1)	23p (2)	1p (3)	2p (4)	3p (5)
S8	It was easy to complete the task without too much effort	4p (1)	3p (2)	4p (3)	25p (4)	14p (5)	4p (1)	3p (2)	4p (3)	23p (4)	16p (5)			5p (3)	30p (4)	15p (5)	1p (1)	2p (2)	5p (3)	25p (4)	17p (5)	5p (1)	4p (2)	4p (3)	25p (4)	12p (5)
S9	I found that it was easy to know what to do in this app	4p (1)	3p (2)	4p (3)	24p (4)	15p (5)	5p (1)	4p (2)	4p (3)	34p (4)	3p (5)	2p (1)	3p (2)	10p (4)	25p (4)	10p (5)	3p (1)	4p (2)	7p (3)	16p (4)	23p (5)	4p (1)	5p (2)	4p (3)	24p (4)	13p (5)
S10	The app took a lot of time to complete a given task	22p (1)	22p (2)	2p (3)	3p (4)	1p (5)	24p (1)	22p (2)	1p (3)	1p (4)	2p (5)	10p (1)	30p (2)	5p (3)	3p (4)	1p (5)	16p (1)	27p (2)	4p (3)	2p (4)	1p (5)	19p (1)	24p (2)	3p (3)	2p (4)	2p (5)
S11	The app provides task related clue	4p (1)	3p (2)	4p (3)	24p (4)	15p (5)	4p (1)	3p (2)	4p (3)	24p (4)	15p (5)	5p (1)	5p (2)	10p (3)	20p (4)	10p (5)	3p (1)	4p (2)	19p (3)	19p (4)	8p (5)	3p (1)	2p (2)	3p (3)	23p (4)	19p (5)
S12	The app provides error recovery	4p (1)	4p (2)	3p (3)	24p (4)	15p (5)	4p (1)	3p (2)	3p (3)	24p (4)	16p (5)	15p (1)	15p (2)	10p (3)	7p (4)	3p (5)	5p (1)	29p (2)	13p (3)	2p (4)	1p (5)	4p (1)	3p (2)	5p (3)	24p (4)	14p (5)
S13	The app provides useful help information	4p (1)	3p (2)	4p (3)	25p (4)	14p (5)	3p (1)	2p (2)	3p (3)	32p (4)	10p (5)			5p (3)	10p (4)	35p (5)	1p (1)	2p (2)	5p (3)	33p (4)	12p (5)	3p (1)	2p (2)	3p (3)	23p (4)	19p (5)
S14	It was difficult to find help information	15p (1)	23p (2)	4p (3)	5p (4)	3p (5)	4p (1)	33p (2)	5p (3)	3p (4)	4p (5)	3p (1)	17p (2)	15p (3)	8p (4)	7p (5)	2p (1)	4p (2)	23 (3)	21p (4)	2p (5)	20p (1)	22p (2)	3p (3)	3p (4)	2p (5)

S 15	The app has simple multimedia usage	4p (1)	2p (2)	4p (3)	22p (4)	18p (5)	5p (1)	3p (2)	4p (3)	24p (4)	14p (5)	1p (1)	2p (2)	5p (3)	32p (4)	10p (5)		3p (2)	7p (3)	31p (4)	9p (5)	5p (1)	4p (2)	4p (3)	25p (4)	12p (5)
S 16	The app lacks narrative for content structure	13p (1)	25p (2)	3p (3)	5p (4)	4p (5)	7p (1)	7p (2)	33p (3)	1p (4)	2p (5)	10p (1)	5p (2)	20p (3)	10p (4)	5p (5)	2p (1)	4p (2)	26p (3)	15p (4)	3p (5)	16p (1)	24p (2)	5p (3)	3p\ (4)	2p (5)
S 17	I found that it was easy to input the data in this app	4p (1)	3p (2)	5p (3)	24p (4)	14p (5)	4p (1)	3p (2)	4p (3)	25p (4)	14p (5)	25p (1)	10p (2)	10p (3)	3p (4)	2p (5)	28p (1)	17p (2)	4p (3)	1p (4)		5p (1)	4p (2)	5p (3)	24p (4)	12p (5)
S 18	I found that it was easy to use the output in this app	4p (1)	3p (2)	5p (3)	24p (4)	14p (5)	4p (1)	3p (2)	4p (3)	25p (4)	14p (5)				35p (4)	15p (5)				27p (4)	23p (5)	5p (1)	4p (2)	5p (3)	24p (4)	12p (5)
S 19	It was difficult to install this app	24p (1)	21p (2)	2p (3)	2p (4)	1p (5)	20p (1)	30p (2)				18p (1)	32p (2)				27p (1)	23p (2)				24p (1)	22p (2)	1p (3)	1p (4)	2p (5)
S 20	The app provides accurate information	5p (1)	4p (2)	5p (3)	24p (4)	12p (5)	4p (1)	3p (2)	4p (3)	24p (4)	14p (5)			5p (3)	35p (4)	10p (5)		5p (2)	15p (3)	18p (4)	12p (5)	3p (1)	2p (2)	3p (3)	23p (4)	19p (5)
S 21	The app gives error message that clearly tell me how to fix problem	4p (1)	3p (2)	4p (3)	24p (4)	15p (5)	3p (1)	2p (2)	4p (3)	24p (4)	17p (5)	2p (1)	3p (2)	10p (3)	34p (4)	1p (5)	8p (1)	27p (2)	13p (3)	2p (4)		3p (1)	2p (2)	3p (3)	22p (4)	20p (5)
S 22	The app provides a visual display to show the loading process	4p (1)	3p (2)	4p (3)	24p (4)	15p (5)	14p (1)	24p (2)	5p (3)	4p (4)	3p (5)	10p (1)	15p (2)	20p (3)	4p (4)	1p (5)	4p (1)	18p (2)	21p (3)	5p (4)	2p (5)	2p (1)	1p (2)	2p (3)	22p (4)	13p (5)

