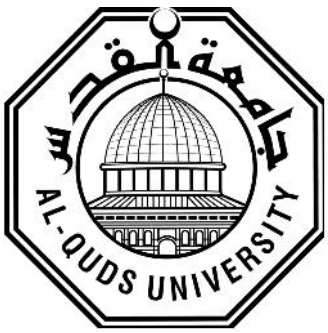
Deanship of Graduate Studies
Al-Quds University



Towards a Modified Scrum for Multimedia Mobile Software Development

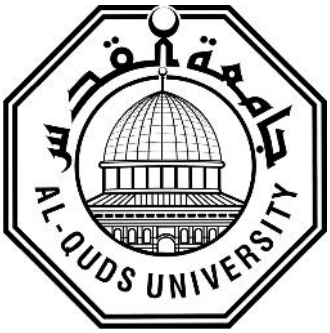
Ramzi Nasri Mikhail Sansour

M.Sc. Thesis

Jerusalem – Palestine

1437/2016

Towards a Modified Scrum for Multimedia Mobile Software Development



Prepared By:

Ramzi Nasri Mikhail Sansour

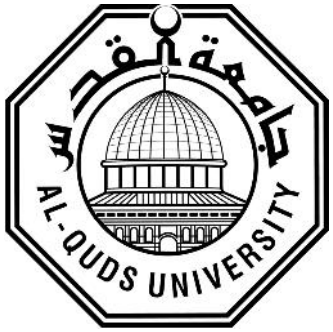
B.Sc.: Al-Quds Open University
Palestine

Internal Supervisor: Dr. Nidal Kafri
External Supervisor: Dr. Muath Sabha

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Computer Scienceat Al-Quds University

1437/2016

Al-Quds University
Deanship of Graduate Studies
Master in Computer Science
Computer Science & Information Technology



Thesis Approval

Towards a Modified Scrum for Multimedia Mobile Software Development

Prepared By: Ramzi Nasri Sansour
Registration No: 20911727

Internal Supervisor: Dr. Nidal Kafri
External Supervisor: Dr. Muath Sabha

Master thesis submitted and accepted, Date:
The names and signatures of the examining committee members are as follows:

1- Head of Committee: Dr. Nidal Kafri	Signature ……...
2- Internal Examiner: Dr. Jad Najjar	Signature ……...
3- External Examiner: Dr. Nael Salman	Signature ……...
4- Committee Member: Dr. Muath Sabha	Signature ……...

Jerusalem – Palestine

2015/2016

Dedication

Many thanks to all my friends who gave me support. And a special thank for my wife who gave me great support during the study period and for her understanding.

Declaration

I certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed: ……

Ramzi Nasri Mikhail Sansour

Date: 09 December 2015

Acknowledgments

First of all, I want to thank my internal and external supervisors, Dr. Nidal Kafri and Dr. Muath Sabha for giving me support, assistance, encouragement, and providing data during the master thesis preparation and work of the thesis and for developing the project included. I also want to thank all the faculty members in Computer Deanship of Graduate Studies at Al-Quds University. I am also grateful to the team who helped me in the application development, for helping me understand the chemical experiment for doing the application and the developer for helping me, where we used the Scrum methodology to develop the application.

Abstract

In spite of the development of mobile devices, there are some drawbacks such as limited processor speed, short battery life and small storage capacity. Also most of these applications are small and light and not large systems, compared with mainframe systems. Thus, we need to develop such applications in the shortest possible time, and this led to the adoption of modified software and engineering tools to achieve a reduced life cycle development process. This thesis aims to find more efficient approach and methodology for developing small multimedia mobile applications and for applications with similar products style, trying to follow the best software engineering methodology for this category of applications. A relatively small multimedia mobile application was developed for multi-platform smartphones, using most of Smartphone functionalities, applying the Scrum methodology, where it runs on Android, iOS and BlackBerry. Although we agree that Scrum is the best software paradigm up to now to be followed for developing mobile applications, we suggest some minor modifications on traditional Scrum to speed up the development process, and improve Scrum approach in this kind of applications.

ملخص الدراسة

إن معظم تطبيقات الهواتف الذكية صغيرة وخفيفة وسريعة التطور بالمقارنة مع أنظمة الحواسيب الكبيرة. وبالتالي فإننا بحاجة لتطوير مثل هذه التطبيقات بأقصر وقت ممكن، وهذا أدى إلى اعتماد هندسة برمجيات معدلة وأدوات لتحقيق تقليل دورة حياة عملية التطوير.

ومع وجود العديد من أنظمة التشغيل، بات من الضروري إيجاد الطرق لتحقيق مصطلح "الكتابة مرة واحدة، النشر في كل مكان write-once-deploy-everywhere"، عن طريق استخدام أدوات لعمل مثل هذه البرامج، حيث يتم كتابة الرمز مرة واحدة على نظام معين وتطبيقه على أنظمة أخرى بدون/أو بإضافة قطع صغيرة من التعليمات البرمجية.

وحيث ان المطورين استخدموا الطرق التقليدية في تطوير البرامج، فلقد حاولوا إيجاد أفضل الطرق للتعامل مع الأجهزة النقالة، فلقد كانت طريقة "الأجايل Agile" باستخدام منهجية "السكروم Scrum" من أفضل طريقة لتطوير برامج الأجهزة النقالة.

من هذا المنطلق، لقد اهتمينا في هذه الرسالة بإيجاد الحل الأمثل لتطوير برامج صغيرة وبرامج مماثلة من حيث التطوير للأجهزة النقالة، للسعي وراء أفضل الطرق لتطوير مثل هذه البرامج، حيث قمنا بتطوير تطبيق متعدد الوسائط يستخدم أكثر وظائف الهاتف الذكي، حيث أنه يعمل على نظام أندرويد Android ونظام أي أو أس iOS ونظام بلاك بيري BlackBerry، وقد قمنا باستخدام منهجية سكروم Scrum في تطوير هذا التطبيق. ومع ذلك، وجدنا ان سكروم Scrum ليس هو الحل الأمثل لمثل هذا النوع من التطبيقات، حيث يستغرق وقتاً طويلاً في البرمجة. وفي النتيجة، لقد اقترحنا بعض التعديلات الطفيفة لتسريع عملية التطوير، وتحسين منهجية السكروم Scrum في مثل هذا النوع من التطبيقات.

Table of Contents

List of Tables

Table No. **Table Title** **Page**

List of Figures

Figure No. **Figure Title** **Page**

List of Abbreviations

Abbreviation	Full Name
AOADL	Aspect-Oriented Architecture Description Language
AOSE	Agent-Oriented Software Engineering
API	Application Programming Interface
ASD	Adaptive Software Development
AUP	Agile Uniform Process
CPI	Content Provider Interface
CPU	Central Processing Unit
CSS	Cascading Style Sheet
DSDM	Dynamic Systems Development Method
FDD	Feature Driven Development
GB	Giga Byte
GHz	Giga Hertz
GMA	Generic Mobile Application
GPS	Global Positioning System
GQM	Goal-Question-Metric
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
IDE	Integrated Development Environment
JSFL	JavaScript Flash Language
MASAM	Mobile Application Software development based on Agile
MB	Mega Byte
MEAP	Mobile Enterprise App Platform
MVC	Model View Controller
OEM	Original Equipment Manufacturer
OS	Operating System
PC	Personal Computer
PCM	Project Capability Model
RAM	Random Accesses Memory
RaPiD7	Rapid Production of Documentation - 7 steps
RIM	Research In Motion
SDK	Software Development Kit
SLeSS	Scrum Lean Six Sigma
SPLE	Software Product Line Engineering
SQL	Structured Query Language
UI	User Interface
URL	Uniform Resource Locator
VB	Visual Basic
VoIP	Voice Over Internet Protocol
WYSIWYG	What You See Is What You Get
XNA	Xbox New Architecture
XP Programming	eXtreme Programming

Chapter One

Introduction

Introduction

With the development of technology nowadays, the world became very dependent on information technology and its computing devices in all aspects of life. Since people are always trying to evolve and find the best way for achieving work, they are always seeking for all possibilities available to help them to accomplish their jobs as fast as possible. With greater attention to work anywhere and anytime, handheld mobile devices such as tablet PCs and Smartphones became available for all people including businessmen, students, teaching staff, managers, lawyers, doctors, and others. It became difficult to dispense these devices as they achieve ubiquity because of their portability and high performance.

Over time, the basic requirements for the mobile phone have increased, as users demand more features and functionalities, leading to emergent usage of Smartphones. *Smartphones* even have many functionalities over other computer systems, like portability, mobility, availability, flexibility, light weight and small size (Wasserman, 2010).

Many Smartphones have processors with clock speeds more than 1.5 GHz and 2 GB of RAM and increased storage capacity. Smartphones allow users to perform high performance applications ranging from multimedia streaming and multimedia applications to web browsing. It

includes network connectivity functionality as Wi-Fi, 3G or 4G, GPS, and other functionalities like multi-touch screens, gyroscope sensors, high-resolution cameras, high quality sound systems and other features which are missing in older handheld devices (mobile phones).

Users can take high-resolution photos with Smartphones, edit them, create Office documents, and upload them to the Cloud. They can find their way around the world with the built-in GPS. They can access internet easily, read their emails, play network games, chat by using social network applications, and keep up-to-date with the latest news from around the world.

As mobile devices became capable to cope with the new technology, with its great capabilities, developers began to produce applications to let users deal with them effectively, not only for talking, but also for different categories like Games, Education …etc.

Although, these devices became more developed than before, there have some drawbacks/limitations, such as relatively small screen size, short battery life, computation and storage capacity. Also almost all applications for these devices are small and light client side oriented based on different mobile platforms such as Android, iOS, Windows Phone, BlackBerry and others, compared with the regular large system applications. Consequently, we need to develop such applications in shorter elapsed time (life cycle). Therefore, this led to develop and adopt a new modified software engineering approach and tools to achieve the reduction in the life cycle of the development process, to clarify that not only light applications but also similar products, for example applications for e-learning for some topics like chemistry, physics, math …etc. These topics have similarity in style experience of the development of client demands, to

take in account the accumulative experience. When developing such applications, for the first time, the developers earn experience, where it can contribute to the developing of subsequent applications. In the first development of such applications, there could be convergent meetings. Later the meeting will be relatively far apart, because of the previous experience, and there is no reason for meetings repetition, which gradually shrink. This is in addition to some reasons, including that it is not necessarily for the whole team to meet, where it can result in a waste of time to some, such as the topic expert, that has nothing to do with the work of the developers.

Problem Definition

In spite of the development of mobile devices, there are some drawbacks such as limited processor speed, short battery life and small storage capacity. Also most of these applications are small and light and not large systems, compared with mainframe systems. We noticed that the traditional Agile/Scrum has some drawbacks which make it not the perfect software engineering paradigm for such applications. This is because of redundant time and meetings spent to discuss issues in first work. Thus, we need to develop such applications in the shortest possible time, and this led to the adoption of a modified software engineering tools to achieve a reduced life cycle development process. Obviously, the developers follow the traditional methodology/framework for developing their applications.

To speed up the development process, we can move beyond this by improving and customizing the Scum technique for such applications.

Contribution

The goal of this work is to develop/improve existing traditional software methodologies to speed up the developing process of small applications for mobile devices (handheld devices, smartphones, tablets, etc.). In other words, the aim of this research is to improve and customize the standard Agile/Scrum methodology to achieve our goal to speed up the development process for some kind of applications which have very similar interfaces, style and functionalities. In order to do that, we carried out a survey on mobile multimedia application development frameworks. After that, we developed a small multimedia mobile application using one of the cross-platform application developing tools, applying the new suggested modifications in the developing process to develop a series of similar applications. We believe that developed software technique can be suitable and fulfill the needs for such applications development for mobile devices efficiently. As a result and conclusions, we came up with some suggestions for developing a small multimedia mobile application of such category.

Thesis Organization

In chapter two, we introduced a background for handheld devices applications development and platforms, differences between traditional software engineering and mobile software engineering, kinds of application development (Native, Web, and Hybrid), Agile Scrum, and introduce statistics on Smartphones, in terms of hardware and software. Chapter three covers intensive literature review about multimedia software mobile applications development, mobile development frameworks and cross-platform mobile development. In chapter four, we introduce our approach and methodology of research. Chapter five clarifies the methodology for our approach, results of our survey, Scrum development team, our experiment description, our case

study application we have made, testing and validation in details, and our suggestions and modifications to Scrum. Finally, chapter six concludes our work and suggests future works.

Chapter Two

Background

In this chapter we introduce a background regarding handheld devices applications development and platforms, differences between traditional software engineering and mobile software engineering, kinds of application development (Native, Web, and Hybrid), Agile Scrum, and introduce statistics on Smartphones, in terms of hardware and software.

Mobile Application Platforms

There are different mobile platforms running different operating systems in Smartphones. These OSs manage mobile hardware, multimedia functions, and connectivity, as well as phone access. Android, Apple iOS, BlackBerry, Windows Phone, and Symbian are the most well-known mobile platforms. Most of these platforms are tied to specific device hardware.

In this section we will make a comparison between the different mobile platforms in terms of architecture, SDKs, and others, as illustrated in table 1 below (Dorokhova & Amelichev, 2010).

A. *Symbian:* Open-source OS, developed by Symbian Foundation (Nokia, Sony Ericson, Motorola, and others).

B. *iPhone:* Developed by Apple. Its internal structure is kept secret.

C. *Android:* Open-source OS. It is maintained by Open Handset Alliance (headed by Google).

D. *Windows Phone:* Unveiled by Microsoft in 2009. It is a totally re-worked platform. It has integrated Bing search. Native applications are prohibited to install and run.

E. *BlackBerry:* BlackBerry is developed by Research in Motion (RIM). Hardware specifications are a secret.

Mobile Application Development

Developers began to produce applications for these devices to let users deal with them effectively. It became important to develop applications in line with the capabilities of the Smartphones. Although these devices match PCs, they have some drawbacks like battery drains, significantly if it is used frequently, or by using graphical applications. Another drawback is small screen, which has some limitations in applications. Other drawback like connectivity, which is unreliable because it varies in some environments.

There is a difference between traditional software engineering and mobile software engineering, where there are additional characteristics in mobile computing, like power consumption, security, testing complexity, user interface, native and web application, and context awareness (Wasserman, 2010). On the other hand, there are different hardware characteristics, which are screen size, battery life, sensors, and network connectivity (Dehlinger & Dixon, October 2011) (Ha, et al., October 2012) (Hartmann, Stead, & DeGani, March 2011) (Intel IT, August 2012).

Table 1: Comparison between the most well-known mobile platforms **(Wikipedia, 2015)**.

Feature	iOS	Android	Firefox OS	Windows Phone	BlackBerry OS
Company	Apple, Inc	Open Handset Alliance/Google	Mozilla Foundation	Microsoft	BlackBerry Ltd.
Market share	11.9%	84.6%	N/A	2.7%	0.6%
Current version	8.1.1	5.0	1.4.0	8.1.14147	10.2.1.3247
License	Proprietary EULA except for open source components	Free and open-source, but usually bundled with proprietary apps and drivers	Free and open-source, mainly the MPL; Apache	Proprietary	Proprietary
OS family	Darwin	Linux	Linux	Windows NT 8+	QNX
Supported CPU architecture	ARM, ARM64	ARM, x86, MIPS and the 64-bit variants of all three	ARM, x86	ARM	ARM
Programmed in	C, C++, Objective-C, Swift	C, C++, Java	HTML5, CSS, JavaScript,[9] C++	7+: XNA (.NET C#), Silverlight, native C/C++ (only for vendors and partners) 8+: .NET C#, VB.NET, Silverlight, native C/C++, WinRTP (XMLA), DirectX	C/ C++: Native SDK, C++/Qt: Cascades SDK, HTML5/Javascript /CSS: Webworks SDK, ActionScript: Adobe AIR, Java: Android runtime
Package manager	iTunes	APK	Firefox OS Packaged Apps	Zune Software (not since Windows 8)	BlackBerry Link
Official SDK platform(s)	Mac OS X using iOS SDK	Linux, Mac OS X and Windows	All where Firefox is available	Windows	Windows, Mac OS X, Linux (only Native SDK for 10+)

Smartphones have different platforms, and the most well-known platforms are Android, iPhone iOS, Windows Phone, Blackberry, and Symbian OS. Developers have to build applications for each platform, which takes more time and money. Smartphones have built-in internet browser, this helps in doing applications that run on multiple platforms, but it cannot deal directly with the mobile hardware, like camera, screen, and sensors (Wasserman, 2010).

Technology in mobile development precedes research in many steps, because of the huge competition between mobile companies. Apple iPhone 5, Samsung S3 and S4, and many other Smartphones have modern features, which make developers make advanced applications in

sound, camera, display, and others. From these advanced apps: Siri for iPhone, Google voice, flouting touch, 360˚ Photo Sphere Camera picturing and many other functionalities.

It is important to note the overall experience a user has with the application. Certainly, to adopt an application, it is important to have a successful user experience, which can be divided into two categories: The context – multiple contexts (Uncontrolled elements, such as hardware affordances, platform capabilities and UI conventions) and the implementation (Controlled elements, such as performance, design, and integration with platform features) (Charland & Leroux, 2011).

It is also important for a successful application to consider all the hardware devices, such as the display (physical size, color depth, screen resolution, pixel density, aspect ratio); the input (trackball, touch screen, physical keyboard, microphone, camera); and the capability (processing power, storage, antennae, and so on). Also must consider the platform conventions, as each platform has its own user interface (Charland & Leroux, 2011).

The life cycle of mobile application development is faster than traditional programs. This requires updating the applications continuously, thereby requiring quicker ways for making these applications. Software developers became interested in developing applications for the *Smartphones*, where there have been many platforms; these systems are Android, iOS, Windows Phone, BlackBerry and others. Each platform requires making a different application.

There are different categories of applications in smartphone stores, which opened a wide range of areas in Entertainment like Games, Services, Social, Educational Applications, and Critical applications like Business, most of them are multimedia applications. These applications are implemented in a different way from other computer systems. As mobile phones became more important, some companies force their employees to use these devices in their job, especially outdoor.

For saving time and effort, there exist tools for making cross-platform applications, where developers only need to write the code once and then deploy it across multiple platforms. Another important thing for developing applications in quicker ways is using better methodology for mobile application development. Agile practice is the best for that, using one of its methodologies, which is the Scrum (Flora & Chande, 2013); even it has minor problems with relatively small mobile applications.

In mobile computing, there are several constraints not present in desktop computing, including wireless communication problems, mobility issues, variety of platforms and technologies, limited capabilities of terminal devices, and time-to-market requirements. These constraints should be taken in consideration when developing mobile applications. Also high competence and large app stores play important role in developing mobile applications (Corral, Sillitti, & Succi, 2013).

Because of increasing demand of many kinds of applications, and increasing of kinds of devices, which needs a fast production to reduce life cycle, while such applications on PCs differ than on

Smartphones, as the latter main characteristics are small screen which needs more simpler interface with minimum number of details, for better readability. From this point of view, we interested in seeking for better development ways for achieving such approach. The traditional approaches in designing such applications needs relatively long life cycle.

Many researchers recommend Scrum approach in designing mobile applications, but this approach still has long life cycle, and needs iterative work, which consequently the life cycle of developing small mobile applications will be too long, therefore, we choose to improve Scrum approach, by reduction of some Scrum steps and reducing its time. In mobile development, the programming process is different; it may be for companies like for example “Angry Bird” which needs a whole Scrum Team and process, where some applications can be made by individuals, like “Flappy bird” which was invented by one person. He served as the Product Owner, The development Team, and The Scrum Master, with only the help of a multimedia specialist, in which he applies all steps of the Scrum methodology by himself.