S 23	I found that the app was compatible for any device	5p (1)	4p (2)	5p (3)	24p (4)	12p (5)	3p (1)	2p (2)	3p (3)	24p (4)	18p (5)			30p (3)	15p (4)	5p (5)			38p (3)	8p (4)	4p (5)	4p (1)	3p (2)	4p (3)	24p (4)	15p (5)								
S 24	The app provides easy touch screen facilitation	5p (1)	5p (2)	4p (3)	24p (4)	12p (5)	4p (1)	3p (2)	4p (3)	23p (4)	16p (5)	2p (1)	3p (2)	30p (3)	13p (4)	2p (5)			7p (3)	31p (4)	1p (5)	4p (1)	3p (2)	5p (3)	25p (4)	13p (5)								
S 25	The app provides useful voice instruction	15p (1)	25p (2)	10p (3)			20p (1)	20p (2)	10p (3)			30p (1)	20p (2)				33p (1)	15p (2)	2p (3)			15p (1)	25p (2)	10p (3)										
S 26	The app notifies automatic update while updates available	5p (1)	4p (2)	4p (3)	25p (4)	13p (5)	4p (1)	3p (2)	4p (3)	24p (4)	15p (5)			8p (3)	32p (4)	10p (5)			6p (3)	37p (4)	7p (5)	4p (1)	3p (2)	5p (3)	25p\ (4)	14p (5)								
S 27	It was difficult to understand the interface in this app	22p (1)	22p (2)	3p (3)	1p (4)	2p (5)	24p (1)	22p (2)	1p (3)	1p (4)	2p (5)	15p (1)	25p (2)	10p (3)			13p (1)	27p (2)	7p (3)	3p (4)		23p (1)	21p (2)	1p (3)	2p (4)	3p (5)								
S 28	The app uses appropriate font style, size and color	3p (1)	4p (2)	5p (3)	23p (4)	15p (5)	4p (1)	3p (2)	4p (3)	24p (4)	15p (5)			5p (3)	25p (4)	20p (5)			8p (3)	32p (4)	10p (5)	5p (1)	4p (2)	5p (3)	23p (4)	13p (5)								
S 29	The app lacks message precision	24p (1)	21p (2)	2p (3)	1p (4)	2p (5)	10p (1)	32p (2)	3p (3)	3p (4)	2p (5)	2p (1)	38p (2)	10p (3)			3p (1)	26p (2)	17p (3)	4p (4)		22p (1)	23p (2)	1p (3)	2p (4)	2p (5)								
S 30	The app provides appropriate feedback for my action	5p (1)	4p (2)	4p (3)	25p (4)	13p (5)	4p (1)	3p (2)	4p (3)	34p (4)	5p (5)	4p (1)	4p (2)	12p (3)	25p (4)	5p (5)		3p (2)	14p (3)	28p (4)	5p (5)	3p (1)	4p (2)	4p (3)	23p (4)	16p (5)								

S 31	I found that the app was suitable to use	4p (1)	3p (2)	4p (3)	24p (4)	15p (5)	5p (1)	4p (2)	5p (3)	25p (4)	11p (5)			5p (3)	35p (4)	10p (5)			2p (3)	40p (4)	8p (5)	5p (1)	5p (2)	5p (3)	24p (4)	11p (5)
S 32	The app provides alert while error happens	5p (1)	5p (2)	4p (3)	24p (4)	12p (5)	4p (1)	3p (2)	4p (3)	24p (4)	15p (5)	2p (1)	3p (2)	5p (3)	35p (4)	5p (5)			3p (3)	26p (4)	21p (5)	4p (1)	5p (2)	5p (3)	23p (4)	13p (5)
S 33	I was satisfied by the app design and format	3p (1)	2p (2)	3p (3)	24p (4)	18p (5)	5p (1)	4p (2)	5p (3)	25p (4)	11p (5)			2p (3)	38p (4)	10p (5)			1p (3)	11p (4)	38p (5)	4p (1)	5p (2)	4p (3)	24p (4)	13p (5)
S 34	The logical design of menu and button is confusing	13p (1)	25p (2)	4p (3)	3p (4)	5p (5)	24p (1)	22p (2)	1p (3)	1p (4)	2p (5)	1p (1)	39p (2)	10p (3)			2p (1)	29p (2)	19p (3)			19p (1)	23p (2)	4p (3)	2p (4)	3p (5)
S 35	The app provides comfort while using	4p (1)	5p (2)	4p (3)	25p (4)	13p (5)	4p (1)	4p (2)	5p (3)	35p (4)	2p (5)				35p (4)	15p (5)				30p (4)	20p (5)	4p (1)	5p (2)	5p (3)	24p (4)	12p (5)
S 36	I found that the app has eye-catching user interface	4p (1)	4p (2)	3p (3)	23p (4)	16p (5)	4p (1)	5p (2)	4p (3)	24p (4)	13p (5)	4p (1)	4p (2)	2p (3)	25p (4)	15p (5)			15p (3)	20p (4)	15p (5)	3p (1)	4p (2)	3p (3)	24p (4)	16p (5)
S 37	I was comfortable with screen orientation of this application	3p (1)	2p (2)	4p (3)	23p (4)	19p (5)	5p (1)	4p (2)	5p (3)	25p (4)	11p (5)	4p (1)	3p (2)	3p (3)	35p (4)	5p (5)			2p (3)	27p (4)	21p (5)	4p (1)	3p (2)	5p (3)	24p (4)	14p (5)
S 38	The app provides clear and understandable navigation keys	3p (1)	2p (2)	1p (3)	24p (4)	20p (5)	4p (1)	3p (2)	4p (3)	25p (4)	14p (5)	4p (1)	3p (2)	3p (3)	34p (4)	6p (5)		2p (2)	4p (3)	32p (4)	12p (5)	5p (1)	5p (2)	4p (3)	25p (4)	11p (5)

S39	It was difficult to understand the language used in the app	17p (1)	23p (2)	3p (3)	3p (4)	4p (5)	24p (1)	22p (2)	1p (3)	2p (4)	1p (5)	30p (1)	20p (2)				36p (1)	14p (2)				25p (1)	21p (2)	2p (3)	1p (4)	1p (5)							
S40	The app provides different difficult level that I could easily change according to my choice	4p (1)	5p (2)	4p (3)	25p (4)	13p (5)	3p (1)	2p (2)	3p (3)	24p (4)	18p (5)	1p (1)	2p (2)	2p (3)	35p (4)	5p (5)		2p (2)	4p (3)	34p (4)	10p (5)	3p (1)	3p (2)	3p (3)	23p (4)	19p (5)							
S41	The app provides progress report /result for my performance in every activity	5p (1)	5p (2)	4p (3)	24p (4)	12p (5)	5p (1)	4p (2)	5p (3)	24p (4)	11p (5)	4p (1)	3p (2)	3p (3)	32p (4)	8p (5)			8p (3)	35p (4)	8p (5)	4p (1)	4p (2)	5p (3)	24p (4)	13p (5)							
S42	Overall, I was satisfied by this app	3p (1)	4p (2)	3p (3)	24p (4)	16p (5)	4p (1)	5p (2)	4p (3)	24p (4)	13p (5)				27p (4)	23p (5)				15p (4)	35p (5)	4p (1)	4p (2)	3p (3)	24p (4)	15p (5)							

Result Analysis

For comparative analysis of the five mobile applications result of usability evaluation from Table [5.4] is considered. The comparative analysis is carried out to determine whether one application has better usability than the other. Moreover, this was useful to determine whether the suggested usability testing process iteration framework was effective for usability evaluation or not. The data from the questionnaire was collected through five-point system usability scale form participants after performing the given tasks. Average scale of the questionnaire from different usability attributes is shown in Table [5.5].