We adopted Scrum to focus on the work of developing mobile applications, and we selected one topic in Chemistry to be a case study to test our approach. For such topic, we asked a teacher (Subject Matter Expert in Chemistry) for implementing a chemical experiment on Smartphones, to obtain the outcomes of this experiment.

Mobile Application Categories

According to our survey (Sansour, Kafri, & Sabha, 2014) on market, we found that application development on mobile Smartphones can be classified into three kinds of applications, which

are Native applications (runs entirely in the mobile device), Web applications (on a web browser and executed on a remote server), and Hybrid applications (works like a Native, it is a mixture of the Native and the Web application) (BayTechServices, 2013) (Charland & Leroux, 2011) (IBM, April 2012).

Software Development: Agile & Scrum

Because Mobile application development moved from a process-intensive to more Agile approach, mobile developers are following a Scrum like process, even individual ones (Wasserman, 2010).

Agile methodologies help companies build the right product by accommodating changing business requirements and better software development management, facilitating communication between the developer and the customer, and be ready for change at any time. Agile empowers teams to redesign continuously their release, optimizing its value. Agile Manifesto enables high-performance, efficiency and outputs by detailing four core values: *1- Individuals and their interactions, 2- Delivering working software, 3- Customer collaboration, and 4- Responding to changes* (Flora & Chande, 2013).

The authors in (Dehlinger & Dixon, October 2011) counted on Agile as one of the challenges for mobile application software engineering development mentioning that Agile approach is been utilized making the applications be self-adaptive using ad-hoc approaches. Applications then are required to independently modify its behavior and provide less functionality than providing none.

In (Ferdiana, December-2012) the authors adopts an Agile method eXtreme Programming (XP) to deal with the multichannel application developing difficulties.

In (Flora & Chande, 2013) the authors mentioned some researches that highlighted the suitability of Agile software development in mobile apps, as of small teams, short deadlines, importance of usability, fast delivery, offers opportunities and value, and it is best for development and non-sequential projects.

In (Mahalakshmi & Sundararajan, 1 Feb. 2014) the authors present a glimpse of all Agile methodologies concentrating on Scrum as the best among all, which is based on iterative development. Agile has many methodologies which are: *Adaptive Software Development "ASD"* (replaces the traditional waterfall cycle, for continuous learning, uses time boxed), *Agile Unified Process "AUP"* (focuses only on high value activities and independent to use any set of tools), *Crystal Methods "Crystal Clear"* (It focuses on project efficiency, and always focuses on people instead of process), *Dynamic Systems Development Method "DSDM"* (It is an iterative and incremental approach, focuses on frequent delivery of products), *Extreme Programming "XP"* (To improve quality of software, it has built-In Quality), *Feature Driven Development "FDD"* (It is iterative and incremental software development process, it delivers results at all steps), *Kandan* (For managing the products creation and delivering the project just-in-time without giving burden to the team), and finally *Scrum* (A team pack, it is to deliver the project just-in-time and minimal cost, used for building software project, and for planning any kind of work) Table 2 illustrates a comparison between Agile methodologies.

***Table 2: Comparison between Agile methodologies* (Mahalakshmi & Sundararajan, 1 Feb. 2014).**

Characteristics	Development approach	Size of the Project	Documentation
ASD	Iterative	Small projects	Basic documentation
AUP	Iterative and Incremental	Small or large Projects	Less documentation
Crystal Methods	Incremental	All types of Project	Basic Documentation
DSDM	Iterative	All types of project	Documentation exist
XP	Iterative	Small projects	Basic documentation
FDD	Iterative and Incremental	Complex projects	Documentation is important
Kanban	Incremental	Small or large projects	Basic documentation
Scrum	Iterative	All types of project	Basic documentation

In this section, Agile Scrum for developing mobile applications will be explained in details.

SCRUM

Scrum is an iterative and incremental framework for addressing complex adaptive problems, optimizing predictability and control risk, which is productive and creative in delivering products of the highest possible value. It is ***lightweight***, ***simple to understand***, and ***difficult to master***. With Scrum, developers can employ various processes and techniques. They can improve product management and development practices. The Scrum framework consists of *Scrum Teams* and their associated roles, events, artifacts, and rules, in which each component is for a specific purpose and has an important role in the success of the Scrum (Schwaber & Sutherland, July 2013), see Figure 1.

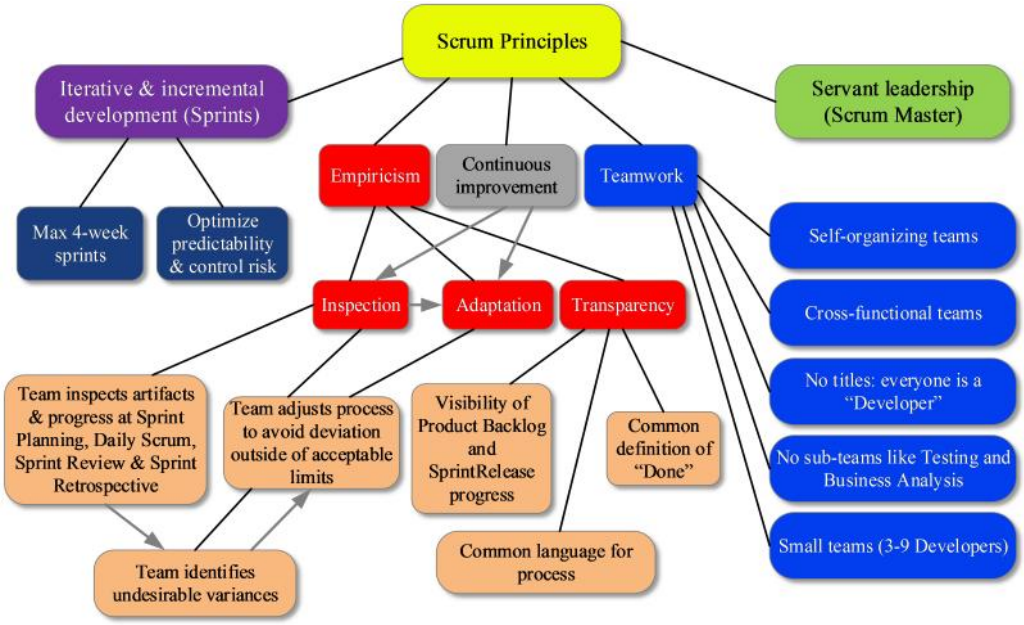


Figure 1: Scrum Principles (Habib, 2013)

Scrum employs an iterative, incremental approach to optimize predictability and control risk, which keeps up the application constantly improving. By the inspection process, the *Product Owner* decides if the outcome can be released or discard. This makes the outcome very coherent and credible. While the Scrum techniques maintain a strict focus on delivering products, at the same time it can notice the technical risks easily. The iterative/incremental object-oriented development cycle is enhanced by Scrum (Software Development Using Agile & Scrum, Nov. 17-18, 2012), as shown in figure 2.

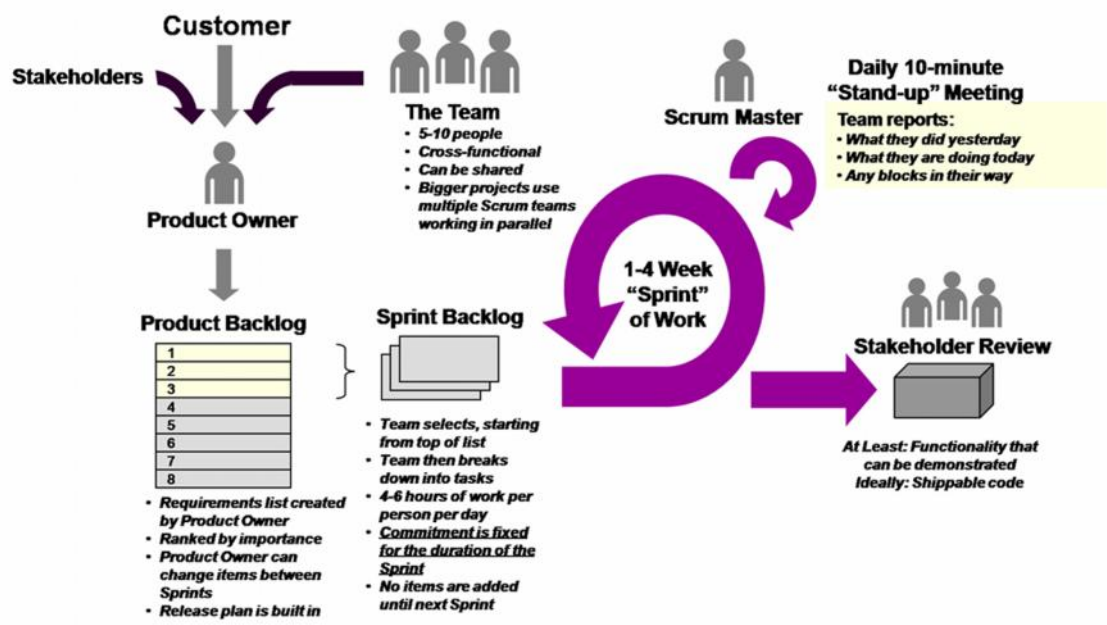


Figure 2: Agile SCRUM process flow (Rao, 2010)

In Scrum, combining known and workable tools with the devise of the best development team, development process is defined as a loose set of activities for building systems (Javagal, Rantakala, Rajeshwari, & Nandita, Nov. 17-18, 2012).

Scrum Team is ***self-organizing*** (Choosing how to best accomplish their work) and ***cross-functional*** (Internally have all competencies to accomplish their work). The Scrum Team consists of a *Product Owner* (Responsible of maximizing product value and Development Team work), the *Development Team* (Consists of professionals who deliver releasable Increment of "Done" product at the end of each Sprint), and a *Scrum Master* (A servant-leader for the Scrum Team, responsible for ensuring Scrum understanding). Scrum Team is designed to improve flexibility creativity and productivity, maximizing opportunity for feedback (Schwaber & Sutherland, July 2013).

Scrum Team can measure an experiment usefulness, after a short period, by ***Transparency*** (Important aspects must be seen and reviewed by those responsible for the results), ***Inspection*** (Scrum artifacts must be inspected frequently to detect undesirable variances), and ***Adaptation*** (after inspection of undesirable variance, it must be adjusted to minimize future deviation) (Schwaber & Sutherland, July 2013).

The Scrum heart is the Sprint. It is a time-boxed one-month or less during the creation of the increment. After each conclusion of a Sprint, a new Sprint starts. The Scrum Team collaborates with stakeholders about what was done in the Sprint (Schwaber & Sutherland, July 2013).

The Sprint consists of Sprint Planning Meeting, Daily Scrums, the development work, the Sprint Review, and The Sprint Retrospective (Schwaber & Sutherland, July 2013).

The Sprint must not be more than one month to avoid complexity and risk increment. If the Sprint takes a long time more than one month or it does not make sense, it will be cancelled, only by the Product Owner (Schwaber & Sutherland, July 2013).

In (ITSpree, 2011) the internet software house ITSpree company uses Agile/SCRUM with QA & testing, as a way of planning, executing, reporting and finally deployment of a project, to deliver super results, providing high quality IT services, and web & mobile applications development. ITSpree typical project consists of *Planning and Designing*, *Development*, *Quality Assurance & Testing*, and *Knowledge Management*.

Sprint Planning Meeting:

The Sprint Planning Meeting is planning the work to be performed in the sprint. The Sprint Planning Meeting time is eight hours for a one-month Sprint. During the Sprint Planning Meeting they determine what will be developed in the sprint, and how the Development Team will build this functionality into "Done" (Schwaber & Sutherland, July 2013).

Daily Scrum:

The Development Team takes a 15-minute time-boxed daily event to reduce complexity and to synchronize activities (accomplishment, next meeting work, and obstacles) for the next 24 hours plan creation (Schwaber & Sutherland, July 2013), see Figure 3.

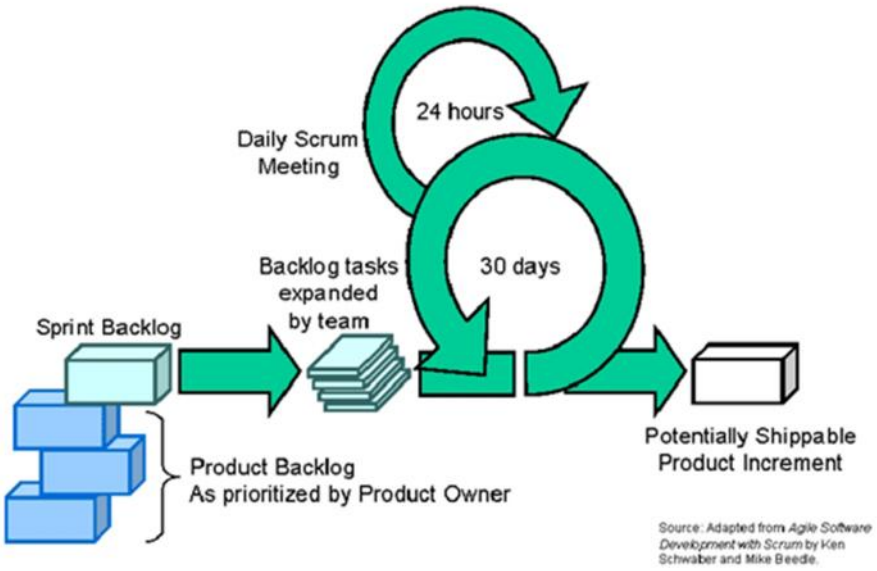


Figure 3: Daily Scrum (Lima, de Castro Freire, & Costa, 2012)

Sprint Review:

This is to inspect the increment and adapt the Product Backlog (A list of everything needed in the product and requirements for any product changes) if needed. It is held at the end of the Sprint. This takes four hours for a one-month Sprint. In every sprint review, the Product Owner tracks the total work, comparing the remaining work for the current sprint with the previous Sprint (Schwaber & Sutherland, July 2013).

Sprint Retrospective:

Sprint Retrospective is for the Scrum Team to inspect itself and creating an improvements plan. It takes a three hour for a one-month Sprint (Schwaber & Sutherland, July 2013).

Definition of "Done":

At the end of a Sprint, the Increment (The sum of all Product Backlog items) must be "Done". It is the completed work on product Increment. Finally, all Increments are added together and tested thoroughly to make sure that it will work together properly and as required (Schwaber & Sutherland, July 2013).

Advantages of Scrum

According to (Mahalakshmi & Sundararajan, 1 Feb. 2014), Scrum has some advantages which are:

- Scrum increases the quality of product / project.

- It is not only suitable for IT project, but is also suitable for non-IT projects also.
- Accepts and expects the changes.
- Benefits to customer and project manager.
- Scrum is fast methodology when compare to other methodologies.
- Easy to adopt changes.
- Work estimates are much easier.

Mobile Development with Scrum

Scrum can be used for all projects the same way, regardless of the different functionalities of the mobile devices. Mobile devices with its different functionalities, like wireless communication problems, mobility issues, variety of platforms and technologies, limited capabilities, screen sizes, power consumption, and time-to-market requirements, require fast, effective, and efficient responses to the changing requirements, and Scrum could be perfect for that with low price.

For the vast development of the mobile multimedia applications, this can deliver products of the highest possible value, and it can produce many applications in a short period. Also multiple versions of the same product is always available in short periods, as mentioned in (Holler, 2006) that Agile methods is lightweight, continuous and long-range planning as in mobile development, in which changes occur too frequently, which ensures more reliability and opportunity to users feedback.

To make increment in Scrum, the period is very short up to 4 weeks. Also, the number of the development team is good enough to make an increment within a specific period. The review

process is at very short intervals (daily), so that the adjustment must be made as soon as possible to minimize further deviation, which leads to consistent outcomes. Software is tested at frequent intervals (generally between 1 and 4 weeks). So mobile development needs simplicity, low overhead, business adaptability, rapid delivery, and customer feedback (Holler, 2006).

In (Su, 2011) the author discussed the adoption of Agile methodologies and Scrum in mobile application development, describing some management lessons from the mobile development. He pointed out that educators or project managers should clarify the concept of teamwork, with supervisory role to ensure overall communication.

As mentioned in (Habermas, July 29, 2013), Agile is suitable in mobile application development, as it gives developers freedom for choosing tools and incremental design.

In (Corral, Sillitti, & Succi, 2013) the authors assume that phone apps are simple, activity-centered, small number of actions, small team developers, which propose to use Scrum for mobile software development.

As Scrum shows the efficiency of a product management and development practices so one can improve it constantly that makes the application in line with the current development. The authors in (Corral, Sillitti, & Succi, 2013) proposed a framework with a customized Agile practices for mobile applications development. Based on collaboration, reduction development time, managing changes in requirements, variation of target platforms, and producing small-sized products, Agile methods are the solution for applications development.

Scrum Pitfalls

(ADELL, 2013), (Szkopiñski, 2012) and (Uhlig, 2014) found out that Scrum has some pitfalls in mobile development, which are:

1. Difficulty of testing on multiplatform because of multi vendors (Multi technology), as every device has its own technology, like quality, screen sensitivity, system power, memory, battery consumption, etc…, which makes it difficult to test the code as it recommended in Scrum to do a regression testing after each sprint.
2. The daily Scrum meetings and frequent reviews require substantial resources.
3. Stakeholders keep demanding for new functionalities to be delivered, because Scrum leads to scope creep.
4. Because it is needed to have many specialties in Multimedia, and although mobile projects are considered small, if one of the team member (specialists) are not there (have low commitment or left the team during the development), the rest of team will be delayed, which may lead to failure of the project.
5. Strict controlling by the Scrum Master makes the Team Member feel frustrated, which may lead to failure of the project.

Chapter Three

Literature Review

This chapter covers intensive literature review about multimedia software mobile applications development, mobile development frameworks and cross-platform mobile development.

Multimedia Mobile Application development

There are different categories of applications in smartphone stores, which opened a wide range of areas in Entertainment like Games, Services, Social, Educational Applications, and Critical applications like Business, most of them are multimedia applications. These applications were implemented in a different way from other computer systems. As mobile phones became more important, some employees use their devices in their job, accessing web-based and native line-of-business applications (Intel IT, August 2012).

Developers are interested in Smartphone production, because of its advanced capabilities, in video and music, cameras, GPS, wireless, and many other capabilities, as it achieves ubiquity. An example of multimedia for ubiquity is video streaming such as TV shows, sports, and news anywhere.

Some of the researchers like Zhang J. et. al. (2009) interested in increasing the efficiency of the smartphones in multimedia applications, such as multimedia streaming, VoIP, and mobile TV, as to utilize the limited power source (very limited battery capacity) and network resources to achieve a good balance of the power consumption between computation usage and

communication usage. They stated that major signal processing units in codec has tradeoff the power consumption. On the other hand, video communication over the network increases the power consumption (Zhang, Wu, Ci, Wang, & Katsaggelos, 2009).

They reviewed power-aware component latest advances for mobile multimedia systems. They are concerned about some points in understanding power managements in mobile devices like: Dynamic Voltage and Frequency Scaling, Maximizing Available Battery Capacity, and Power-Aware Transmission and Buffer Management. (Zhang, Wu, Ci, Wang, & Katsaggelos, 2009).