Table 5. 5 Average scale of the questionnaire

Usability attribute	Telegram	Messenger	Xender	Amharic Dictionary	Bubble Shooter
Effectiveness	**33.58**	**31.02**	**33.25**	**31.92**	**32.83**
Efficiency	**33.8**	**31.48**	**29.24**	**28.76**	**33.44**
Satisfaction	**33.97**	**32.46**	**34.28**	**36.73**	**32.93**

The result shows that telegram is better usability than other. It means the participants were more satisfied with telegram and has good experience using it. However, both apps showed poor usability regarding help, navigation, voice instruction and error message. Furthermore, the participants were unsatisfied with interface color, text size, readability and virtual keyboard of messenger. These user interface design need to improved.

The result indicates that the questionnaire developed in the framework is reliable and effective to collect qualitative data for testing the usability of mobile apps. Generally, the overall analysis shows that both measurement instrument or evaluation mechanisms developed from the metrics are appropriate (effective) for testing the usability of mobile apps. The result also shows that the framework is useful for evaluating usability and comparison of different mobile applications. Therefore, standing from the above result analysis the suggested usability testing process iteration framework is useful for conducting usability testing for different categories of mobile applications as well as for developing usable mobile applications.

In addition to this the suggested framework has better performance. Because firstly, it helps to discover usability issues that may led to users for errors and highlights the user interface design areas for suggested improvement. Secondly, it indicates new features which is included in mobile applications. Thirdly, it enables to have standardized usability testing language, consistency during usability testing and finally to have structured and general methodology to develop usable mobile applications. Therefore, the suggested framework in this work is effective and reliable.

Chapter Six Conclusion, Recommendation and Future Studies

The objective of this work is to provide an innovative way of evaluating the usability of mobile app to improve their usability for end users. Therefore, in this paper we suggest usability testing process iteration framework for mobile apps. A review of existing user interface guidelines is carried out to develop usability guidelines for interface design of mobile apps for mobile app developers. The framework provides a comprehensive structure for testing the usability of mobile apps.

Usability testing metrics are developed to evaluate each goal (UI design criteria). Finally, two evaluation techniques task list and questionnaire are developed from the metrics in order to collect qualitative and quantitative data from test users to complete usability evaluation. The paper provides a starting point for carrying out usability testing and will be helpful for evaluators and developers by serving as a guideline for testing the usability of mobile apps.

The validation of the framework is done by implementing in a usability study. Usability testing was carried out with five selected mobile apps. The main purpose of usability study was to determine whether the framework is effective to collect qualitative and quantitative data for usability, analyze and compare the apps, provide results to uncover the usability issue and limitation with regard to the UI design and highlighting the areas of improvement. The results of this work explain that the framework is useful for testing the usability of mobile apps and also enables to have suggested usability testing language, consistency in usability testing of mobile applications and enables the tester to consider technical knowledge of mobile users and mobile application variance during testing.

We recommend that further studies should be carried out because, the rapid change in mobile technology and a large number of mobile apps being developed may cause the interface design criteria and metrics presented in this paper to be updated in future in order to much the need of changing technology. The framework can also be modified based on the new design guideline. Therefore, goals, guideline, quality characteristics and metrics can be added or deleted. A new measure can be including in the framework. Thus, the developed tasks and questionnaire can also be updated accordingly.

Reference

[1] http://en.wikipedia.org/wiki/Mobile_apps, Retrieved 29 January 2012

[2]“Usability 101: Introduction to Usability.” [Online]. Available: http://www.nngroup.com/articles/usability-101-introduction-to-usability/.[Accessed:5-Feb-2019]

[3] J. Nielsen, & J. Levy, “Measuring usability: Preference vs. Performance”, Communications of the ACM, vol. 37(4), pp.66-75, 2003

[4] Hevner, A., & Chatterjee, S. (2010). Design Science Research in Information Systems. In: Design Research in Information Systems, pp. 9–22, Springer. [Accessed: 28-Dec-2019]

[5] Vaishnavi, V., & Kuechler, W. (2004). Design research in information systems. Available at: http://www.citeulike.org/group/4795/article/6505471

[6] Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28(1), pp. 75–105

[7] Maria Shitkova, 12th International Conference on Wirtschaftsinformatik, March 4-6 2015, Osnabrück, Germany, Towards usability guidelines for mobile app & website

[8] Alshehri, F. & Freeman, M. (2012). Methods for usability evaluations of mobile devices. In J. W. Lamp (Eds.), 23rd Australian

Conference on Information Systems (pp. 1-10). Geelong: Deakin University.

[9] Peffers et al. (2008) A Design Science Research Methodology for Information Systems Research

[10] Tapanee Treeratanapon, Design of the usability measurement framework for mobile applications, International Conference on Computer and Information Technology (ICCIT'2012) June 16-17, 2012, Bangkok

[11] "Mobile Market Assocation," sep 2008, p. 12.

[12] J. M. &. S. Gowell, Proffesiona Mobile Application Development, Indiana: John Wiley & Sons, Inc. , 2012.

[13] A. K. &. R. Mente, "M-Commerce: Services and applications," *International Journal of Advanced Science and Research,* vol. 3, no. 1, pp. 19-21, March 2018.

[14] N. Ratnakar, 15 February 2015. [Online]. Available: http://www .softwaretestinghelp.com/beginners-guide-to-mobile-application-testing/. [Accessed 6 april 2019].

[15] X. Zhiming, "Mobile application development," p. 30, 2 september 2016.

[16] Z. M. Gillenwater, 15 December 2010. [Online]. Available: http://zomigi.com/blog/examplesof-flexible-layouts-with-css3-media-queries. [Accessed 6 April 2019].