Mobile Software Engineering

There are some immature researches in finding some software engineering roadmap development. Some of the authors interested in context-aware mobile computing, another interested in different platforms development, others interested in comparing between mobile software engineering and traditional software engineering.

Some of the authors like Buthpitiya et. al. (2010) considered that mobile phones and portable PCs with connectivity and sensing capabilities would achieve ubiquitous context-aware computing. They highlighted some challenges face the mobile developers which are mostly the limitations of the mobile device which are **Limited resources, Large number of information sources, Increased security risk, Usability issues,** and **Component and information reuse** (Buthpitiya, Cheng, Sun, Griss, & Dey, 2010).

The authors described a designed framework called Hermes, for context widgets. They designed a widget that can discover other widgets providing the required context information it requires. Each widget ensures high level of security for the application. (Buthpitiya, Cheng, Sun, Griss, & Dey, 2010).

As for a context-aware applications Ojala et. al. (2003) presented a multimedia middleware mobile service system called SmartRotuaari, which is a set of services offering personalized mobile direct marketing, information communication and payment services. SmartRotuaari is a web portal with content provider interface (CPI), a collection of functional context-aware mobile multimedia services that consists of a wireless multi-access network. It provides a functional research framework for prototyping and empirical evaluation of context-aware mobile multimedia services, customer behavior and business models (Ojala, et al., December, 2003).

Biegel & Cahill (2004) early considered the context awareness and mobility are the main concepts for the ubiquitous computing. They developed a framework that eases the development of mobile context-aware applications which make developers build their own applications fast and easily, allowing them to collect and merge data from sensors, context, and reason about context. It also provides an efficient approach to intelligent reasoning based on a hierarchy of contexts, and an event-based communication between sensors, objects and actuator (Biegel & Cahill, 2004).

Some authors, like Charland & Leroux (2011) compared the native and the web applications. They stated that Native applications are better for mobiles, but this faces two problems, first, the

native applications is very expensive because developers have to develop different applications for each platform separately, second these applications is faster and uses more resources, which affects in less performance. Native code is faster than JavaScript, but in low CPU, JavaScript obtained better performance and longer battery life. (Charland & Leroux, 2011).

They stated also that to adopt an application, it is important to have successful **User Interface**, and **User Experience: Context vs. Implementation** (Charland & Leroux, 2011).

The authors added here that it is important for a successful application to consider all the hardware devices, such as the *display*, the *input*, and the *capability* (Charland & Leroux, 2011).

Bareiss & Sedano (2011) tried to find some special needs for mobile applications to improve the quality of mobile applications. They defined the End-user software engineering as “systematic and disciplined activities performed by end-user programmers that address software quality concerns”. They mentioned the different platforms in mobile applications, adding that reuse includes both patterns (at user interface level by platform design) and code (at sensor interfaces). They mentioned that mobile applications must be exciting and attractive and better designed way to better meet the user's needs, to have higher adoption rates, also user testing is important to discover requirements (Bareiss & Sedano, 2011).

Carbon & Hess (2011) put some methods for developing a business mobile application, in order to be used in business-critical situations. They developed a mobile business application offering

valuable mobile services to customers. They found that user experience, security, and flexibility are beyond key quality requirements (Carbon & Hess, 2011).

On the other hand, Dehlinger, J. & Dixon, J. (2011) counted on the rapid development of the mobile platforms categorizing four mobile application challenges for mobile application software engineering which are: **1-Creating Universal User Interfaces. 2-Enabling Software Reuse across Mobile Platforms.3-Designing Context-Aware Mobile Applications. 4-Balancing Agility and Uncertainty in Requirements.**

The authors also mentioned four research directions in software engineering fields that contribute to mobile application software engineering: **1- User Interfaces for the Differently-Abled. 2-Mobile Application Software Product Lines.3- Context-Aware Applications.4- Self-Adaptive Requirements** (Dehlinger & Dixon, October 2011).

Rosa et. al. (2012) considered that mobile new features improved communication for users' efficiency to access information by new approaches like social networks, blogs, and Web pages. GPS has become used dramatically in applications, because of the existence of location-aware and context-aware technologies in mobile devices. Smartphones with GPS capability can be used as a personal navigator, and because of the portability of smart mobile devices, it is possible to develop intelligent mobile multimedia applications, where smart mobile devices can access multimedia content over the internet. (Rosa, Dias, Lopes, Rodrigues, & Lin, 2012).

The authors proposed a multimedia application called CityEvents, which follows location, context, and context-awareness approaches. This application allows users to have information about culture events with multimedia contents, anywhere and anytime showing information to the user on a table or on a map (Rosa, Dias, Lopes, Rodrigues, & Lin, 2012).

Ha et. al. (2012) describe the benefits of cloud computing for mobile devices, in which it helps to improve processing, storage, and energy limits, as it reduces latency significantly in the critical path of user interaction. Cloud can reduce the mobile energy consumption by the processor, memory and storage (Ha, et al., October 2012).

They performed a set of experiments on some familiar applications that are available anywhere and anytime. These were: *Face Recognition*, *Speech Recognition*, *Object and Pose Identification*, *Mobile Augmented Reality*, and *Physical Simulation and Rendering* (Ha, et al., October 2012).

About software development, some researches described ways for developing applications in quicker ways, by using better methodologies.

Wasserman (2010) stated the fact of the accelerated growth of mobile applications since the emergence of Mobile stores, in addition to some characteristics of these applications namely small size, one or two developers. He focused on individual developers, stating the difference between Native and Web applications. He also worked on the differences between mobile and traditional software engineering. (Wasserman, 2010).

He focused on the differences between mobile applications and traditional software engineering. He showed that mobile software engineering needs urgently more than development techniques (Wasserman, 2010).

He explained that as mobile application development is similar to software engineering, in addition to common and traditional issues like security, performance, reliability, and storage limitation, there are additional requirements for mobile applications development. These requirements are: ***Potential interaction with other applications***, ***Sensor handling***, ***Native and hybrid***, ***Families of hardware and software platforms***, ***Security***, ***User interface***, ***Complexity of testing***, and ***Power consumption*** (Wasserman, 2010).

He also mentioned some of the most promising areas for software engineering which are the **user experience**, **non functional requirements Processes**, **Tools and Architecture** and **Portability** (Wasserman, 2010).

Finally, he added that people in mobile development have moved from a process-intensive to more Agile approach, finding widespread acceptance and because of the complexity of mobile applications (Wasserman, 2010).

Ali, N. & Ramos, I. (2012) mentioned the importance of software architecture, which focuses on the design, and specification of overall system structure. They talked about Aspect-Oriented Software Development and how it reduces complexity by increasing reusability, flexibility, and maintainability. They concerned about explaining the steps of designing software architectures of

the mobile applications, using Aspect-Oriented Architecture Description Language (AOADL) which follows the Ambient-PRISMA approach that enables designers to notice the notion of location and mobility, by providing a set of guidelines to facilitate detecting distribution and mobility characteristics (Ali & Ramos, 2012).

Flora, H. & Chande, S. (2013) aim to analyze prevalent methodologies utilizing Agile techniques. They mentioned that mobile applications need assimilation with databases, and Web services, lifecycle of a mobile app moves much faster than that of a traditional Web app, also in Security and testing. In order to meet the new requirements of the software development, Agile methodologies must be introduced. They mentioned that Agile methodologies help companies build the right product by accommodating changing business requirements and better software development management, facilitating communication between the developer and the customer, and be ready for change at any time. Agile empowers teams to continuously redesign their release, optimizing its value (Flora & Chande, 2013).

Software engineering is similar to mobile application development in integration with device hardware, traditional issues of security, performance, reliability, and storage limitations, but there are some limitation that are found in mobile development only including: interface with other applications, handling device movement, mobile application types, cross platform compatibility, varying hardware complexities, security risks, privacy, user interfaces, testing complexity, power consumption, application size, and screen size (Flora & Chande, 2013).

Ferdiana, R. (2012) provides a simple lightweight framework addressing the technical and non-technical side in mobile software development, targeting small and medium business, taking in consideration limitation of resources, time, and budget. He also mentioned that developers face difficulties when developing multichannel application which are: - Choosing the right application type, - Paying attention on the selecting of technology, instead of focusing on the problem itself, and - Users may not like application that is done (Ferdiana, December-2012).

Corral, L. et. al. (2013) concerned about finding customized development processes to meet the challenges for rapid development of high quality and low price mobile applications that differs from desktop applications, like execution environment, limited resources, high autonomy requirement, market competition, etc. They present in the research the Agile software development processes, discussing if the Agile paradigm was adopted, dismissed, or created new one. (Corral, Sillitti, & Succi, 2013).

The main goal of their research is to unveil whether using Agile, ignoring, or creating different one for mobile development. They proposed a framework with a customized Agile practices for mobile applications development. Based on collaboration, reduction development time, managing changes in requirements, variation of target platforms, and producing small-sized products, Agile methods are the solution for applications development (Corral, Sillitti, & Succi, 2013).

Javagal, B. et. al. (2012) described an involvement of developers to develop some mobile platform applications following Agile and Scrum. They attempted to provide additional end-to-

end support, such as RTC and Rally by generating the burn down charts, which is maintained by Scrum Master. They added that in Scrum, combining known and workable tools with the devise of the best development team, development process is defined as a loose set of activities for building systems. In their research, the authors described involving developers in global software development following Agile and Scrum. (Javagal, Rantakala, Rajeshwari, & Nandita, Nov. 17-18, 2012).

Mobile Development Frameworks

Many researchers are interested in mobile applications development, in which they concern about the problems facing them, producing frameworks to meet the requirements of application developing. Some of the researches are concerned about context-aware development to achieve ubiquity, as the existence of sensors, Wi-Fi, and GPS in the mobile devices, which help users to benefit from these functionalities to get data from mobile environment. Other researches concerned in providing tools helping developers in their application development.

Technology in mobile development precedes research in many steps, because of the huge competition between mobile companies. Many researches describe frameworks for application mobile developing, which can be divided into several categories: Applications, Tools, or Guidance.

Cross-Platform Mobile Development

To support multiple OSs, Cross-Platform mobile development tools used to deploy native applications for many OSs. These tools mainly depend on web development languages. It makes the application run identically on different platforms.

In this thesis, we will explain many major available cross-platforms tools (Sansour, Kafri, & Sabha, 2014). These mobile application development tools make it easy to develop an application and save time and requires less coding.

Chapter Four

Methodology

In this chapter, we introduce the methodology for our approach, survey publication, Scrum development team, our experiment description, and our case study application we have made.

Our Approach

In order to find out the recent and mostly used development frameworks for multimedia mobile applications on different platforms, at the beginning, we made a detailed analysis supported by selective presentation and comprehensive comparison for published literature in the mobile application development frameworks, in addition to a comparison between multimedia mobile software engineering and traditional software engineering and what are the additional characteristics in the mobile software engineering. During that, we studied different mobile platforms and the capabilities of the new mobile phones, making comparison between them, in terms of the features in each of them, revealing what distinguish one from another and to find the strong and weak points in each of them.

For this purpose, we found that a set of experiments in one subject for e-learning can fulfill our requirement. Such applications have the same pedagogical rules, similar interface, style, and functionalities. Therefore, for our work as an example and case study, we choose some experiments in chemistry to be implemented as an application.

After that, we have formed the team consisting of one Product Owner, two developers, one Stakeholder and a Subject Matter Expert in Chemistry.

In this development we applied the Scrum Agile Software Engineering. In this software development, we were interested in making a small multimedia application for mobile, taking advantage of advanced functionalities and capabilities of the modern mobile devices (Smartphones). Some of these functionalities are Touch/Multi-touch Screens, Sensors and many others. We found some minor problems, where it was not suitable for this kind of applications. We tried to overcome these problems, to speed up the development process. We wrote down our experience in recommendations to modify the Scrum for "*Small Mobile Multimedia Application Development*".

We made the application with *Adobe Flash Professional CS6* as it has the ability to do applications dealing with high-performance tools for multimedia and for multiple platforms.

For the purpose of this research we recorded each event and duration in all phase and milestones of the development process.

In the next chapter we illustrate the progress of our experiments during the development process and the concluded results.

Chapter Five

Experiments and Discussion

This chapter consists of two parts. First part introduces the results of our survey on the latest and most well-known frameworks on mobile multimedia software engineering. The second part introduces in details our experiments progress in developing our applications/sub applications, using Scrum and the modified Scrum in different stages. Also we discuss our results of the developing progress.

Part One:

As we said before, our survey presented some results as a comparison for published literature in the mobile application development frameworks, in addition to a comparison between multimedia mobile software engineering and traditional software engineering and what are the additional characteristics in the mobile software engineering. Mobile application categories, and cross-platform environments, and some statistics on Smartphones in terms of hardware and software.

We found that application development on mobile Smartphones can be classified into three kinds of applications, which are Native applications, Web applications and Hybrid applications.

Native application:

Native application runs entirely on the mobile device; deals with the mobile hardware directly and effectively. Native code is compiled and faster than interpreted languages like JavaScript (Charland & Leroux, 2011), but this requires writing the code each time for each platform, as the code written for one platform does not run on another because of differences among the Software Development Kit (SDK) of each platforms, which increases cost and time. When the application initialized on the device, it interfaces directly with the mobile OS without any intermediary or container (Charland & Leroux, 2011) (BayTechServices, 2013).

Web application:

The web application runs on a web browser and executed on a remote server. It achieves the "Write-once-deploy-anywhere" solution, as it runs on multiple platforms. It does not deal directly with the mobile hardware. It is developed on standard Web technologies, like HTML5, JavaScript and Cascading Style Sheet (CSS) (BayTechServices, 2013) (IBM, April 2012). In low CPU performance, JavaScript obtained better performance and longer battery life also (Charland & Leroux, 2011).

Hybrid application:

The Hybrid application works like a Native (BayTechServices, 2013). It is a mixture of the Native and the Web application. This needs a third party, like the cross-platform applications. Since developers find it difficult to write the code on each platform, the cross-platform applications facilitate this issue. The software makers must balance between the Web and native

debate (Charland & Leroux, 2011). Cross-platform applications are the solution for developing effective applications reducing cost and time and with fewer skills. Developers write portion of code in Cross-platform technologies, and native portion APIs when required to create an embedded HTML using engine which acts like a bridge between the browser and the APIs of the device (IBM, April 2012).

According to our survey (Sansour, Kafri, & Sabha, 2014) we can categorize the mobile development frameworks as follows:

A. Context-Aware Frameworks

Many researchers wrote about ways for effective mobile applications, concentrating on getting data from mobile environment. Buthpitiya et. al. (2010) developed a framework for developing powerful Context-Aware applications. It is a tool, which make a widget discover other widgets taking context information it requires (Buthpitiya, Cheng, Sun, Griss, & Dey, 2010). Biegel & Cahill (2004) developed a framework that eases the development of mobile context-aware applications allowing the developers to collect and manage data from sensors, context, and reason about context (Biegel & Cahill, 2004). Simon & Fröhlich (2007) presented an application framework that takes advantage from the web geospatial content from interaction of the user interface and the high-end mobile phone devices (Simon & Fröhlich, May 8-12, 2007). Sz´ant´o (Fall 2010) extended the Java Context Awareness Framework to adapt to specific situation by making use of information, to help developing context aware applications (Szántó, Fall 2010).

B. Guidance Frameworks

Other researches describe frameworks as support and guidance for developing mobile applications. Inter created a mobile application development framework their employees as a set of specific capabilities, tools and resources to enable building successfully planned applications. It enables developers to provide feedback about what guidance worked best (Intel IT, August 2012). Cheng & Yuan (2007) proposed a framework that is capable to fit different devices or situations according to mobile user interface formats, computing power and functionality (Cheng & Yuan, 2007). Sierra Systems (2011) provided guidance on how to select mobile development framework to help reuse of existing enterprise assets in web applications (Sierra, 2011).

C. Cross-Platform Frameworks

Other researches made a description for Cross-platform frameworks; tools for developing successful mobile applications. Oracle (2011) created a framework which makes developers quickly develop applications for multiple mobile platforms. The application can access the device services, offering more experience for users than the browser can offer (Oracle, 2011). Singh & Palmieri (2011) made a comparison between major available cross-platform tools in the market, which are Rhomobile, DragonRad, PhoneGap and MoSync (Singh & Palmieri, 2011). Microsoft (2011) presents two mobile applications frameworks Silverlight and XNA. Silverlight and XNA applications require only a few small changes to run on multiple platforms (Microsoft, 2011).

D. Measurement Frameworks

Kim (2012) presented a model-based performance prediction at mobile software development time for project optimization to establish a goal-driven measurement system for mobile software development (Kim, 2012).

And as for Cross-Platform Mobile Development tools, according to our survey (Sansour, Kafri, & Sabha, 2014), these mobile application development tools make it easy to develop an application and save time and requires less coding.

Rhodes

RhoMobile Rhodes is an open source framework for cross-platform Smartphone applications, developed by RhoMobile. It aims to manage enterprise application and data, and to provide a high level productivity and web programming portability. The developed files of Rhodes are compiled into native executable programs, providing access to native device through a set of Ruby APIs. Developers do not need to learn SDK or each platform native language, but it is necessary to have web experience to make mobile application. It provides an (IDE) called RhoStudio. (Singh & Palmieri, 2011) (Ribeiro & da Silva, 2012).

The supported platforms are Android, BlackBerry, iOS, Windows Phone and Symbian, with free under the MIT license (Singh & Palmieri, 2011) (Ribeiro & da Silva, 2012).

PhoneGap

PhoneGap is an open-source framework for developing mobile applications. It is useful for applications using modern web applications. The application is hybrid and created by HTML5, CSS3 and JavaScript. It targets mainly web developers. It does not provide a unique IDE; so developers should execute the source code on each IDE. It provides a PhoneGap Build for compiling applications in the cloud. PhoneGap makes it easy to work with a short span of time, without the need of maintaining native SDK, it just do the job for developers, as it will be built with the recent SDK (Singh & Palmieri, 2011) (Ribeiro & da Silva, 2012).

The supported platforms are Android, iOS, webOS, Windows Phone, Symbian, Blackberry and Bada, with free Apache License, under MIT License (Singh & Palmieri, 2011) (Ribeiro & da Silva, 2012).

DragonRad

DragonRad is developed by Seregon (Ribeiro & da Silva, 2012). It focuses on database driven mobile enterprise applications with a WYSIWYG drag-and-drop tool, which provides the drag and drop visual environment GUI and helps developers to create logics. Its application is like a native application as it runs in the designer. It supports several databases such as MySQL, Oracle or SQL Server. It provides an owner IDE, which is a DragonRad Designer (Singh & Palmieri, 2011) (Ribeiro & da Silva, 2012).

The supported platforms are Android, Windows Mobile, and Blackberry, with a 30-day trial offer for the designer. After that its license is $4900 per year (Singh & Palmieri, 2011) (Ribeiro & da Silva, 2012).

MoSync

MoSync is an open source tool developed by Swedish company. It helps developer to build all types of applications, simple, advanced and complex, sharing the same code base. It provides the full fledge Eclipse-based IDE with the use of standard C/C++. MoSync application is a native application, where a service layer supports many functions like file I/O, threading, networking, memory management, and other functions. It provides an IDE based on Eclipse (Singh & Palmieri, 2011). Developers can add a map to their application by the MoSync Widget C API's Map, available only for Windows Phone 7 and iOS (Spiridon, 2012).

The supported platforms are iOS, Windows Phone, Android, JavaME, BlackBerry and Symbian, with free license (Singh & Palmieri, 2011).