[17] F. lab, "Choosing between responsive web design and a separate mobile site to improve mobile visitor's experience.," 17 April 2015. [Online]. Available: http://www.foraker.com/choosing-between-responsive-web-designand-a-separate-mobile-site-to-improve-mobile-visitors-experience/. [Accessed 6 April 2019].

[18] W3C, "Standards for Web Applications on Mobile: current state and roadmap," 1 May 2015. [Online]. Available: http://www.w3.org/Mobile/mobileweb-app-state/. [Accessed 6 April 2019].

[19] J. Cowart, "Developer Economics: Pros and Cons of the Top 5 CrossPlatform Tools," 12 November 2013. [Online]. Available: http://www. developereconomics.com/pros-cons-top-5-cross-platform-tools/. [Accessed 6 April 2019].

[10] A. Cordova, 5 April 2015. [Online]. Available: https://cordova.apache.org/. [Accessed 6 April 2019].

[21] M. Project, 20 Juanuary 2015. [Online]. Available: http://www.mono-project.com/. [Accessed 6 April 2019].

[22] C. Reynolds, "Ten of the Best Cross-Platform Mobile Development Tools for Enterprises," 16 Augest 2015. [Online]. Available: http://appindex.com/blog/ten-best-cross-platform-development-mobileenterprises/. [Accessed 6 April 2019].

[23] I. Dalmasso, D. N., B. Christian and N. N., "Survey, Comparison and Evaluation of Cross Platform Mobile Application Development Tools," 10 April 2015. [Online]. Available: http://www.academia.edu /6486265/Survey_Comparison_and_Evaluation_of_Cross_Platform_Mobile_ Application_Development_Tools. [Accessed 6 April 2019].

[24] R. Budiu, 19 January 2016. [Online]. Available: https://www.nngroup.com/articles/mobile-native-apps/. [Accessed 29 March 2019].

[25] "Smartphone OS Market Share," 29 December 2015. [Online]. Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp. [Accessed 6 April 2019].

[26] "Open Handset Alliance," 20 March 2015. [Online]. Available: http://www. openhandsetalliance.com. [Accessed 6 April 2019].

[27] D. C. Rajapakse, "Fragmentation of mobile applications,," 28 April 2008. [Online]. Available: http://www.comp.nus.edu.sg/~damithch/df/devicefragmentation.htm. [Accessed 6 April 2019].

[28] FreeBSD.org, "FreeBSD Handbook – Who uses BSD?," 4 May 2015. [Online]. Available: https://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook /nutshell.html. [Accessed 6 April 2019].

[29] Apple, "App Store Distribution," 16 February 2015. [Online]. Available: https://developer.apple.com/support/appstore/. [Accessed 6 April 2019].

[30] T. Bradley, "Apple iOS Is More Fragmented Than It Seems," 20 Juanaury 2015. [Online]. Available: http://www.forbes.com/sites/tonybradley /2013/06/21/apple-ios-is-more-fragmented-than-it-seems/. [Accessed 6 April 2019].

[31] W. c. Microsoft, 23 December 2014. [Online]. Available: http://www.windowscentral.com/microsoft-71-percent-windows-phone-appdownloads-go-low-memory-devices. [Accessed 7 April 2019].

[32] J. Jeesmon, "Beginner's Guide for Mobile Applications Testing," 2015.

[33] Keynote, "Testing Strategies and Tactics for Mobile Applications," 20 April 2015. [Online]. Available: http://www.keynote.com/resources/white-papers/ testing-strategies-tactics-for-mobile-applications. [Accessed 7 April 2019].

[34] M. &. C. M. Kumar, "Best Practices in Mobile Application Testing," 14 March 2015. [Online]. Available: http://www.infosys.com /flypp/resources/Documents/mobile-application-testing.pdf. [Accessed 7 April 2019].

[35] R. D. &. J. S. P. Craig, "Systematic Software Testing," 2002.

[36] ISTQB, "Foundation Level Syllabus," 29 April 2015. [Online]. Available: http://www.istqb.org/downloads/finish/16/15.html. [Accessed 7 April 2019].

[37] D. Carver, "Unit Test Android Without Going Bald," 2015. [Online]. Available: http://www.slideshare.net/davidcarver7798/unit-test-androidwithout-going-bald. [Accessed 7 April 2019].

[38] ISTQB, "Standard Glossary of Terms Used in Software Testing Version 3.0," 26 March 2015. [Online]. Available: http://www.istqb.org/downloads /finish/20/194.html. [Accessed 7 April 2019].

[39] A. &. H. D. Kolawa, "Automated Defect Prevention," *Best Practices in Software Management,* 2007.

[40] M. Cohn, Succeeding with Agile, 2009.

[41] M. Fowler, "Continuous Integration," 6 May 2006. [Online]. Available: http://www.martinfowler.com/articles/continuousIntegration.html. [Accessed 7 April 2019].

[42] M. Budhabhatti, "MSDN: Test-Driven Development and Continuous Integration for Mobile Applications," 2008. [Online]. Available: https://msdn.microsoft.com/en-us/library/bb985498.aspx..

[43] Xamarin, "Introduction to Continuous Integration," 2015. [Online]. Available: http://developer.xamarin.com/guides/cross-platform/ci/intro_to_ci/.

[44] I. 25010, "Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) Quality model," 2008.

[45] Compuware, "What Users Want from Mobile," 2011. [Online]. Available: http://e-commercefacts.com/research/2011/07/what-usrs-wantfrom-mobil/19986_WhatMobileUsersWant_Wp.pdf.. [Accessed 7 April 2019].

[46] J. Arbon, "App Quality: Secrets for Agile App Teams, Kindle edition," 2015.

[47] Dynatrace, "Mobile Apps: What Consumers Really Need and Want," 2012.

[48] Smashingmagazine, 2 November 2014. [Online]. Available: http://www.smashingmagazine.com/2014/10/02/what-every-appdeveloper-should-know-about-android/. [Accessed 7 April 2019].

[49] H. Packard, "HP Research Reveals Nine out of 10 Mobile Applications Vulnerable to Attack," 13 December 2013. [Online]. Available: http://www8.hp.com/us/en/hp-news/press-release.html?id=1528865#. [Accessed 7 April 2019].

[50] O. M. S. Project:, "Top 10 Mobile Risks,," 2014. [Online]. Available: https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=Top_10_Mobile_Risks.. [Accessed 7 April 2019].

[51] G. Kalra, "Mobile Application Security testing," [Online]. Available: http://www.mcafee.com/us/resources/white-papers/foundstone/wp-mobileapp-security-testing.pdf. [Accessed 7 April 2019].

[52] FireEye, "SSL Vulnerabilities: Who listens when Android applications talk?," 2014. [Online]. Available: https://www.fireeye.com/blog/threatresearch/2014/08/ssl-vulnerabilities-who-listens-when-android-applicationstalk.html. [Accessed 8 April 2019].

[53] OWASP, "Lack of Binary Protections," 2015. [Online]. Available: https://www.owasp.org/index.php/Mobile_Top_10_2014-M10. [Accessed 8 April 2019].

[54] N. Pavlov, "Non-Functional Testing on Mobile devices," 2015. [Online]. Available: http://www.nordictestingdays.eu/sites/default/files/ NTD2012%20Presentations/Non-

functional%20testing%20on%20mobile %20devices%20-%20Nikolai%20Pavlov.pdf. [Accessed 8 April 2019].

[55] E. a. S. A.Fernandez, "Usability evaluation method for a web," in *A systematic mapping study* , Aug 2011 , pp. 789-811.

[56] A. K. W. S. a. A. S. A. Abran, "Usability meanings and interpretation in ISO standareds," *Software quality journal* , vol. 11, pp. 325-338, 2003.

[57] L. L. a. J. Barnes, Usability engineering process, products and examples, Pearson Prentic Hall, 2008.

[58] A. M. a. S. K. Dubey, "Usability Evalution Methods," vol. 4, pp. 590-599, 2012.

[59] C. C. Whitehead, "Evaluationg web pages and web sites usability," in *Proceeding of the 44th annual southeast regional conference*, 2006.

[60] R. E. Al-qutaish, "Quality model in software engineering litrature," pp. 166-175, 2010.

[61] A. K. W. S. a. A. S. A. Abran, "Usability meanings and interpretation in ISO standard," *Software quality journal*, vol. 11, pp. 325-338, 2003.

[62] P.-Y. Yen, "Health information technology usability evaluation method, models and measures," 2010.

[63] A. H. a. E. Ferneley, "Usability metrics for mobile application A GQM," pp. 567-570, 2008.

[64] A. Khalili, "User interface for semantic count authoring," *A Systematic litrature review*, May 2012.

[65] M. M. K, TRIGENT VANTAGE, 20 February 2017. [Online]. Available: https://blog.trigent.com/different-types-of-mobile-applications-native-hybrid-and-web-apps. [Accessed 29 March 2019].

[66] Aruna Alluri, Usability testing of Android Applications, 2012

[67] Bikash Singhal, Usability Testing of iPhone, 2011

[68] Junnu Pitkanen, Mobile Application Usability Research, Feb 6, 2016

[69] Babita Shivade & Meena Sharma, Usability Evaluation Too for Android based Mobile Application, International Journal of Emerging Trends & Technology in Computer Science, Vol 3, Issue 3, May-Jun 2014

[70] Bernhaupt, R., Mihalic, K. & Obrist, M. (2008): Usability Evaluation Methods for Mobile Applications. In: Lumsden, J. (Ed.). Handbook of Research on User Interface Design and Evaluation for Mobile Technology, IGI Global. 745-758.

Appendix 1: Consent Form

Consent form

Please read and sign this form. You have been invited to participate in a usability testing which is part of my master thesis studies at Addis Ababa University in 4-kilo campus. By participating in the evaluation, you will help to evaluate and improve the usability of mobile applications. The evaluation process includes three parts:

1) In the tasks, you will be asked to perform a set of tasks in different scenarios;

2) In the questionnaire, you will give scales to a series of statements;

3) In the interview, you will answer the questions given by the tester.

Participation in this interview is voluntary. All information will remain strictly confidential. The findings might be used to analyze the suggested design in this work. However, at no time will your name or any other identification be used. You can withdraw your consent to the interview and stop participation at any time. If you have any questions, please contact kirmtu2009@gmail.com.

I have read and understood the information on this form.

Participants Name: _____

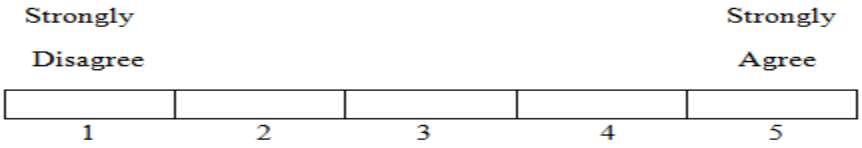
Data and Place: _____

Signature: _____

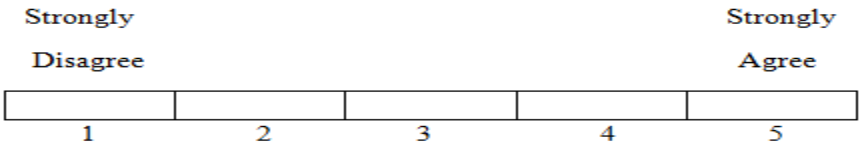
Appendix 2: Questionnaire

Effectiveness

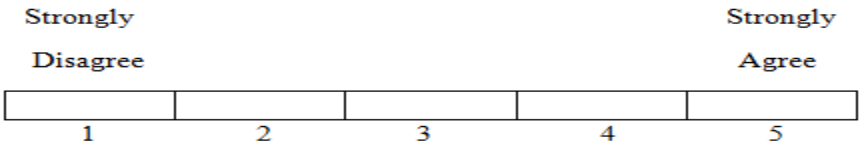
S1. I found that it was easy to understand the apps and its content



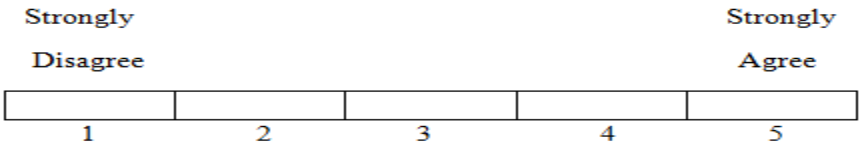
S2. I found that it was easy to learn the app and its content



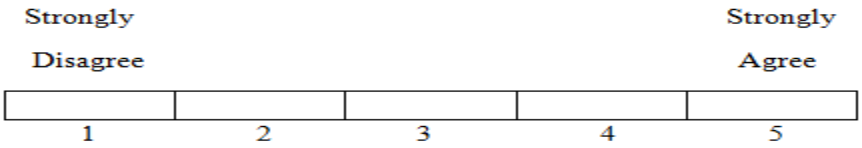
S3. The app was easy to navigate and easy to know help information