Appcelerator Titanium

Appcelerator is an open source platform for developing mobile applications using web technologies. It links JavaScript to native libraries. The output application is a native code. It provides an IDE based on Eclipse called Titanium Studio. Appcelerator uses native UI and platform APIs (Ribeiro & da Silva, 2012), but to provide UI it does not use a browser engine on

the device. Data can be stored either in the cloud or on the device (Hartmann, Stead, & DeGani, March 2011).

The supported platforms are iOS, Android and BlackBerry, with a free under Apache License (Ribeiro & da Silva, 2012).

Sencha Touch 2

Sencha Touch 2 is a high performance open source framework for mobile applications. It enables developers to build fast applications. It takes advantage of hardware acceleration. A web server is needed to run locally for developing applications (Sencha, 2014).

The supported platforms are iOS, Android, BlackBerry, Kindle and Bada, with a free commercial license for application development, and a paid commercial license for OEM uses (Sencha, 2014).

jQuery Mobile

jQuery is a unified system for all popular mobile device platforms that is an HTML5-based user interface. The code is lightweight and built with progressive enhancement, and has a flexible design. It adopts the “write less, do more” concept. It does not create native applications. It has a broad support for the vast majority of all Smartphones, feature phones and older browsers (jQueryMobile, 2014).

The supported platforms are iOS, Android, BlackBerry, Windows Phone, Bada, palm WebOS, Symbian and MeeGo, with free under the MIT and GPL license (jQueryMobile, 2014).

Xamarin

Xamarin allows developing cross-platform applications written in C#. It produces a native application or integrated .NET application. It provides IDE (MonoDevelop IDE and Xamarin plug-in for Visual Studio). It is not necessary to have XCode IDE installed to develop for iOS, as it integrates with XCode Interface Builder (Xamarin, 2014) (Xamarin I. , 2012). Java SDK must not be installed to develop for Android when using Xamarin (Avram, 2013). On each platform, the application user interface uses native controls, taking advantage of native UI toolkits (Xamarin, 2014).

The supported platforms are iOS, Android, and Windows Phone. (Xamarin, 2014).

Unity 3D

Unity 3D is a cross platform 3D game engine, focusing on asset centric as a 3D modeling application. Developers can make motion applications (i.e. games) using Unity 3D engine (Winkler & Barrett, Fall 2011). It uses 3D modeling tools models and game objects. With Unity 3D, developers can access different motion sensors of mobile devices from run-time classes. Unity 3D tool provides a visual aid that reduces the amount of time for coding, as developers can simply drag codes from one object to another (Winkler & Barrett, Fall 2011) (Ipek, 2012).

During writing this thesis, the supported platforms are iOS, Android, and Windows Phone (Winkler & Barrett, Fall 2011). Unity 3D engine costs $399. The Unity Pro costs $1500 and additional $1500 for the Pro iOS version. It offers a 30 days trial for iPhone users (Ipek, 2012).

Corona SDK

Corona SDK is the leading mobile development cross-platform framework for building rich interactive applications, based on the Lua scripting language. Building applications in Corona SDK is easy and quick in a text editor (Ipek, 2012). Also developers can add features like Facebook and physics by writing a very short code, with Corona APIs (Corona, 2014).

The platforms it supports are iOS, Android, Kindle Fire and NOOK. Corona tool is a free unlimited trial, but to publish the application to the App Store and Android marketplace, developers have to pay $199/year for the license (Ipek, 2012).

Flash Professional

Flash Professional allows developing cross-platform applications, written by JavaScript Flash Language (JSFL) and ActionScript 3. With Adobe Flash Professional CS6 an interactive or games developer can create high-performance games to multi-platforms. It accesses mobile device capabilities, such as vibration, gyroscope, and multi-touch gestures. Applications with AIR can access Flash Player API, with enhanced functionality for vector-based drawing, multimedia support, and a full networking stack. It also embeds SQLite (Wagner, 2011). The supported platforms are iOS, Android, and Blackberry. It provides an IDE Air 3.2 SDK that must

be installed for Android and iOS. Air for iOS is part of the application file (large file size), while for Android, it is installed alongside the application (small file size) (Wagner, 2011).

Adobe Flash Professional CS6 has the ability to do applications dealing with high-performance tools for multimedia and for multiple platforms. It is a multimedia program used to create games, movies, web applications, and mobile phone applications (Adobe, 2012).

Large mobile development Statistics

Some statistics were gathered for large mobile development platforms to show how important these devices in all aspects of life, and its availability to all people.

On March 3, 2014, the total Apple Apps Approved for US App Store was 1,517,950, the total Active Apps "currently available for download" was 1,135,438, the total Inactive Apps "no longer available for download" was 382,512, and the total number of Active Publishers in the US App Store was 281,030, Figure 4 shows the categorized total number of applications in the Apple App Store (Mimbo, 2014).

On February 13, 2014, the total number of applications in Android market was 1,134,919 applications, categorized as shown in Figure 5 (AppBrain, 2014).

On March 8, 2014, Microsoft Windows published more than 190,000 applications in its marketplace. Figure 6 shows 190,458 applications published to Windows Phone Marketplace (WindowsPhoneAppList, 2014).

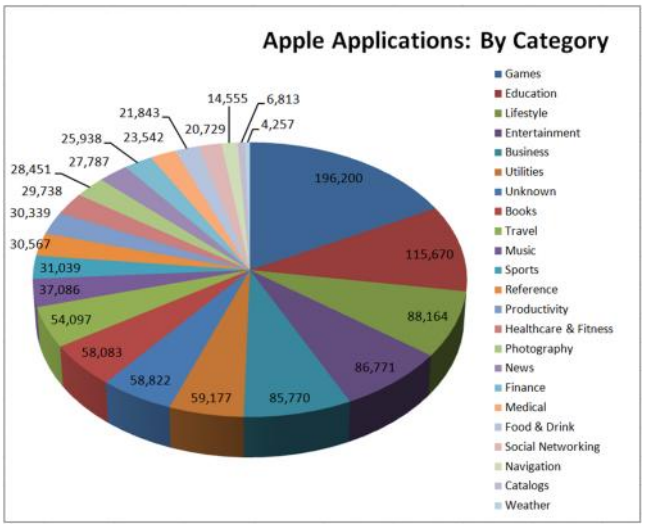


Figure 4: The number of available applications on the Apple market. Last updated on March 3, 2014.

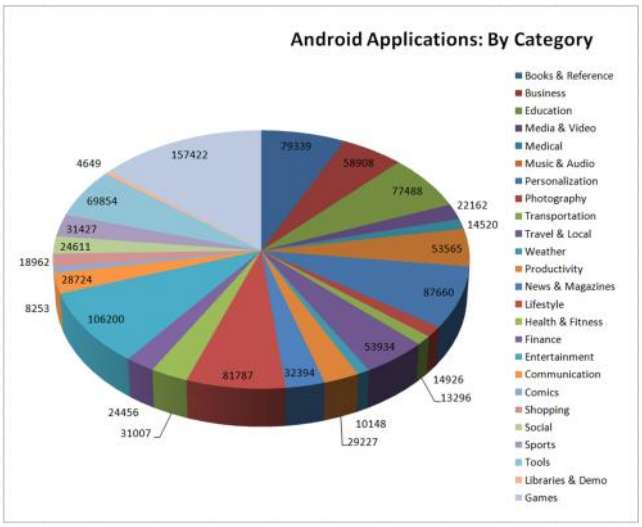


Figure 5: The number of available applications on the Android market. Last updated on February 13, 2014.

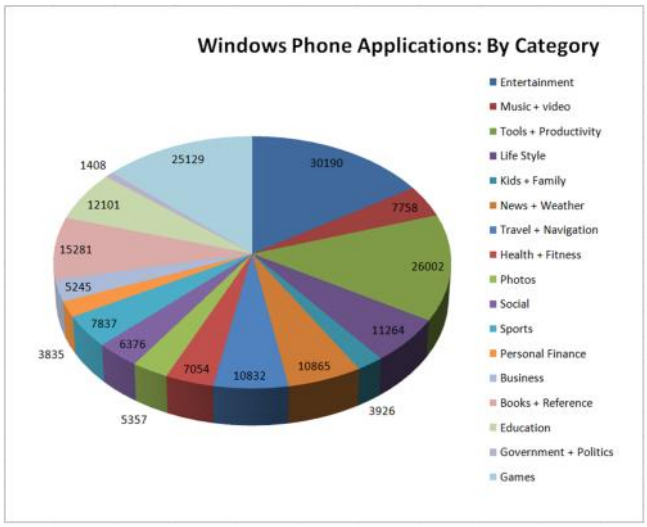


Figure 6: The number of available applications on the Windows Phone 7 market. Last updated on March 8, 2014.

We can see from figures 4, 5 and 6 that mobile applications became necessary in all aspects of life with high demand, where it became indispensable.

On September 27, 2013, the BlackBerry World now contains over 130,000 BB10 apps**,** as announced by Kunal Dua**,** As shared the Vice President for Developer Relations, Alec Saunders, that the number of BlackBerry 10 apps available in the BlackBerry World store now stood at 131,708, approximately seven months after the OS was released. He added that the application economy has grown significantly since the first debut of Apple and Google app store. They announced that by adding BlackBerry 7 apps to BlackBerry 10 apps, the total number of apps stood at 256,668 (Dua, 2013). Figure 7 shows a list of categorized applications for largest three platforms market, the Apple App Store, the Android Store, and the Windows Phone Store (Mimbo, 2014) (AppBrain, 2014) (WindowsPhoneAppList, 2014).

We can see from figure 7 that the largest amount of applications exist in these three platforms is Android and that because of the large amount of free applications.

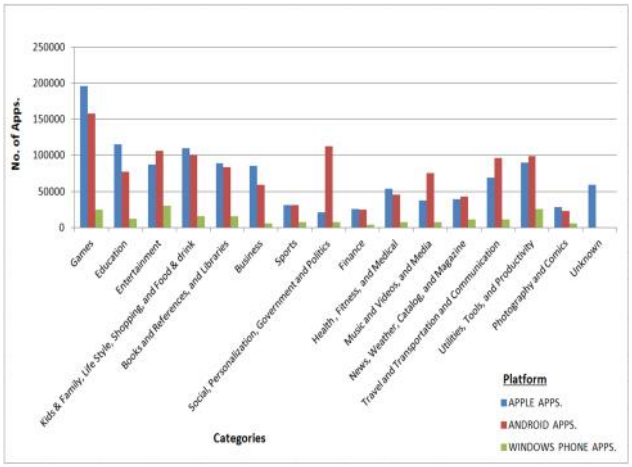
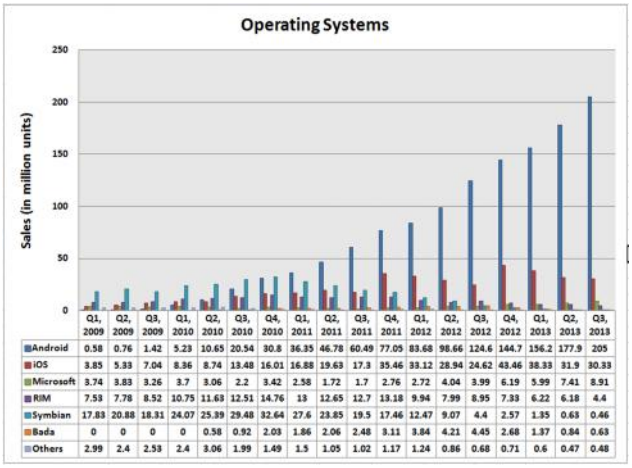


Figure 7: List of categorized applications for the largest three platforms market, the Apple App Store, the Android Store, and the Windows Phone Store.



	Q1, 2009	Q2, 2009	Q3, 2009	Q1, 2010	Q2, 2010	Q3, 2010	Q4, 2010	Q1, 2011	Q2, 2011	Q3, 2011	Q4, 2011	Q1, 2012	Q2, 2012	Q3, 2012	Q4, 2012	Q1, 2013	Q2, 2013	Q3, 2013
Android	0.58	0.76	1.42	5.23	10.65	20.54	30.8	36.35	46.78	60.49	77.05	83.68	98.66	124.6	144.7	156.2	177.9	205
iOS	3.85	5.33	7.04	8.36	8.74	13.48	16.01	16.88	19.63	17.3	35.46	33.12	28.94	24.62	43.46	38.33	31.9	30.33
Microsoft	3.74	3.83	3.26	3.7	3.06	2.2	3.42	2.58	1.72	1.7	2.76	2.72	4.04	3.99	6.19	5.99	7.41	8.91
RIM	7.53	7.78	8.52	10.75	11.63	12.51	14.76	13	12.65	12.7	13.18	9.94	7.99	8.95	7.33	6.22	6.18	4.4
Symbian	17.83	20.88	18.31	24.07	25.39	29.48	32.64	27.6	23.85	19.5	17.46	12.47	9.07	4.4	2.57	1.35	0.63	0.46
Bada	0	0	0	0	0.58	0.92	2.03	1.86	2.06	2.48	3.11	3.84	4.21	4.45	2.68	1.37	0.84	0.63
Others	2.99	2.4	2.53	2.4	3.06	1.99	1.49	1.5	1.05	1.02	1.17	1.24	0.86	0.68	0.71	0.6	0.47	0.48

Figure 8: Global smartphone sales to end users from 1st quarter 2009 to 3rd quarter 2013, by operating system (in million units).

Figure 8 shows Global smartphone sales to end users from 1st quarter 2009 to 3rd quarter 2013, by operating system (in million units) (Statista, 2014).

Figure 9 shows the quarterly worldwide Smartphones sales percentage to End Users by Vendors in 2011, 2012, and 2013 (Gartner, February, 2013) (Gartner, May, 2012) (Gartner, August, 2012) (Gartner, November, 2012) (Bora, 2013) (Biztech2.com, 2013).

Figure 10 shows the quarterly worldwide Mobile Phone Sales percentage to End Users by Operating System in 2011, 2012, and 2013 (Gartner, February, 2013) (Gartner, May, 2012) (Gartner, August, 2012) (Gartner, November, 2012) (MyBroadband.co.za, 2013) (Bora, 2013) (Biztech2.com, 2013).

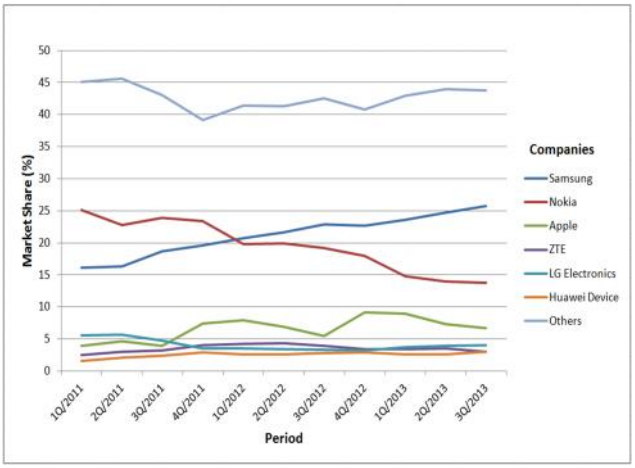


Figure 9: Worldwide Mobile Device Sales Percentage to End Users by Vendor in 2011, 2012, and 2013 (Thousands of Units).

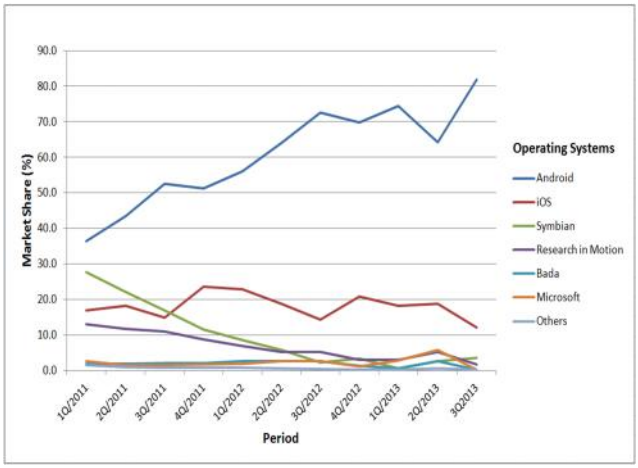


Figure 10: Worldwide Mobile Device Sales Percentage to End Users by Operating Systems in 2011, 2012, and 2013 (Thousands of Units).

On July 30, 2013, the Canaccord Genuity Group Inc. in its daily letter introduced Smartphone Unit Sales Estimates by vendors and Smartphone Sales Percentage Estimates by vendors, as shown in figure 11 and 12 (Walkley, Ramsay, & Sinha, 2013).

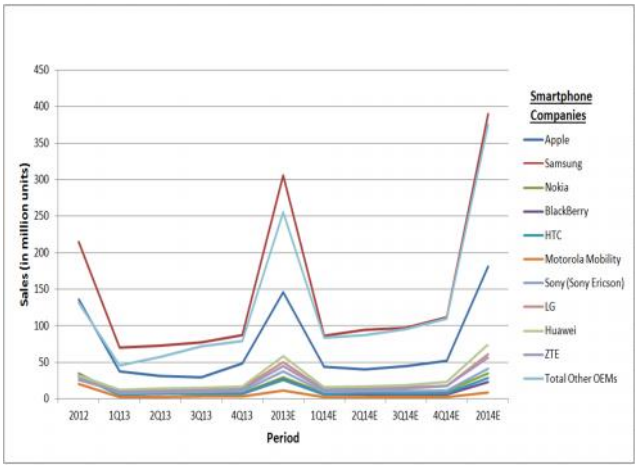


Figure 11: Canaccord Genuity Smartphone Unit Sales Estimates by OEM (millions).

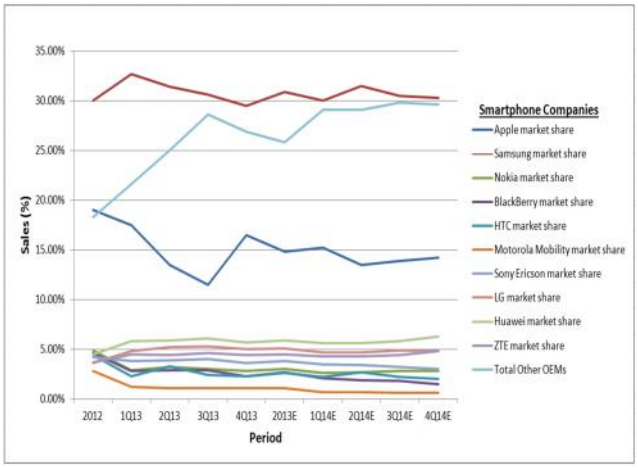


Figure 12: Canaccord Genuity Smartphone Unit Sales Estimates Percentage by OEM.

As for the worldwide mobile device sales by operating systems we can see from figure 8 and 10 that the largest amount of sales are for the Android platform. Also for the worldwide mobile device sales by vendors, as they are interested in providing the latest capabilities in mobile devices, we can see in figures 9, 11 and 12 that the largest amount of sales is from Samsung.

As we can see, the large mobile platform use is for Android first, iOS second and Windows comes third.

As a result we made a paper with the title “A SURVEY ON MOBILE MULTIMEDIA APPLICATION DEVELOPMENT FRAMEWORKS”, which was published a survey paper on

mobile software engineering at the Multimedia Computing and Systems (ICMCS), 2014 International Conference and appeared in IEEE Xplore, and was on:

- *Mobile Software Engineering.*
- *The latest and most well-known Platforms.*
- *Mobile application frameworks.*
- *Mobile Application Categories.*
- *Cross-platform environments.*
- *Statistics for large mobile development platforms*.