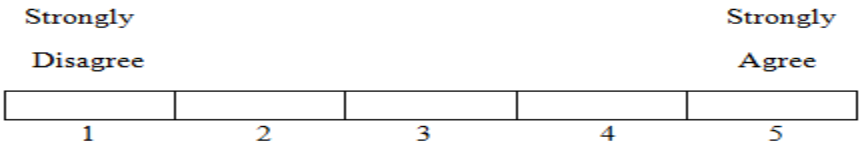
S4. The app provides easy way for interaction



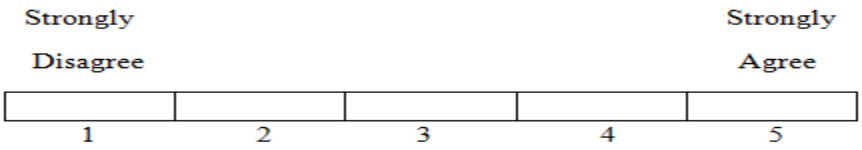
S5. The app was easy to use and easy to customize



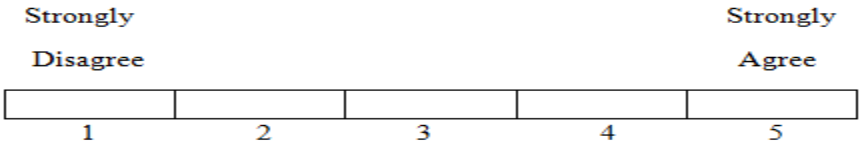
S6. The app took a lot of time to load



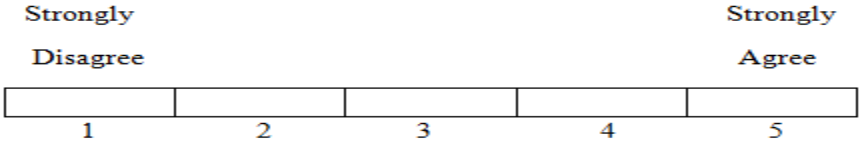
S7. The app is too slow and I had to wait for response to continue



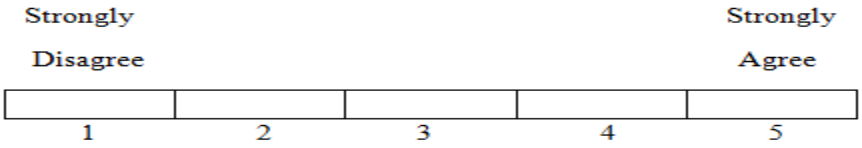
S8. It was easy to complete the task without too much effort



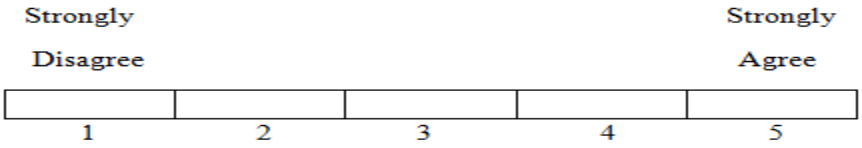
S9. I found that it was easy to know what to do in this app



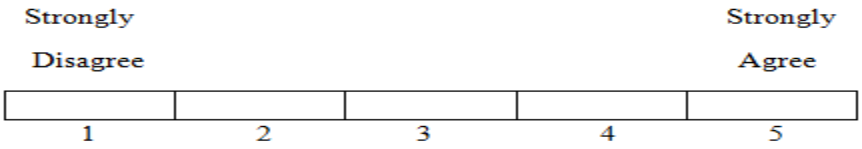
S10. The app took a lot of time to complete a given task



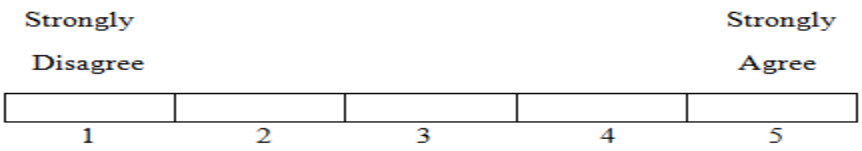
S11. The app provides task related clue



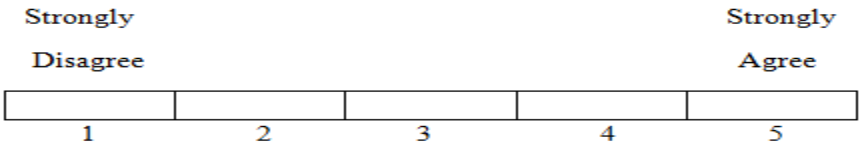
S12. The app provide error recovery



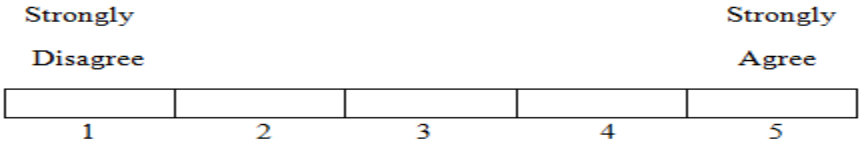
S13. The app provides useful help information



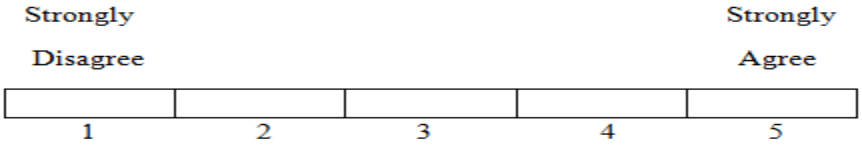
S14. It was difficult to find help information



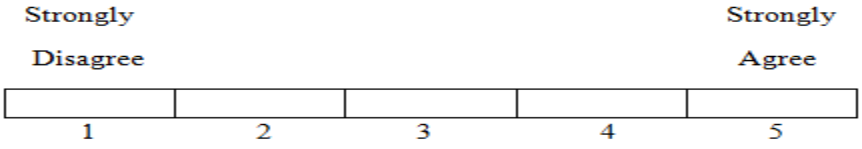
S15. The app has simple multimedia usage



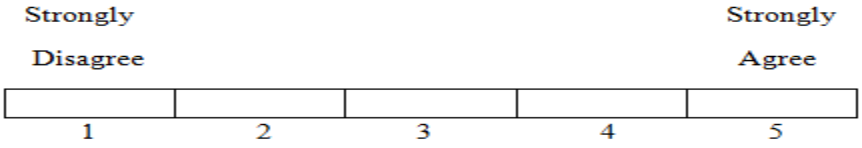
S16. The app lacks narrative for content structure