We concluded that the mobile multimedia application development frameworks are not mature neither well-defined. Some work is needed to well define a mobile software engineering.

Part Two:

As mentioned, we selected an experiment in chemistry subject for school education, namely "*Strong acid and Strong base neutralization reaction*" to carry out our development experiments. Since most such applications have the same pedagogical rules and similar interface, style, and functional requirements.

Application Development Stages

In this section, we discuss the stages of the development of our application based on Scrum methodology, and the problems we faced in each stage.

Stage 1: Application idea

As we decided to follow the Agile/Scrum methodology, we have formed the team consisting of one Product Owner, two developers, and one Stakeholder. We met to discuss the idea of the application we want to develop. In the first meeting, the idea was heading forward to do a mobile game using the most mobile functionalities, such as multitouch and sensors. At the next daily meeting, we decided that the application would be a football game, where we can control the players by the sensors and also we can use the multitouch to zoom in and out the screen, and some other functionalities. This idea took about two days of meetings, and after we began with development, we faced a problem, in that the application was not useful, where the stakeholder opinion was that the beneficiary is of the school students and in education field (e-learning). At the next daily meeting, we decided that the idea is to be a chemical experiment as an educational supporting e-learning tool.

We searched and came out of the "*Strong acid and Strong base neutralization reaction*". Since we have made sure of the idea, we began to apply the Scrum methodology. In the fifth daily meeting, we met with the Subject Matter Expert in Chemistry, where he became one of the team, and he explained and clarified the idea of the experiment. We took three days to study the experiment, and we have searched the internet also to see some explanations about the experiment. The whole team met daily for 15 minutes to discuss the work.

The next daily meeting was to discuss the programming tool. We intended to do the application on a cross-platform tool, as we found, in the survey we made, that it is the best for deploying applications on different mobile platforms. We decided to use the Adobe Flash Professional with

Adobe Illustrator, where we can deploy the applications of multi platforms, which are Android, iOS, and BlackBerry. This took two daily meetings.

We worked in total for 5 hours at home, in addition to the daily meetings, until we came out and decide what the application we must develop is.

Following the standard Scrum approach, the whole team with the Chemistry expert and the stakeholder, have met, and wrote down all the ideas and requirements of the application, and its backlog size and the estimated end date. The whole team made a Sprint Planning, meeting discussing roles and identifying the requirements of the project. The whole team put the Product Backlog for the application.

In this stage we followed the scrum methodology as it is. The whole team was in all daily meetings. To support our work, we used Microsoft Project in each stage, to analyze our work. Table 3 illustrates the time line of this stage, which is also illustrated in Figures 13 and 14.

Table 3: Stage 1 Time Line

Task Name	Duration	Start	Finish	Resource Names
STAGE 1 - Application Idea	**11 days (12.5 hrs.)**	**Thu 10/31/13**	**Thu 11/14/13**	
First meeting: Making up the idea	1 hr	Thu 10/31/13	Thu 10/31/13	All Team
Second meeting: Deciding to do mobile football game	1 hr	Sat 11/2/13	Sat 11/2/13	All Team
Third meeting: Discussing the application idea	1 hr	Mon 11/4/13	Mon 11/4/13	All Team
Fourth meeting: Last idea e-Learning App.	1 hr	Tue 11/5/13	Tue 11/5/13	All Team
At Home: Searching about the idea	2 hrs	Tue 11/5/13	Tue 11/5/13	Developers only
Fifth meeting: Discussion with the chemistry expert	15 mins	Wed 11/6/13	Wed 11/6/13	All Team
Sixth meeting: Studying the experiment	15 mins	Thu 11/7/13	Thu 11/7/13	All Team
Seventh meeting: Studying the experiment	15 mins	Sat 11/9/13	Sat 11/9/13	All Team
Eighth meeting: Studying the experiment	15 mins	Mon 11/11/13	Mon 11/11/13	All Team
Ninth meeting: Discussing the programming tool	15 mins	Tue 11/12/13	Tue 11/12/13	All Team

Tenth meeting: Discussing the programming tool	15 mins	Wed 11/13/13	Wed 11/13/13	All Team
At Home: Studying the programming tool	3 hrs	Wed 11/13/13	Wed 11/13/13	Developers only
Sprint Planning meeting	2 hrs	Thu 11/14/13	Thu 11/14/13	All Team

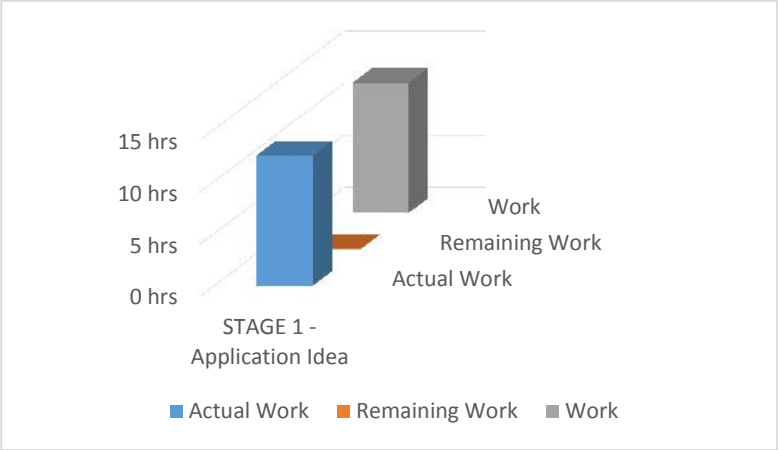


Figure 13: Stage 1 Work Chart

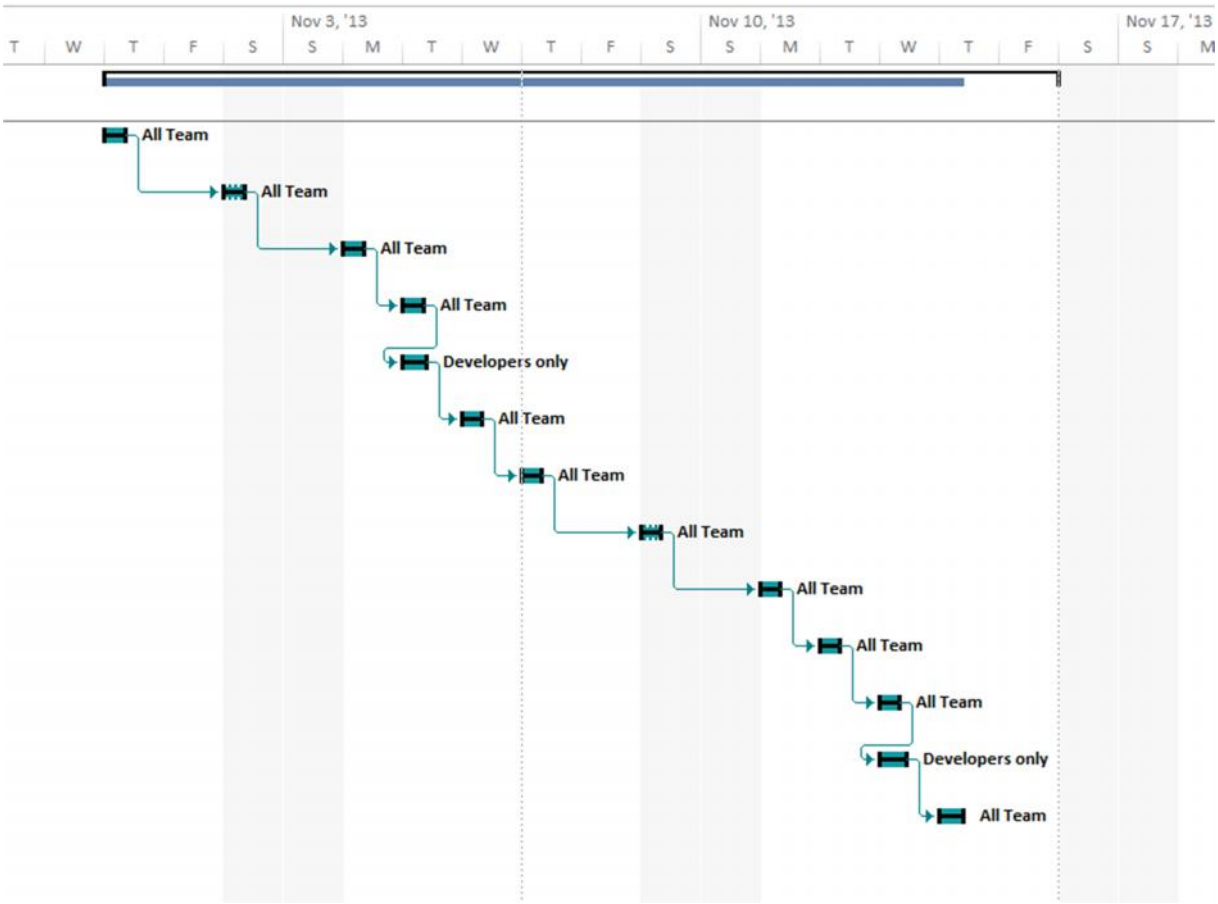


Figure 14: Stage 1 Gantt Chart

Stage 2: First Screen "Main Menu"

In each stage, we applied the development phases, (Analysis, Design, Development, and Testing).

Analysis Phase:

At the first daily meeting, we discussed applying the idea for the mobile, and we made an initial prototype for the application. One of the developers, at the next daily meeting, came up with the idea of the first screen as a prototype, as illustrated in figure 15, where he worked at home for 6 hours.

At the next daily meeting, we put the sprint backlog for the design of the first screen (main screen). The next daily meeting in the analysis, we discussed the elements of the menu, and we came out that it must be a third element, which is an explanation of the experiment.

Design Phase:

In the design phase, there were seven daily meetings. At the first one, the developers presented some suggestions about the form of the first screen, as illustrated in Figure 16.

The idea was that the options on the form are to be as a hyperlink in blue, but at the second daily meeting, the stakeholder opinion was that it is better to as a buttons, and the whole team agreed on this idea. At the third daily meeting, the developers presented the design of the last suggested main screen, after working at home for two hours, as illustrated in Figure 17.

Figure 15: Proposed prototype for the First screen of the App.

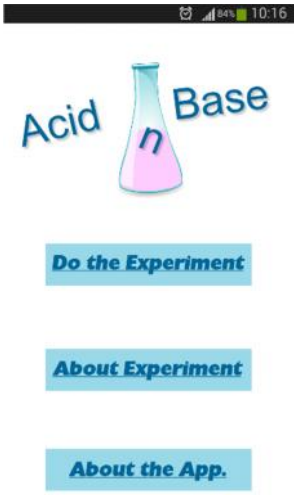


Figure 16: Suggested design for the First screen of the App.

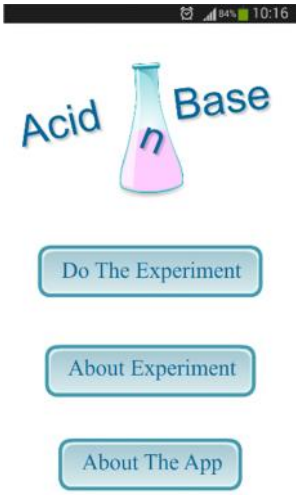


Figure 17: Suggested design for the First screen of the App. With buttons.

The fourth daily meeting ended so that the design is to be designed at the next day. The next two daily meetings were not effective because no one of the developers do anything. They were having other commitments. The chemistry expert and the stakeholder came to the meeting with nothing to do. The team worked at home with the design for 3 hours. At the seventh daily meeting, after the design is presented, the chemistry expert suggested to be a fourth button explaining the experiment with interactivity, not just a text. This idea was accepted by the whole team, and was suggested to be done with multimedia. Another button was added displaying a video explaining the experiment.

Development Phase:

As for the development phase of the first screen, we made first daily meeting discussing it. The developers were discussing about the coding of the buttons, but the Chemistry expert and the stakeholder was doing nothing with the developers. At this meeting, we have decided to have the

first screen programmed, the "About Experiment", and the "*About The App*" to be done next daily meeting. At the second daily meeting, we met to discuss the achievement after working 2 hours at home, but it was not fully accomplished. At the third and the fourth daily meeting, nothing done, because both developers were very busy and do nothing. At the fifth daily meeting, the developers displayed the first screen and the two buttons have been agreed to do, as illustrated in Figure 18, after working for 12 hours.

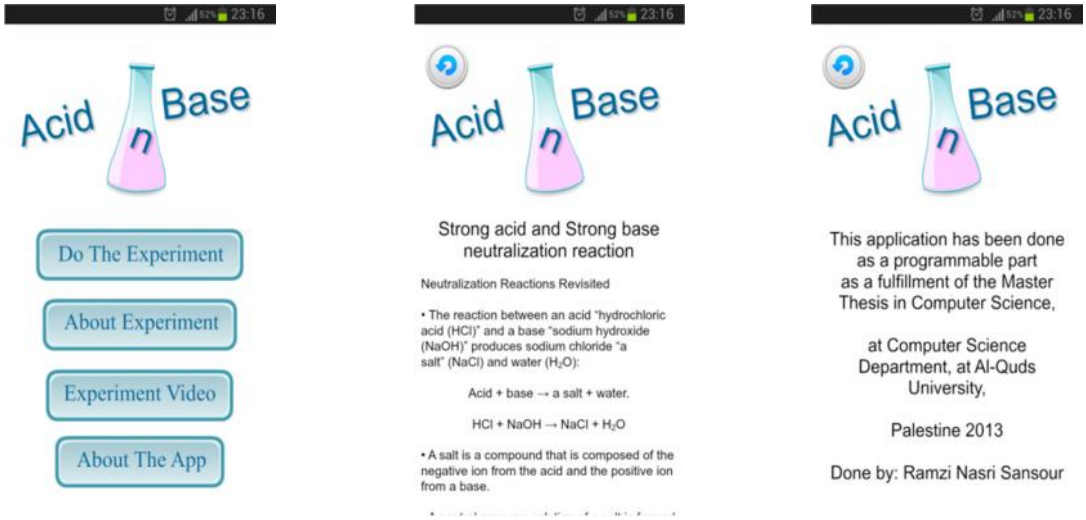


Figure 18: First screen of the App. and About Experiment and About The App.

At the sixth daily meeting, one of the developers displayed the video for the "Experiment Video". It was done as an mp4 file. It took 2 hours working at home. The Chemistry expert suggested that the video be a link to the YouTube, to let the user open other links for the experiment, as illustrated in Figure 19. The next daily meeting everything asked for have been done after working at home for 6 hours. At this stage, we have come to the end of the first screen development.

Testing Phase:

The next daily meeting was for testing the first screen. The whole team, except the stakeholder have met to test the first screen, but because he was not existed, we postponed the testing until the next day. By this, the first sprint comes to an end. At the end of the Sprint, we held a Sprint Review meeting with the whole team. In this stage, we applied the Traditional Scrum Methodology.

Figure 19: A video of the Experiment linked on the internet.

In this stage, we also followed the scrum methodology as it is. The whole team was in all daily meetings. Some of the meetings were not effective as the developers did nothing at home. Also the chemistry expert and the stakeholder in some meetings had nothing to do. Table 4 illustrates the time line of this stage, which is also illustrated in Figures 20 and 21.

Table 4: Stage 2 Time Line

Task Name	Duration	Start	Finish	Resource Names
STAGE 2 - First Screen (Main Menu)	**22 days (39 hrs.)**	**Sat 11/16/13**	**Sun 12/15/13**	
Analysis Phase	**4 days**	**Sat 11/16/13**	**Wed 11/20/13**	
First meeting: Discussion of applying the idea for the mobile.	15 mins	Sat 11/16/13	Sat 11/16/13	All Team
Second meeting: Making an initial prototype.	15 mins	Mon 11/18/13	Mon 11/18/13	All Team
At Home: Working with the first screen prototype.	6 hrs	Mon 11/18/13	Mon 11/18/13	Developers only
Third meeting: Put the sprint backlog	15 mins	Tue 11/19/13	Tue 11/19/13	All Team
Fourth meeting: Discussing the elements of the menu.	15 mins	Wed 11/20/13	Wed 11/20/13	All Team
Design Phase	**8 days**	**Thu 11/21/13**	**Mon 12/2/13**	
First meeting: Discussion about the form	15 mins	Thu 11/21/13	Thu 11/21/13	All Team
Second meeting: Taking the Stakeholder opinion about the screen selections.	15 mins	Sat 11/23/13	Sat 11/23/13	All Team
At Home: Working on the First Screen.	2 hrs	Mon 11/25/13	Mon 11/25/13	Developers only
Third meeting: Discussing the design of the main screen.	15 mins	Tue 11/26/13	Tue 11/26/13	All Team
Fourth meeting: Accepting the design.	15 mins	Wed 11/27/13	Wed 11/27/13	All Team
Fifth meeting: Nothing was done	15 mins	Thu 11/28/13	Thu 11/28/13	All Team
Sixth meeting: Nothing was done	15 mins	Sat 11/30/13	Sat 11/30/13	All Team
At Home: Working.	3 hrs	Sat 11/30/13	Sat 11/30/13	Developers only
Seventh meeting: A fourth button was added.	15 mins	Mon 12/2/13	Mon 12/2/13	All Team
Development Phase	**7 days**	**Tue 12/3/13**	**Wed 12/11/13**	
First meeting: Discussion about the coding.	15 mins	Tue 12/3/13	Tue 12/3/13	All Team
At Home: Developing.	2 hrs	Tue 12/3/13	Tue 12/3/13	Developers only
Second meeting: Discussing the achievement.	15 mins	Wed 12/4/13	Wed 12/4/13	All Team
Third meeting: Nothing was done.	15 mins	Thu 12/5/13	Thu 12/5/13	All Team
At Home: Developing.	12 hrs	Thu 12/5/13	Fri 12/6/13	Developers only
Fourth meeting: Discussing the First Screen.	15 mins	Sat 12/7/13	Sat 12/7/13	All Team
Fifth meeting: Nothing was done.	15 mins	Mon 12/9/13	Mon 12/9/13	All Team
At Home: Developing.	3 hrs	Mon 12/9/13	Mon 12/9/13	Developers only
Sixth meeting: Illustrating the video.	15 mins	Tue 12/10/13	Tue 12/10/13	All Team
At Home: Developing.	5 hrs	Tue 12/10/13	Tue 12/10/13	Developers only
Seventh meeting: Discussing the achievements.	15 mins	Wed 12/11/13	Wed 12/11/13	All Team
Testing Phase	**3 days**	**Thu 12/12/13**	**Sun 12/15/13**	
First meeting: Testing the work.	15 mins	Thu 12/12/13	Thu 12/12/13	All Team
Second meeting: Continue testing.	15 mins	Sat 12/14/13	Sat 12/14/13	All Team
Sprint Review meeting	1 hr	Sun 12/15/13	Sun 12/15/13	All Team

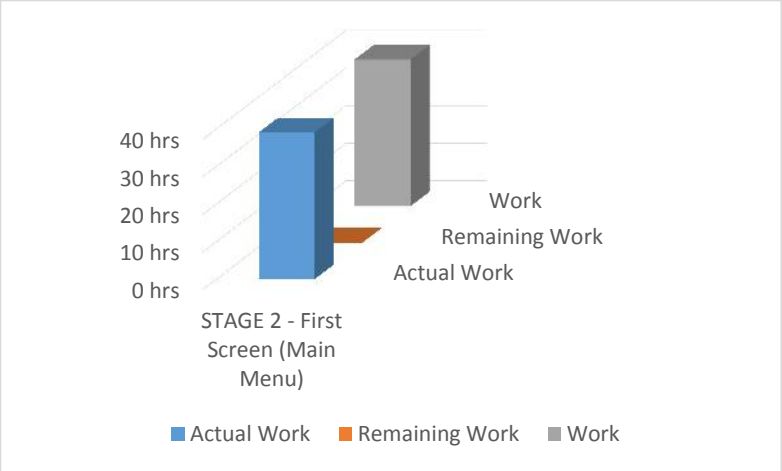


Figure 20: Stage 2 Work Chart

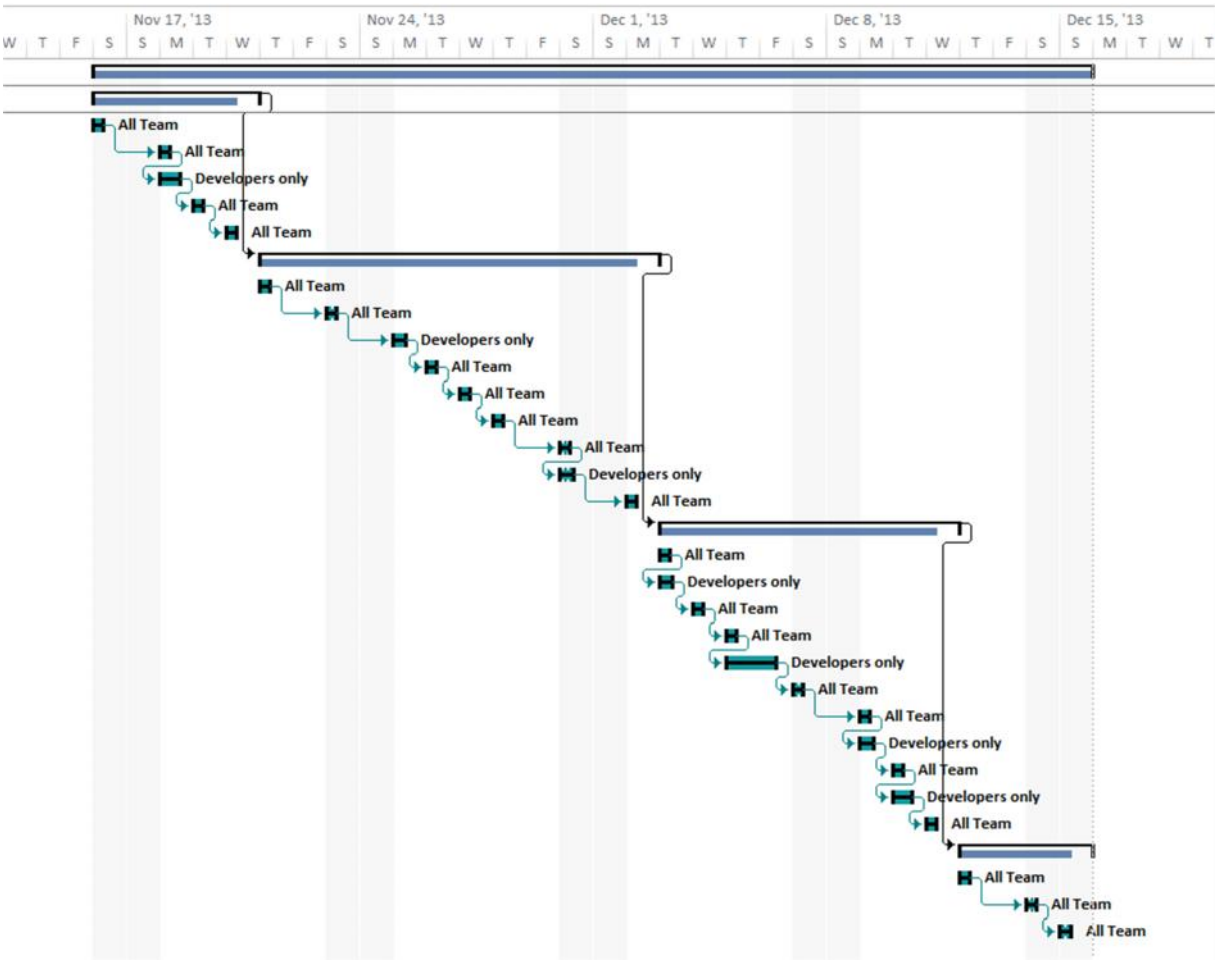


Figure 21: Stage 2 Gantt Chart

Stage 3: Second Screen (Base Selection)

Analysis Phase:

At the second stage, we had a meeting to analyze the next screen, the “Do the experiment” button and putting the sprint backlog. After three daily meetings, we came up with the functioning of the next screen.

Design Phase:

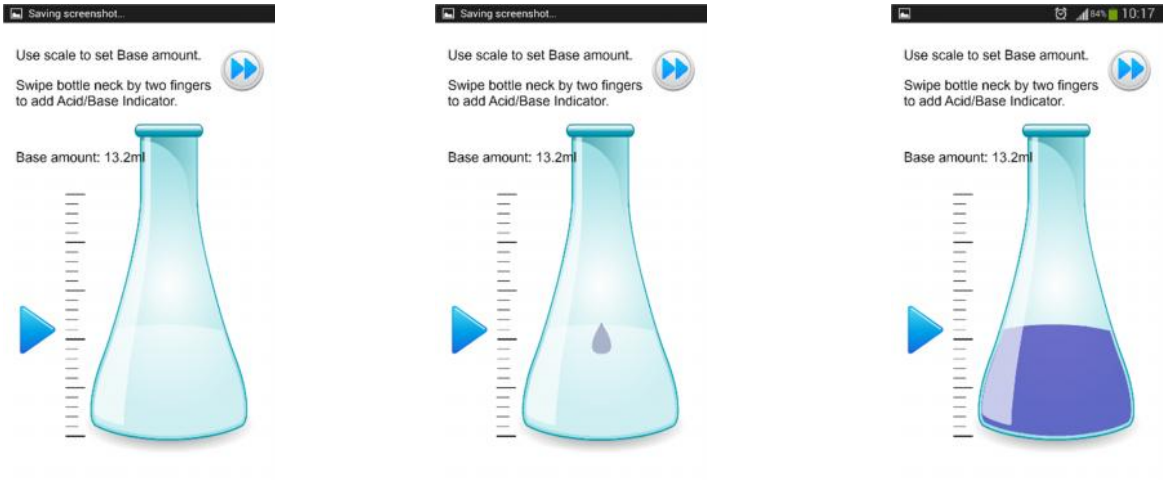
At the design phase, we met several daily meetings. In the first daily meeting, the whole team met and discussed the design of the next screen. Next, we put the initial design of the screen. The user clicks on the “Do the Experiment” button, which takes him to the Base amount determining. At first, the user puts the amount of the base, and then the user must add some drops of the Acid/Base indicator by pressing a button, to see its color, which is blue. This is illustrated in Figure 22. That was the opinion of the Chemistry expert, which is entering the base amount and pressing the button. However, the developers wanted to do something more impressive and use the most mobile functionalities and use more multimedia. Therefore, at the third daily meeting, after working at home for 6 hours, the developers preferred to use the touch functionality to select the solution of the *Base solution* by dragging the selector up and down to determine the amount of the solution. And as for adding the Acid/Base indicator, we preferred to use the MultiTouch functionality, by swapping the bottle neck by two fingers together to add four drops of the Acid/Base Indicator.

Figure 22: The initial design of determining the Base amount.

Therefore, the next daily meeting, we made an initial design for the screen showing our suggestion, after working at home for 2 hours.

Development Phase:

In the development phase, and because of the previous experience from what we have done in the first screen development, we applied the modified Scrum. We met and agreed that it was not necessary for the chemistry expert and the stakeholder to attend the daily meeting all the time, therefore we suggested that they might come at the end of this phase. The developers began coding; they worked for 8 hours daily, for each of them both, at home. They met three daily meetings. In the first and second daily meeting, they discussed plenty of things, but in the third, fourth and fifth day, they didn't meet, as they must continue coding at home. At the end of this phase, the whole team have met and discussed many things about what they have done. In the testing phase, the whole team had two daily meetings. They showed the second screen and tested it, as illustrated in Figure 23.



"a" ***"b"*** ***"c"***

Figure 23: "a" - Selector of the Base amount. "b" - Dropping the Acid/Base Indicator to appear the color of the Base. "c" – The Base after it is colored (Blue).

Testing Phase:

In the testing phase, the whole team met and tested the second screen. By this, the second sprint comes to an end. At the end of the Sprint, we held a Sprint Review meeting with the whole team.

In this stage, and in particular, in the development phase, we applied the modified scrum so as to reduce the time. In some meeting, not all the team meets. In some meeting the developers only have met. Also they didn't meet daily. Table 5 illustrates the time line of this stage, which is also illustrated in Figures 24 and 25.

Table 5: Stage 3 Time Line

Task Name	Duration	Start	Finish	Resource Names
STAGE 3 - Second Screen	**16 days (28 hrs.)**	**Mon 12/16/13**	**Sat 1/4/14**	
Analysis Phase	**3 days**	**Mon 12/16/13**	**Wed 12/18/13**	
First meeting: Discussion the second screen development.	15 mins	Mon 12/16/13	Mon 12/16/13	All Team
Second meeting: Coming up with the functioning of the second screen.	15 mins	Tue 12/17/13	Tue 12/17/13	All Team
Third meeting: Continuing what were done in the previous meeting.	15 mins	Wed 12/18/13	Wed 12/18/13	All Team
Design Phase	**3 days**	**Thu 12/19/13**	**Mon 12/23/13**	
First meeting: Discussion.	15 mins	Thu 12/19/13	Thu 12/19/13	All Team
Second meeting: Putting the initial design of the second screen.	15 mins	Sat 12/21/13	Sat 12/21/13	All Team
At Home: Working.	5 hrs	Sat 12/21/13	Sat 12/21/13	Developers only
Third meeting: Touch functionality suggestion.	15 mins	Sun 12/22/13	Sun 12/22/13	All Team
At Home: Developing.	3 hrs	Sun 12/22/13	Sun 12/22/13	Developers only
Fourth meeting: Presenting the initial design.	15 mins	Mon 12/23/13	Mon 12/23/13	All Team
Development Phase	**8 days**	**Tue 12/24/13**	**Thu 1/2/14**	
First meeting: Discussion between the developers only.	15 mins	Tue 12/24/13	Tue 12/24/13	All Team
At Home: Developing.	16 hrs	Thu 12/26/13	Fri 12/27/13	Developers only
Second meeting: Discussion.	15 mins	Sat 12/28/13	Sat 12/28/13	Developers & Product Owner
Third meeting: Discussion.	15 mins	Mon 12/30/13	Mon 12/30/13	Developers & Product Owner
Fourth meeting: Discussing achievements.	15 mins	Thu 1/2/14	Thu 1/2/14	Developers & Product Owner
Testing Phase	**1 day**	**Sat 1/4/14**	**Sat 1/4/14**	
First meeting: Testing the work.	15 mins	Sat 1/4/14	Sat 1/4/14	All Team
Sprint Review meeting	1 hr	Sat 1/4/14	Sat 1/4/14	All Team

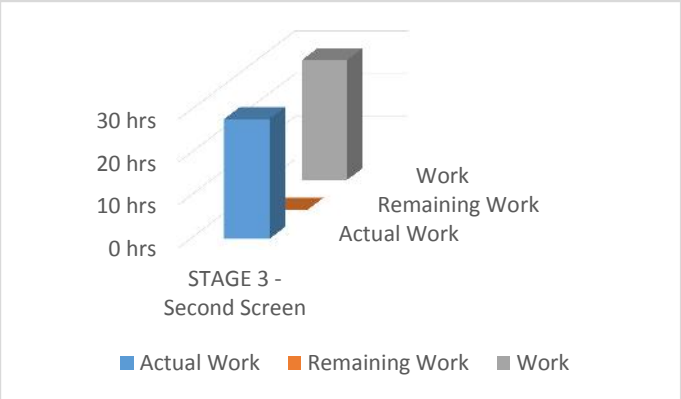


Figure 24: Stage 3 Work Chart

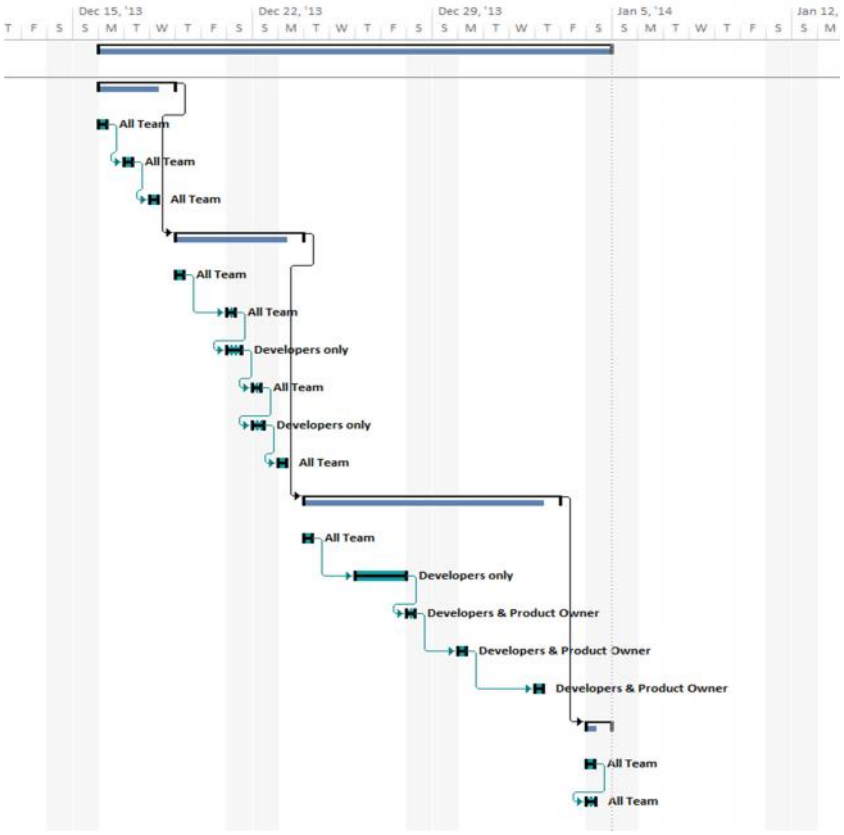


Figure 25: Stage 3 Gantt Chart

Stage 4: Third Screen (Acid Selection)

Analysis Phase:

At the third stage, we have a meeting to analyze the next screen, in which the user will select the Acid amount. The whole team met to analyze the work; they put the sprint backlog. In this screen, the user must select the amount of the Acid and adds the Acid/Base Indicator to color the solution.

Design Phase:

Because of the previous experience from the previous screen, we didn't meet daily in the design phase, but in the first day only.

Development Phase:

The developers worked for 6 hours for two days only and met one daily meeting within four days. The whole team have met and preview the work.

"a" ***"b"*** ***"c"***

Figure 26: "a" - Selector of the Acid amount. "b" - Dropping the Acid/Base Indicator to appear the color of the Acid. "c" – The Acid after it is colored (Red).

Testing Phase:

In the testing phase, the whole test the screen, as illustrated in Figure 26. By this, the third sprint came to an end.

In this stage we applied the modified scrum dramatically. Because of the previous experience from the previous screen, not all the team have met, and not daily. Table 6 illustrates the time line of this stage, which is also illustrated in Figures 27 and 28.

Table 6: Stage 4 Time Line

Task Name	Duration	Start	Finish	Resource Names
STAGE 4 - Third Screen	**8 days (13.25 hrs.)**	**Mon 1/6/14**	**Wed 1/15/14**	
Analysis Phase	**1 day**	**Mon 1/6/14**	**Mon 1/6/14**	
First meeting: Analyzing and putting the sprint backlog.	15 mins	Mon 1/6/14	Mon 1/6/14	All Team
Design Phase	**1 day**	**Tue 1/7/14**	**Tue 1/7/14**	
First meeting: Discussion.	15 mins	Tue 1/7/14	Tue 1/7/14	Developers & Product Owner
At Home: Working.	1 hr	Tue 1/7/14	Tue 1/7/14	Developers only
Development Phase	**4 days**	**Wed 1/8/14**	**Mon 1/13/14**	
First meeting: Discussion.	15 mins	Wed 1/8/14	Wed 1/8/14	Developers & Product Owner
At Home: Working.	10 hrs	Wed 1/8/14	Thu 1/9/14	Developers only
Second meeting: Reviewing the work.	15 mins	Mon 1/13/14	Mon 1/13/14	Developers & Product Owner
Testing Phase	**1 day**	**Wed 1/15/14**	**Wed 1/15/14**	
First meeting: Testing the work.	15 mins	Wed 1/15/14	Wed 1/15/14	All Team
Sprint Review meeting	1 hr	Wed 1/15/14	Wed 1/15/14	All Team