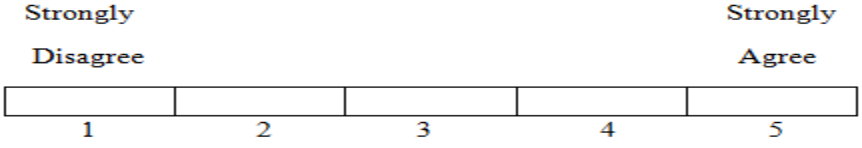
S17. I found that it was easy to input the data in this app



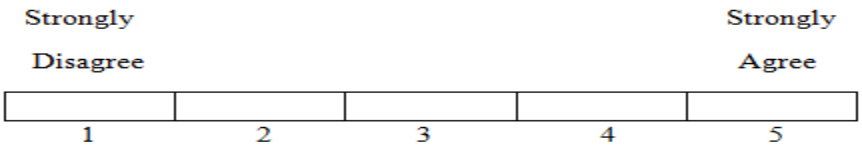
S18. I found that it was easy to use the output in this app



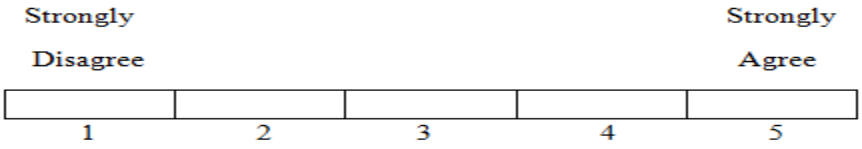
S19. It was difficult to install this app



S20. The app provides accurate information

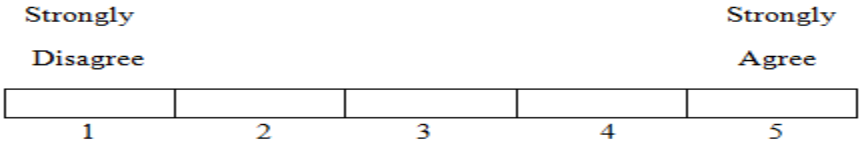


S21. The app gives error message that clearly tell me how to fix problem

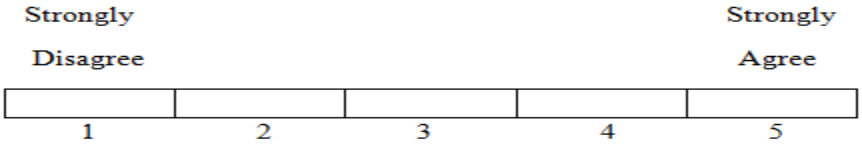


Efficiency

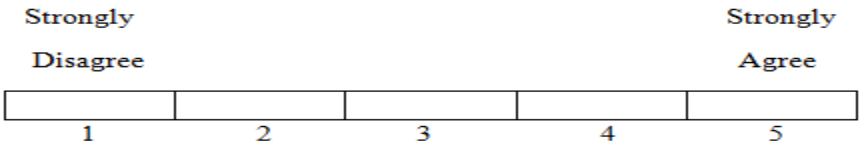
S22. The app provides a visual display to show the loading process



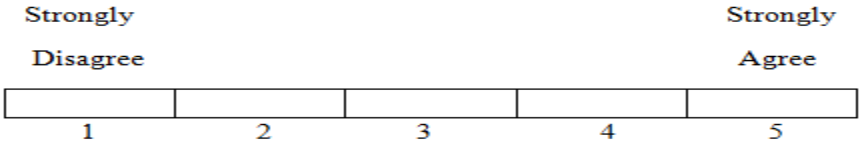
S23. I found that the app was compatible for any device



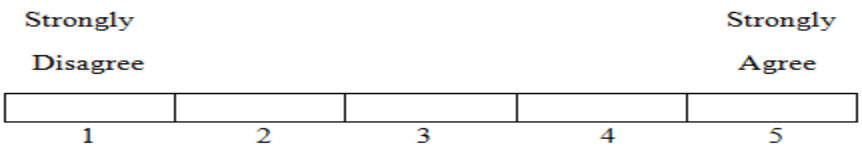
S24. The app provides easy touch screen facilitation



S25. The app provides useful voice instruction

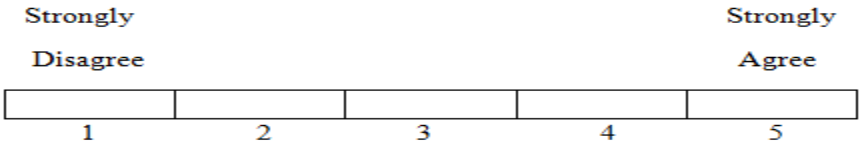


S26. The app notifies automatic update while updates available

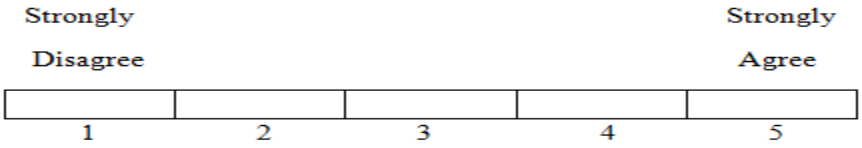


Satisfaction

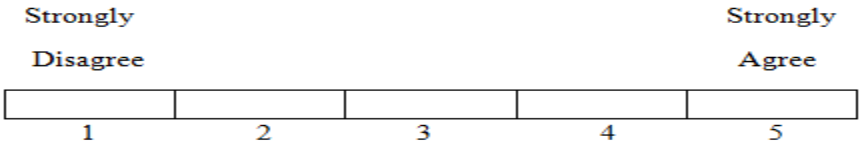
S27. It was difficult to understand the interface in this app



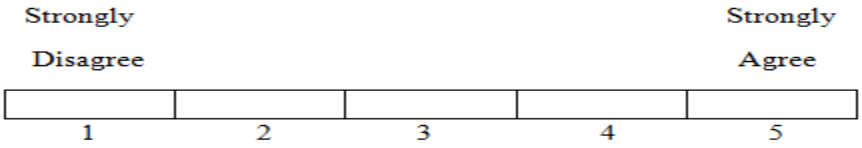
S28. The app use appropriate font style, size and color