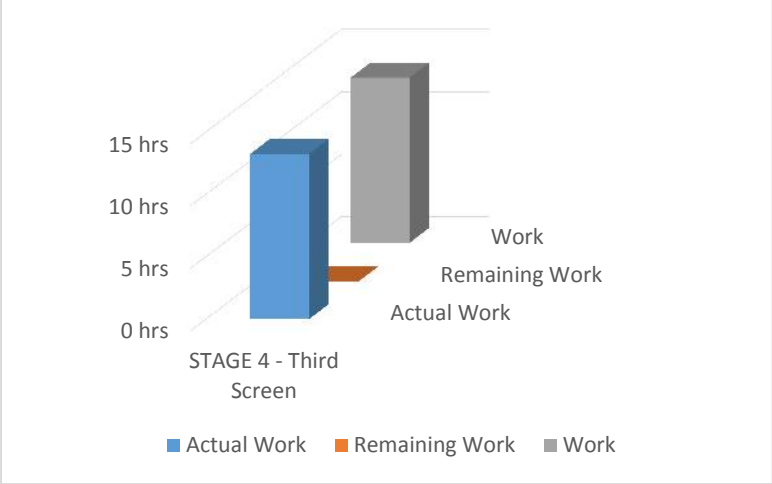


Figure 27: Stage 4 Work Chart

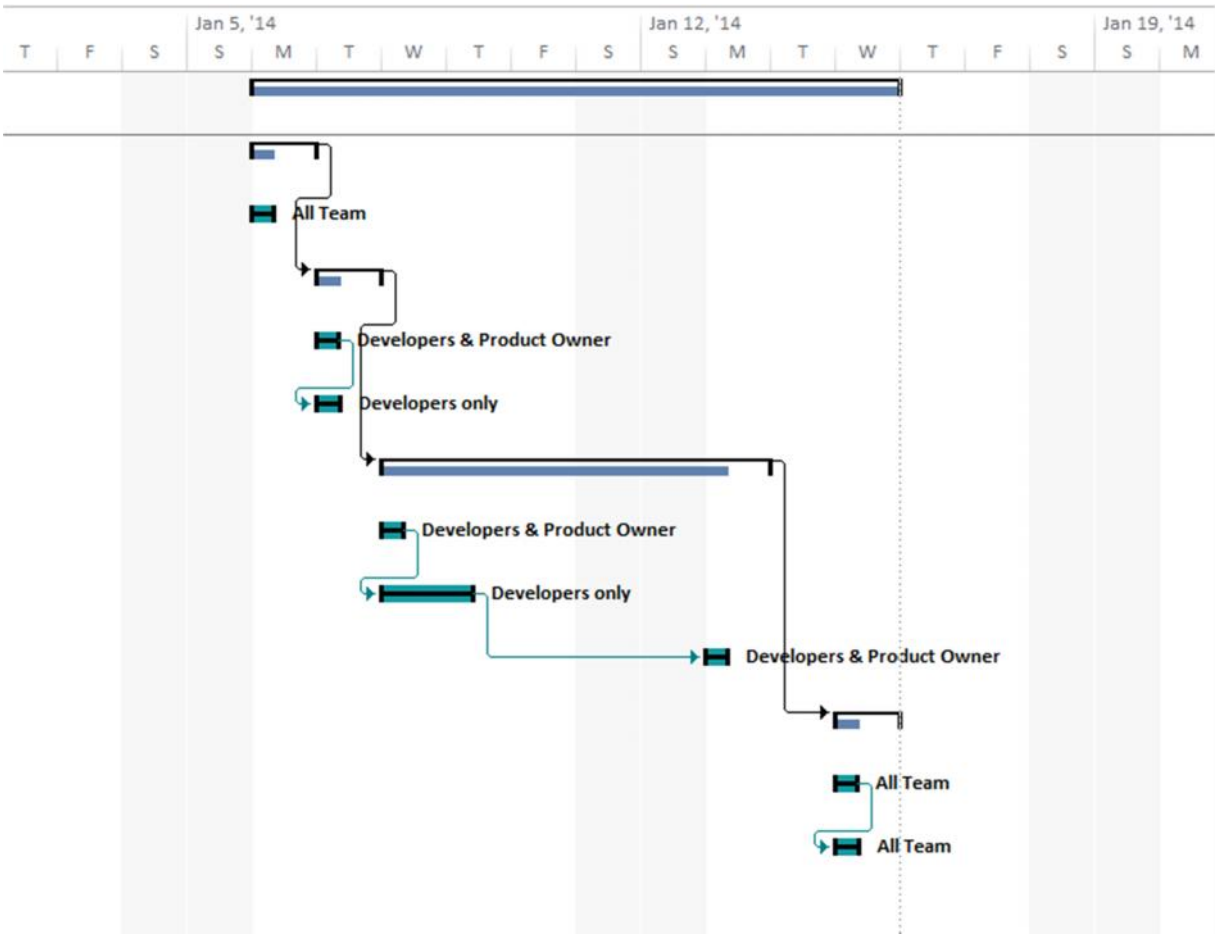


Figure 28: Stage 4 Gantt Chart

Stage 5: Last Screen (Mixing Solution)

Analysis Phase:

At the last stage, we have a meeting to analyze the last screen. We put the sprint backlog. The idea of the last screen was that when the user selects the amount of the Base and the Acid, he must mix the solution to have the result. After mixing the solution, the color will be either blue if the amount of the base is more than the acid, red if the amount of acid is more than the base, if equal solution (*negative ion from the acid and positive ion from a base*) the result is salt and water, which is colored with a green one. The whole team met for one daily meeting, because of the previous experience, as it was very similar to the previous screens, to analyze the work.

Design Phase:

In the design phase, the whole team met for one daily meeting, and was the expert opinion that for mixing the solution, the user must click on a button, as illustrated in Figure 29.

Figure 29: The suggested result for mixing the solution.

The next daily meeting was between the two developers only. In the third daily meeting, the whole team met and the developers' opinion was to use more multimedia and that was by using the mobile sensors, where the user must shake the mobile device to mix the solution to have the result. After shaking it hard enough, the color of the mixed solution appears. This is illustrated in Figure 30. The Stakeholder's opinion was that the process of shaking the device cannot be applied on all devices, as some devices can be old and has no gyroscope. However, as developers, we tried to convince him that the goal of this application is using the multimedia, and the sensors are very important to be used here, and this application is to be applied on the modern mobile devices "Smartphones", and the least specifications of the Smartphones, have sensors.

Figure 30: The mixed solution of Acid and Base when shaking the mobile device it appears Green if equal Acid and Base amount, otherwise Red if more Acid or Blue if more Base.

Development Phase:

In the development phase, the developers only met for one daily meeting, and because the similarity of the work, they worked at home for three days three hours a day, without any daily meetings. After that, they met another one daily meeting. At the end of this phase, the whole team met and discussed the work. The developers explained what they have done.

Testing Phase:

In the testing phase, the whole team met for two daily meetings, tested the whole application. After using sensors, in the application, the Stakeholder was convinced, and liked the idea very much. At the end of the Sprint, we held a Sprint Review meeting with the whole team.

After completing the application development, we noticed that the Scrum daily meetings were relatively too much and it were also boring for the Chemistry Expert as most of the time the Developers kept discussing the code without sharing him, and because the project is relatively

small. Also it is not that easy for the team to assemble every day, so it was enough to meet twice a week. The Scrum Master had the programming experience and he was one of the developers, where there is no need for a separate Scrum Master.

In these meetings, the update process is made, especially in the presence of the Chemistry Expert. The Stakeholder comes once a week, where he discusses the amendment with the developers. In this stage, we also applied the modified scrum dramatically. Table 7 illustrates the time line of this stage, which is also illustrated in Figures 31 and 32.

Table 7: Stage 5 Time Line

Task Name	Duration	Start	Finish	Resource Names
STAGE 5 - Last Screen	**15 days (21 hrs.)**	**Thu 1/16/14**	**Wed 2/5/14**	
Analysis Phase	**1 day**	**Thu 1/16/14**	**Thu 1/16/14**	
First meeting: Analyzing and putting the sprint backlog.	15 mins	Thu 1/16/14	Thu 1/16/14	All Team
Design Phase	**3 days**	**Sat 1/18/14**	**Tue 1/21/14**	
First meeting: Discussion.	15 mins	Sat 1/18/14	Sat 1/18/14	All Team
Second meeting: Discussion by developers only.	15 mins	Mon 1/20/14	Mon 1/20/14	Developers & Product Owner
Third meeting: Discussion.	15 mins	Tue 1/21/14	Tue 1/21/14	All Team
Development Phase	**8 days**	**Thu 1/23/14**	**Sat 2/1/14**	
First meeting: Discussion by developers only.	15 mins	Thu 1/23/14	Thu 1/23/14	Developers & Product Owner
At Home: Working.	6 hrs	Sat 1/25/14	Sat 1/25/14	Developers only
Second meeting: Discussion.	15 mins	Mon 1/27/14	Mon 1/27/14	Developers & Product Owner
At Home: Working.	10 hrs	Tue 1/28/14	Wed 1/29/14	Developers only
Third meeting: Reviewing the work.	15 mins	Sat 2/1/14	Sat 2/1/14	Developers & Product Owner
At Home: Working.	2 hrs	Sat 2/1/14	Sat 2/1/14	Developers only
Testing Phase	**2 days**	**Tue 2/4/14**	**Wed 2/5/14**	
First meeting: Testing the work.	15 mins	Tue 2/4/14	Tue 2/4/14	Developers & Product Owner
Sprint Review meeting	1 hr	Wed 2/5/14	Wed 2/5/14	All Team

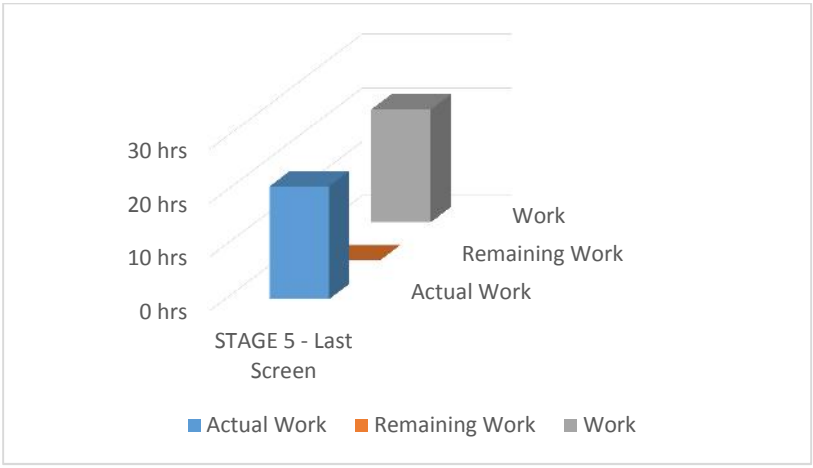


Figure 31: Stage 5 Work Chart

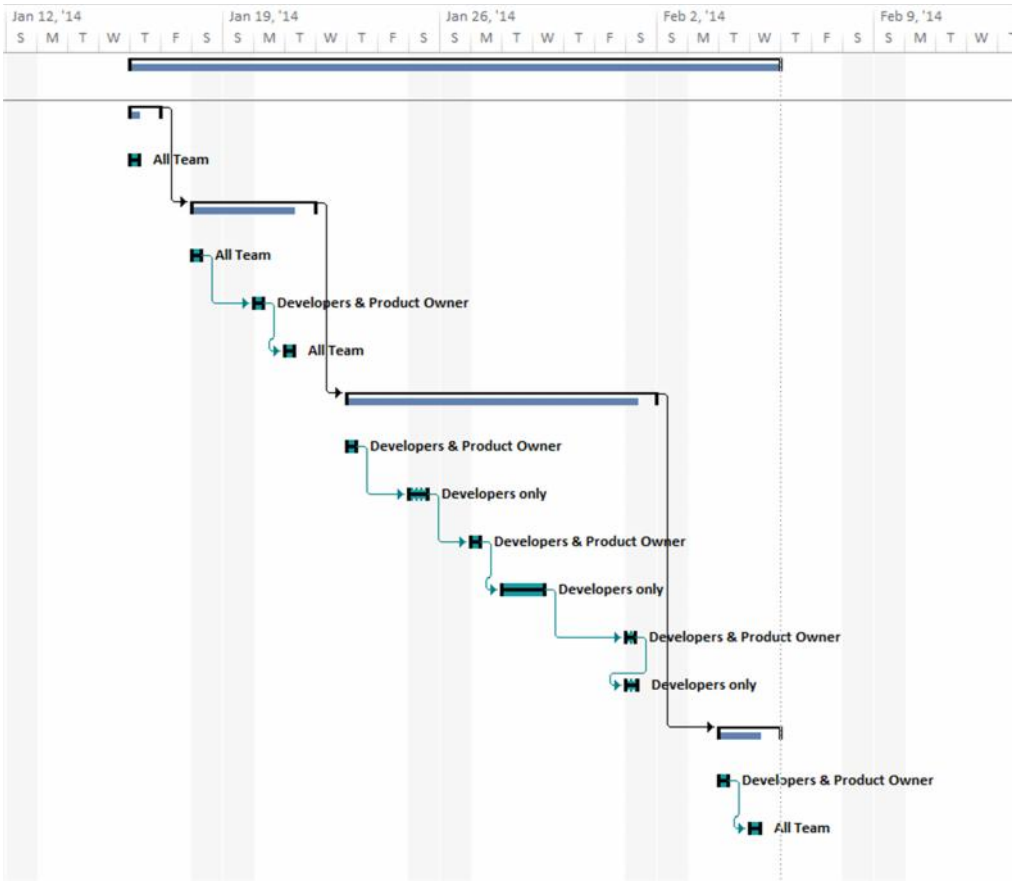


Figure 32: Stage 5 Gantt Chart

Finally, as you notice here, the time and effort in the modified scrum were less than in the traditional one.

Implementation Process

As for development improvement, we care about the feedback; we were having a brief period at the end of each sprint to know how we are doing and to find a way for improving the application. This was done by the sprint retrospective, where the entire team meets to discuss progress in developing the application and discussing the amendments and additions to the application, especially with the Stakeholder. This time requires one hour, and sometimes it took longer depending on the discussing subjects and the size of the updates.

In addition to the mentioned pitfalls, another overheads raised in each meeting with the Stakeholder. These overheads caused by new demands and requirements from stakeholders asking for additional improvements and modification, which were not mentioned at the beginning. This because the multimedia has a lot of features and aesthetics. Consequently, we had to do multiple updates and refinements on the applications.

The application was implemented on one platform and tested at the end of each sprint. Since there are different platforms issued by deferent manufacturers and vendors, all the added increments (“done”) must be tested on all other different platforms and devices with different capabilities.

At the end of the Scrum developing process, we had a ready application developed and tested on Android, later on; we deployed it on iOS, without any modification to the original code, later on we will try to run it on other platforms (Windows Phone and Blackberry). We have run the applications on devices with different technologies and different capabilities to test the efficiency of the application. Table 8 illustrates the devices on which the application was tested. By running the application on these devices, the performance was as good as a Native application. All functionalities worked efficiently on all different devices with its different capabilities.

Table 8: Devices that our application tested on.

Device	OS	Specifications
Samsung Galaxy S3	Android 4.1.2 Android 4.3	CPU: Quad-core 1.4 GHz Cortex-A9 RAM: 1 GB
Samsung Galaxy Note	Android 4.0.4	CPU: Dual-core 1.5 GHz Scorpion RAM: 1 GB
Samsung Galaxy Y	Android 2.3.5	CPU: 830 MHz ARMv6 RAM: 290 MB
LG Optimus L3 E400	Android 2.3.6	CPU: 800 MHz RAM: 384 MB
iPhone 4	iOS 4 iOS 5	CPU: 1 GHz Cortex-A8 RAM: 512 MB

The Experiment – Description

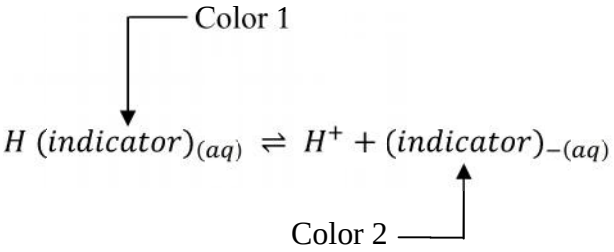
The Chemical experiment *"Strong acid and Strong base neutralization reaction":*

- The reaction between an acid "hydrochloric acid (HCl)" and a base solution "sodium hydroxide (NaOH)" produces sodium chloride "a salt" (NaCl) and water (H_2O):

$$HCl + NaOH \rightarrow NaCl + H_2O$$

- A salt is a compound that is composed of the negative ion from the acid and the positive ion from a base.
- A neutral aqueous solution of a salt is formed by the reaction of any strong acid with any strong base in the mole ratio from the balanced chemical equation.
- Reaction of acids and bases of different strengths usually do not result in neutral solutions.
- The neutralization reaction occurring has no visible signs, unless using an acid-base indicator (a weak acid or a weak base).

  With the indicator it appears as one color:



This is a substance that changes color in acidic and basic solution.

Table 9 illustrates the time in Traditional Scrum methodology and in suggested one.

Table 9: The time in Traditional Scrum and in modified one.

STAGES		TRADITIONAL SCRUM		MODIFIED SCRUM	
		Daily meetings	Working hours at home	Daily meetings	Working hours at home
Application idea		11	5		
First Screen (Main Menu)	**Analysis**	4	6		
	Design	7	5		
	Development	8	22		
	Testing	2	0		
	Sprint Review meeting	1	0		
Total		**22**	**33**		
Second Screen (Base selection)	**Analysis**	3	0		
	Design	4	8		
	Development			4	16
	Testing	1	0		
	Sprint Review meeting	1	0		
Total		**9**	**8**	**4**	**16**
Third Screen (Acid Selection)	**Analysis**			1	0
	Design			1	1
	Development			2	10
	Testing	1	0		
	Sprint Review meeting	1	0		
Total		**2**	**0**	**4**	**11**
Last Screen (Mixing Solution)	**Analysis**			1	0
	Design			3	0
	Development			3	18
	Testing	1	0		
	Sprint Review meeting	1	0		
Total		**2**	**0**	**7**	**18**
Total for all stages		**46**	**46**	**15**	**45**

As we see from the results in the table above, between stage 3 and stage 4, that the modified Scrum has shortened the time, as the number of daily meetings in stage 4 was 6, while in stage 3 was 13, that is, the ratio is about 40%. From this, we can say that the modified Scrum shortened the time to about one third of the original Scrum.

Suggested modifications

We suggest modifications on some of these pitfalls. Table 10 illustrates the pitfalls and its modifications.

Table 10: Modifications of some of Scrum pitfalls.

Scrum pitfall	Suggested modification
1. Difficulty of testing.	In Scrum, the testing process is based on creating the increment after reviewing the work at the end of each sprint to see if it accomplished the requirements, later all increments will be added together and tested thoroughly. In our approach because of mobile multi vendors (multi technologies), it is difficult to test the code on different capability devices at the end of each sprint, so it should only be taken into consideration the capabilities of the different devices in coding, tested on one platform, later at the end, all the added increments must be tested on all other platforms with different capabilities.
2. Daily Scrum Meeting.	As mobile projects considered relatively small, the Multimedia Engineers and the Subject Matter Expert in Educational Materials will not be able to follow up with the technological meetings. Therefore, the Daily Scrum Meetings must be reduced.
3. Stakeholders demanding.	Since multimedia has a lot of features and aesthetics, in terms of picture, sound and video, in particular mobile properties, this opens a way for requesting many changes and additions for mobile applications constantly, leading to a never-ending period because of infinite requests by the stakeholders. Another thing is the fluctuations in desires of the end users. Therefore, there must be a written agreement with the application requirements and end date must be specified.
4. Team members' commitments.	Because mobile projects considered small, the tasks for each member will be small, so if any one of the team members left before the end of the project, we can entrust the task to someone else of the team members to continue.

Other modifications

- Reducing the number of developers, and the Scrum Master and Product Owner can be one person.
- Shortening the Sprint because mobile projects considered small and low price, as lengthen the time increases the cost, which leads to a reduction of profit.

Finally, our suggestions and modifications can be shown in Figure 33, which illustrate our suggested iterative/incremental object-oriented development cycle for Scrum.