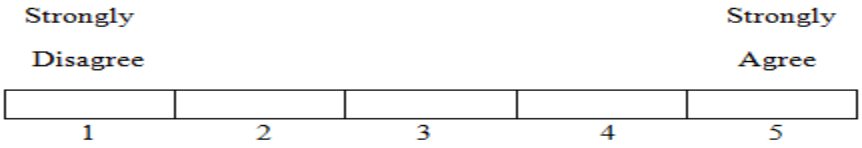
S29. The app lacks message precision



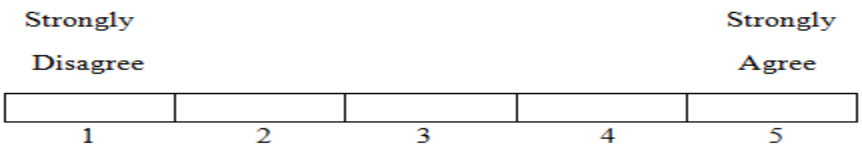
S30. The app provide appropriate feedback for my action



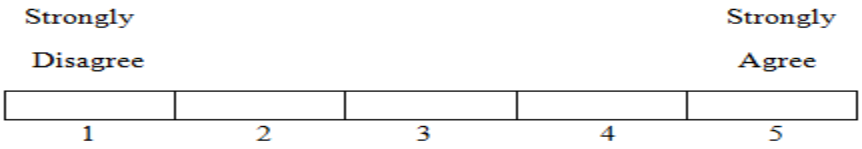
S31. I found that the app was suitable to use



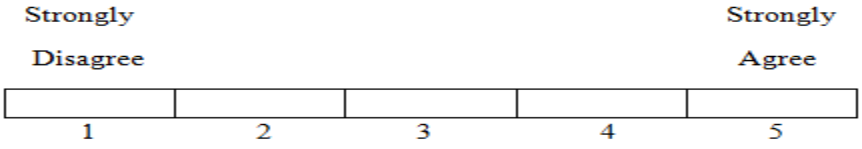
S32. The app provides alert while error happens



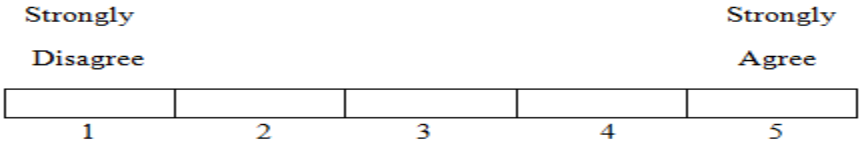
S33. I was satisfied by the app design and format



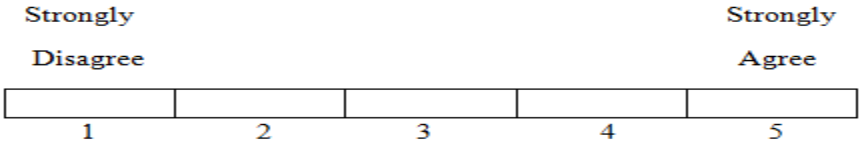
S34. The logical design of menu and button is confusing



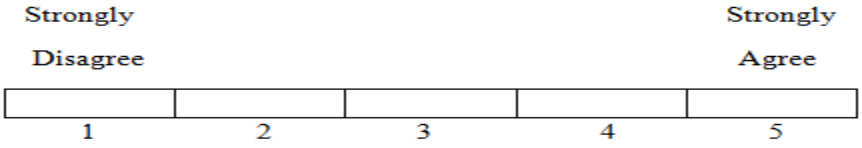
S35. The app provides comfort while using



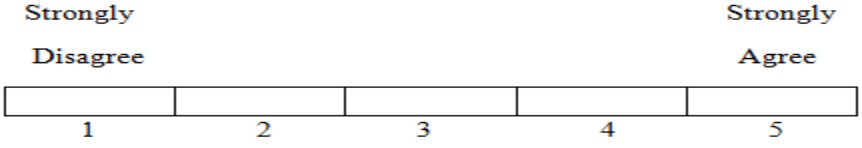
S36. I found that the app has eye-catching user interface



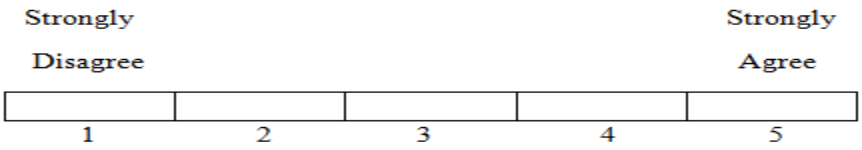
S37. I was comfortable with screen orientation of this application



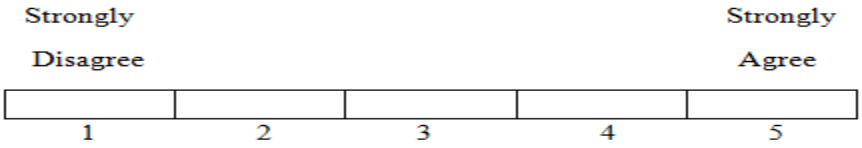
S38. The app provides clear and understandable navigation keys



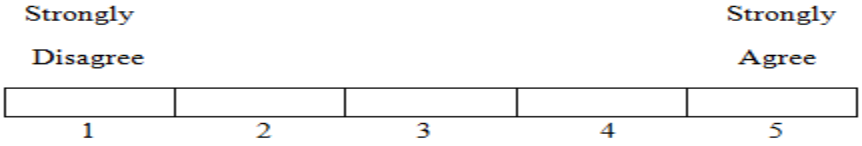
S39. It was difficult to understand the language used in the app



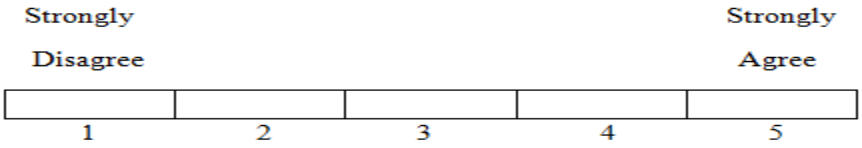
S40. The app provides different difficult level that I could easily change according to my choice



S41. The app provides progress report /result for my performance in every activity



S42. Overall, I was satisfied by this app



DECLARATION

I, the undersigned, declare that this research is my original work and has not been presented for degree in any other university, and that all sources of materials used for the research have been acknowledged.

Declared by:

Name: Kiross Shiferaw

Signature: _____

Date: _____

Confirmed by advisor:

Name: Ayalew Belay (PhD)

Signature: _____

Date: _____

Place and date of submission: Addis Ababa University, April 12, 2021.