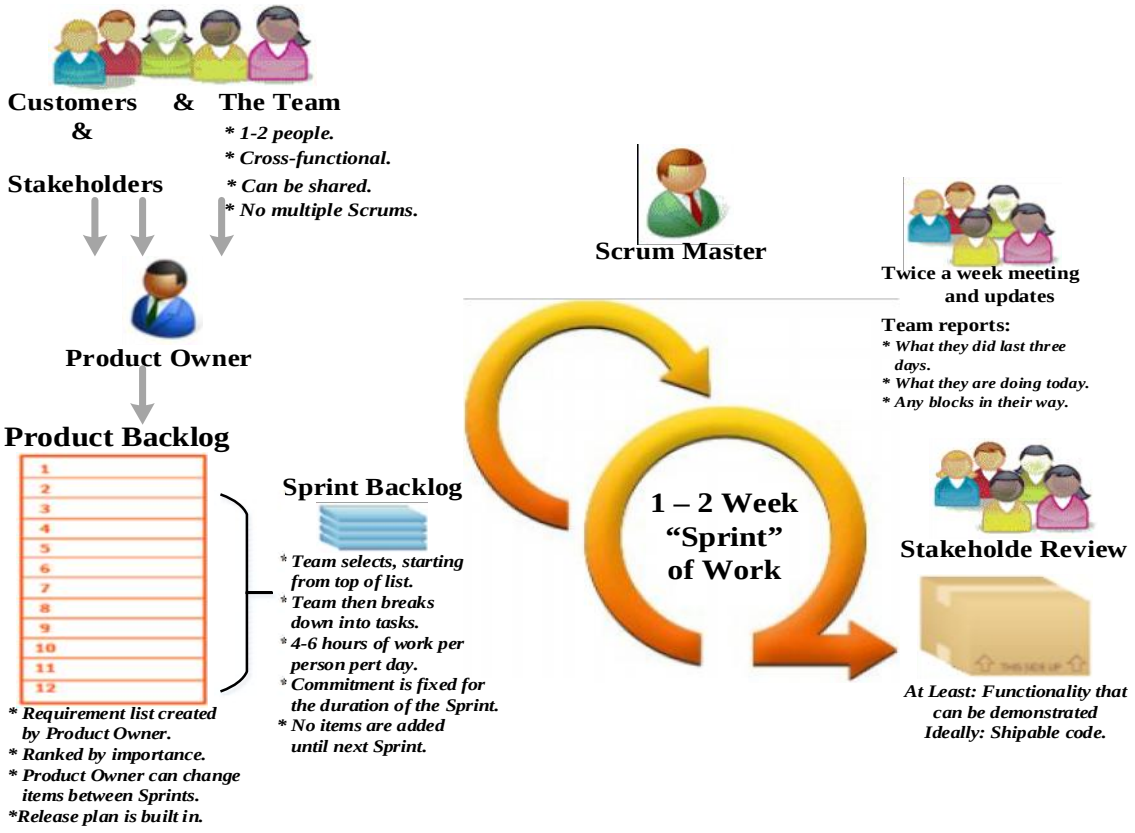


Figure 33: Suggested SCRUM process flow for mobile small applications

Figure 34 illustrates our suggested Twice-a-Week Scrum.

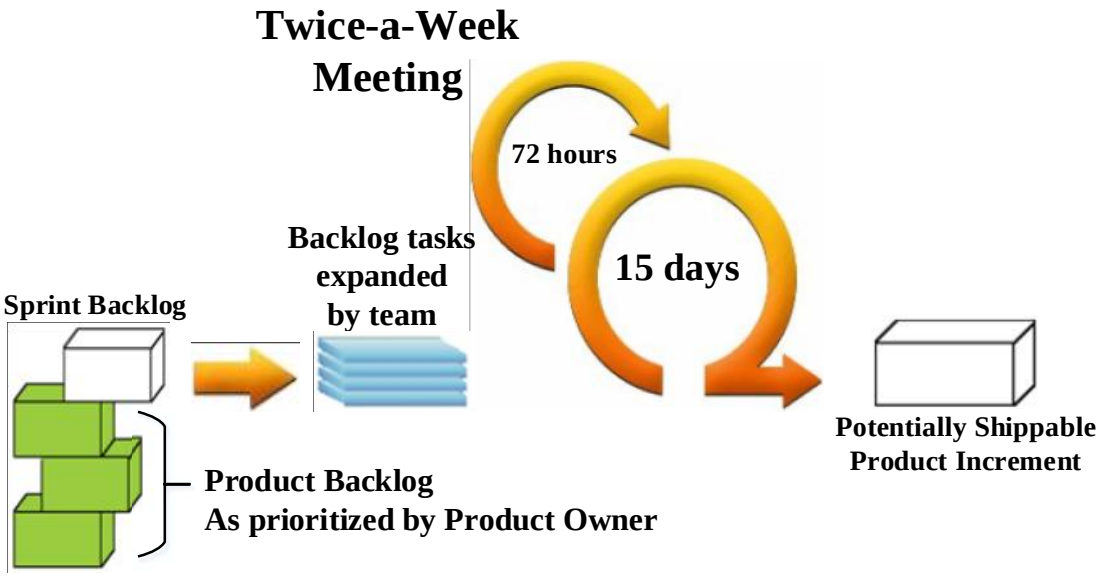


Figure 34: Suggested Twice-a-Week Scrum

Chapter Six

Conclusions and Future Works

Conclusions

In this thesis, we tried to follow the best software engineering methodology for mobile application development. The main goal for this thesis was to find a way to speed up the developing process of applications for mobile devices (handheld devices, smartphones, tablets, etc.) to save time and effort. We developed a small multimedia mobile application, replacing traditional developing methodology with an Agile Practice, using one of its methodologies, which is the Scrum. We put some suggestions and modifications to improve the Scrum methodology by reducing some of its steps an time. We differentiated between Native, Web and Hybrid mobile applications. The mobile software engineering is discussed in detail comparing some related researches. Also, the Scrum methodology is discussed in general concentrating on the multimedia mobile software development with Scrum. The developed application was tested from several aspects on different mobile platforms. While applying it for "*Small Multimedia Mobile Applications Development*", our modifications on Scrum were:

- Twice-A-Week Meetings (Dyamic) instead of *Daily Scrum Meetings.*
- There must be a written agreement with the application requirements and the delivery date must be specified.
- Entrust the tasks to all the team members to continue, for the project not to fail.
- Reducing the number of developers.
- Shortening the Sprint because mobile projects are considered small.

- It should be taken into consideration the capabilities of the different devices in coding, testing on one platform, later at the end, all the added increments must be tested on all other platforms with different capabilities.

Future Works

It should only be taken into consideration the capabilities of the different devices in coding, testing on one platform, later at the end, all the added increments must be tested on all other platforms with different capabilities.

References

ADELL, L. (2013, April 11). *BENEFITS & PITFALLS OF USING SCRUM SOFTWARE DEVELOPMENT METHODOLOGY*. Retrieved from http://blog.belatrixsf.com/benefits-pitfalls-of-using-scrum-software-development-methodology/

Adobe. (2012). Adobe Flash Professional CS6 What's New – Create and deliver rich, engaging experiences across devices. *© 2012 Adobe Systems Incorporated. All rights reserved. 3/12*.

Ali, N., & Ramos, I. (2012). Designing Mobile Aspect-Oriented Software Architectures with Ambients. *Handbook of Research on Mobile Software Engineering, Edit. Paulo Alencar, Donald Cowan IGI global, 10.4018/978-1-61520-655-1, ch029*, 526-543.

AppBrain. (2014, February 13). *Most popular Android market categories.* (AppBrain Stats) Retrieved from http://www.appbrain.com/stats/android-market-app-categories

Avram, A. (2013, Feb. 21). *Xamarin 2.0 Brings a New IDE, Visual Studio Add-in for iOS and a Component Store.* Retrieved from http://www.infoq.com/news/2013/02/Xamarin-2;jsessionid=AA70FD796EBA5F1DF7DA9CCC281A59A8

Bareiss, R., & Sedano, T. (2011). Improving Mobile Application Development. *2nd Annual Workshop on Software Engineering for Mobile Application Development, MobiCASE'11 Santa Monica, CA, USA*, 7-10.

BayTechServices. (2013). *Native, Hybrid or Mobile Web - What's Best for your Mobile Apps?* 45 Coventry Lane - North Andover, MA 01845, www.BayTechServices.com: BayTech Services.

Biegel, G., & Cahill, V. (2004). A framework for developing mobile, context-aware applications. *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications*, 361-365.

Biztech2.com. (2013, August 14). *Smartphone Sales Grew 46.5% In 2Q13: Gartner*. (Biztech2.com Staff) Retrieved from http://www.firstbiz.com/biztech/smartphone-sales-grew-46-5-in-2q13-gartner-17920.html

Bora, K. (2013, November 15). (INTERNATIONAL BUSINESS TIMES, © Copyright 2014 IBT Media Inc. All Rights Reserved) Retrieved from http://www.ibtimes.com/worldwide-smartphone-sales-q3-2013-samsung-android-maintain-lead-apples-market-share-slips-1471754

Buthpitiya, S., Cheng, H., Sun, F., Griss, M., & Dey, A. (2010). Hermes: A Context-Aware Application Development Framework for the Mobile Environment. *Carnegie Mellon University 2010 poster*.

Carbon, R., & Hess, S. (2011). Mobile Business Applications must be thoroughly engineered. *2nd Annual Workshop on Software Engineering for Mobile Application Development, MobiCASE'11 Santa Monica, CA, USA*, 14-16.

Charland, A., & Leroux, B. (2011). Mobile Application Development: Web vs. Native. *Communications of the ACM 54*, 49-53.

Cheng, M., & Yuan, S. (2007). An Adaptive and Unified Mobile Application Development Framework for Java*. *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 23*, 1391-1405.

Corona. (2014). *Corona SDK – The World #1 Mobile Development Platform.* http://www.coronalabs.com/products/corona-sdk: Copyright © 2009-2014 Corona Labs Inc. A mobile development software company.

Corral, L., Sillitti, A., & Succi, G. (2013). Software Development Processes for Mobile Systems. *1st International Workshop on Mobile-Enabled Systems (MOBS 2013) in connection with ICSE 2013*, 19-24.

Dehlinger, J., & Dixon, J. (October 2011). Mobile Application Software Engineering: Challenges and Research Directions. *Workshop Papers*.

Dorokhova, R., & Amelichev, N. (2010). Comparison of Modern Mobile Platforms from the Developer Standpoint. *8TH CONFERENCE OF FINNISH-RUSSIAN UNIVERSITY COOPERATION IN TELECOMMUNICATIONS,*

8TH CONFERENCE OF FINNISH-RUSSIAN UNIVERSITY COOPERATION IN TELECOMMUNICATIONS, pp. 36-45.

Dua, K. (2013, September 27). *BlackBerry World now contains over 130,000 BB10 apps*. (© Copyright NDTV Convergence Limited 2014. All rights reserved) Retrieved from http://gadgets.ndtv.com/mobiles/news/blackberry-world-now-contains-over-130000-bb-10-apps-424613

Ferdiana, R. (December-2012). Agile Software Engineering Framework for Evaluating Mobile Application Development. *International Journal of Scientific & Engineering Research - ISSN 2229-5518, 3*(12).

Flora, H., & Chande, S. (2013). A REVIEW AND ANAYSIS ON MOBILE APPLICATION DEVELOPMENT PROCESSES USING AGILE METHODLOGIES. *International Journal of Research in Computer Science eISSN 2249-8265, 3*(4), 9-18.

Gartner. (August, 2012, August 14). (© 2012 Gartner, Inc. and/or its Affiliates. All Rights Reserved) Retrieved from http://www.gartner.com/newsroom/id/2120015

Gartner. (February, 2013, February 13). (© 2013 Gartner, Inc. and/or its Affiliates. All Rights Reserved) Retrieved from http://www.gartner.com/newsroom/id/2335616

Gartner. (May, 2012, May 16). (© 2012 Gartner, Inc. and/or its Affiliates. All Rights Reserved) Retrieved from http://www.gartner.com/newsroom/id/2017015

Gartner. (November, 2012, November 14). (© 2012 Gartner, Inc. and/or its Affiliates. All Rights Reserved) Retrieved from http://www.gartner.com/newsroom/id/2237315

Ha, K., Pillai, P., Lewis, G., Simanta, S., Clinch, S., Davies, N., & Satyanarayanan, M. (October 2012). The Impact of Mobile Multimedia Applications on Data Center Consolidation. *Technical Report CMU-CS-12-143, CMU School of Computer Science*.

Habermas, S. (July 29, 2013). *Are Agile development & mobile apps the perfect match?* Retrieved from http://www.verivo.com/mobility-watch/are-agile-development-mobile-apps-the-perfect-match/

Habib, M. (2013, December 30). *Agile software development methodologies and how to apply them*. Retrieved from http://www.codeproject.com/Articles/604417/Agile-software-development-methodologies-and-how-t

Hartmann, G., Stead, G., & DeGani, A. (March 2011). Cross-platform mobile development. *Mobile Learning Environment, Cambridge*.

Holler, R. (2006). *Mobile Application Development: A Natural Fit with Agile Methodologies.* Copyright © 2006 VersionOne, LLC.

IBM. (April 2012). *Native, web or hybrid - mobile-app development.* Produced in the United States of America: © Copyright IBM Corporation.

Intel IT. (August 2012). Building a Mobile Application Development Framework. *IT@Intel White Paper, IT Best Practice, Cloud Computing and Compute Continuum*.

Ipek, B. (2012). *Apple IOS Game Development Engines.* Istanbul, SWE578: Software Engineering Department, Bogazici University.

ITSpree. (2011). *Process and Quality Assurance.* © 2010 - 2011 ITSpree sp. z o.o. All Rights Reserved, http://www.itspree.pl/why-itspree/process-and-quality-assurance.html.

Javagal, Rantakala, Rajeshwari, & Nandita, a. (Nov. 17-18, 2012). Software Development Using Agile & Scrum. *International Conference on Advances in Computer and Electrical Engineering (ICACEE'2012) Manila (Philippines)*, 82-85.

jQueryMobile. (2014). *jQuery Mobile: Touch-Optimized Web Framework for Smartphones & Tablets, Copyright 2014 The jQuery Foundation.* Retrieved from http://jquerymobile.com

Kim, H. (2012). Frameworks for Validation of Mobile Software Project Performance. *Proceedings of the World Congress on Engineering and Computer Science, 1*, 24-26.

Lima, I., de Castro Freire, T., & Costa, H. (2012). Adapting and Using Scrum in a Software Research and Development Laboratory. *Revista de Sistemas de Informação da FSMA*, *9*, 16-23.

Mahalakshmi, M., & Sundararajan, M. (1 Feb. 2014). An Analysis on Scrum Methodology Used For the IT Project for Effective Software Deliverable. *International Journal of Scientific Engineering and Technology, 3*(2), 121-123.

Microsoft. (2011). *Building Mobile Applications for Java Developers.* Microsoft Corporation.

Mimbo. (2014, March 3). *iOS development news and information for the community, by the community*. (Copyright © 2014 Steel Media Ventures • A service of 148Apps • Theme originally by Mimbo) Retrieved from http://148apps.biz/app-store-metrics/?mpage=catcount

MyBroadband.co.za. (2013, February 13). *Worldwide Mobile Phone Sales Declined 1.7 Per Cent in 2012*. (Copyright © 2013 MyBroadband.co.za News, All rights reserved) Retrieved from http://companies.mybroadband.co.za/blog/2013/02/13/worldwide-mobile-phone-sales-declined-1-7-per-cent-in-2012

Ojala, T., Korhonen, J., Aittola, M., Ollila, M., Koivumäki, T., Tähtinen, J., & Karjaluoto, H. (December, 2003). Smartrotuaari-context-aware mobile multimedia services. *In 2nd International Conference on Mobile and Ubiquitous Multimedia, Norrkoping, Sweden*.

Oracle. (2011). *Develop Mobile Applications With Oracle ADF Mobile.* Oracle Data Sheet - Oracle and/or its affiliates.

Rao, A. (2010, April 7). *Adapting Agile Methodology in a Start-up*. Retrieved from http://pmtips.net/adapting-agile-methodology-startup/

Ribeiro, A., & da Silva, A. (2012). Survey on Cross-Platforms and Languages for Mobile Apps. *Eighth International Conference on the Quality of Information and Communications Technology, Instituto Superior Técnico Lisbon, Portugal, IEEE DOI 10.1109 / QUATIC, 56*, 255-260.

Rosa, P., Dias, J., Lopes, I., Rodrigues, J., & Lin, K. (2012). An Ubiquitous Mobile Multimedia System for Events Agenda. *IEEE Wireless Communications and Networking Conference: Mobile and Wireless Networks.*

Sansour, R. N., Kafri, N., & Sabha, M. N. (2014, April 14). A survey on mobile multimedia application development frameworks. *Multimedia Computing and Systems (ICMCS), 2014 International Conference* (pp. 967 - 972). Marrakech: IEEE.

Schwaber, K., & Sutherland, J. (July 2013). *The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game.* © 1991-2013 Ken Schwaber and Jeff Sutherland, All Rights Reserved.

Sencha. (2014). *Sencha Touch - Build Mobile Web Apps with HTML5, 2014 Sencha Inc. All rights reserved.* Retrieved from http://www.sencha.com/products/touch

Sierra. (2011). *Framework Selection For Mobile Enterprise Applications.* Sierra Systems Inc.

Simon, R., & Fröhlich, P. (May 8-12, 2007). A Mobile Application Framework for the Geospatial Web. *WWW 2007 / Track: Pervasive Web and Mobility - Banff, Alberta, Canada*, 381-390.

Singh, I., & Palmieri, M. (2011). COMPARISON OF CROSS-PLATFORM MOBILE DEVELOPMENT TOOLS. *IDT: Malardalen University*.

Spiridon, A. (2012, December 10). *Using the NativeUI Map Widget.* Retrieved from http://www.mosync.com/documentation/manualpages/using-nativeui-map-widget

Statista. (2014). *Global smartphone sales Q1 2009-Q3 2013, by operating system.* (The Statistics Portal, © Statista 2014) Retrieved from

http://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operating-system

Su, H. (2011). Let's Grow in Scrum Practices: Some Management Lessons Learned from a Global Software Development Project. *3rd International Conference on Advanced Management Science IPEDR, IACSIT Press, Singapore, 19*, 223-227.

Szántó, K. (Fall 2010). Extending the Java Context Awareness Framework for Android. *IT University of Copenhagen, In the Pervasive Computing Course*.

Szkopiński, T. (2012, Nov. 30). *Scrum methodology FAQ for Indian professionals*. Retrieved from http://searchbusinessintelligence.techtarget.in/photostory/2240173570/Scrum-methodology-FAQ-for-Indian-professionals/6/Scrum-disadvantages

Uhlig, D. (2014). *Advantages and Disadvantages of the Scrum Project Management Methodology*. Retrieved from http://smallbusiness.chron.com/advantages-disadvantages-scrum-project-management-methodology-36099.html

Wagner, R. (2011). *Professional Flash® Mobile Development Creating Android™ and iPhone® Applications.* Indianapolis, Indiana: Copyright © 2011 by Wiley Publishing, Inc.

Walkley, M., Ramsay, M., & Sinha, S. (2013). *Q2/13 HANDSET MARKET SURVEY & JULY SURVEY: SAMSUNG AND CHINESE OEMS GROW H1/13 SMARTPHONE MARKET SHARE.* CANACCORD Genuity.

Wasserman, A. (2010). Software engineering issues for mobile application development. *in Proceedings of the FSE/SDP workshop on Future of software engineering research - FoSER '10,*, 397-400.

Wikipedia. (2015, Jan 5). *Comparison of mobile operating systems*. Retrieved from http://en.wikipedia.org/wiki/Comparison_of_mobile_operating_systems#cite_note-2

WindowsPhoneAppList. (2014, March 8). *Windows Phone App List*. (© 2011 WindowsPhoneAppList.com - Windows Phone 7 applications) Retrieved from http://www.windowsphoneapplist.com/en/downloadRank/

Winkler, F., & Barrett, C. (Fall 2011). *Intro to Unity3D (vers. 3.4).* Intro to Unity3D workshop.

Xamarin. (2014). *Xamarin - Create iOS, Android and Mac apps in C#, - © 2014 Xamarin Inc.* Retrieved from http://xamarin.com

Xamarin, I. (2012). *Part 1 - Understanding the Xamarin Mobile Platform.* Retrieved from http://xamarin.alley.ws/guides/cross-platform/application_fundamentals/building_cross_platform_applications/part_1_-_understanding_the_xamarin_mobile_platform

Zhang, J., Wu, D., Ci, S., Wang, H., & Katsaggelos, A. (2009). Power-aware mobile multimedia: a survey. *Journal of Communications, 4*(9), 600-613.