جامعة بيرزيت
BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Master of Software Engineering

Exploring Mobile Applications Issues and Advantages When Adopting Agile Methods: A Case Study

دراسة حالة بناء برامج المحمول ضمن البيئة التراكمية من حيث المزايا و العيوب

By:

Ali Asfour

Supervisor:

Dr. Samer Zain

July, 2018

Faculty of Engineering and Technology

Master of Software Engineering

Exploring Mobile Applications Issues and Advantages When Adopting Agile Methods: A Case Study

دراسة حالة بناء برامج المحمول ضمن البيئة التراكمية من حيث
المزايا و العيوب

By:

Ali Asfour

This thesis was submitted in partial fulfillment of the requirements for the Master's Degree in software engineering from the Faculty of Graduate Studies, at Birzeit University, Palestine

July, 2018

Exploring Mobile Applications Issues and Advantages When Adopting Agile Methods: A Case Study

دراسة حالة بناء برامج المحمول ضمن البيئة التراكمية من حيث المزايا و العيوب

By:

Ali Asfour

This thesis was prepared under the supervision of Dr. Samer Zain and has been approved by all members of examination committee:

Dr. Samer Zain
(Chairman of the Committee)
____

Dr. Abdel Salam Sayyad
(Member)
____

Dr. Adel Taweel
(Member)
____

Date of Defense:

June 7, 2018

Abstract

Agile development methods have been proposed as a natural fit for mobile app development contexts. However, mobile apps have their own peculiarities that distinguish them from traditional web and desktop applications. For instance, platform and hardware fragmentation, life-cycle conformance, screen sizes, and tight time-to-market are just a few constraints that are particularly associated with mobile app development contexts.

Accordingly, agile methods have to be tailored to better suit mobile app development in real-world industrial contexts. Despite many studies addressing the adoption of agile methods for traditional web and desktop development, there still remains a lack of studies of how mobile app development teams adopt agile methods and the challenges they are facing.

This study capitalizes in the direction of exploring and understanding how industrial teams approach agile mobile app development, and the challenges they are facing. A qualitative study is conducted involving four different mobile app development companies. This study argues that not all agile development principals are applicable within mobile app context. Further, mobile app development teams face additional challenges when adopting agile methods such as development automation tools and on-line app stores' restrictions. Accordingly, this study proposed a new agile-scrum method to address various challenges involved in mobile application development.

الملخص

تعتبر البيئات التراكمية أو ما يعرف بأساليب التطوير الرشيقة الأكثر ملائمة لتطوير تطبيقات المحمول، إلا أن الخصائص التي تميز هذه التطبيقات عن غيرها من تطبيقات سطح المكتب والويب التقليدية، كالقيود الخاصة بتطوير تطبيقات المحمول مثل التنوع في المنصات والمكون المادي لأجهزة المحمول، دورة حياة التطبيق، أحجام الشاشات المختلفة، وضيق الوقت في السوق لوجود العديد من المنافسين؛ تستوجب ضرورة تعديل أساليب التطوير الرشيقة، للتناسب بشكل أفضل مع تطوير تطبيقات المحمول في السياق الصناعي ضمن العالم الحقيقي.

وعلى الرغم من وجود العديد من الدراسات التي عالجت كيفية استخدام الأساليب الرشيقة في تطوير التطبيقات التلقيدية، إلا انه لا يزال هناك نقص في الدراسات التي تبحث في كيفية استخدام هذه الأساليب في تطوير تطبيقات المحمول من قبل المطورين، والتحديات التي تواجههم أثناء ذلك، لذلك هدفت هذه الدراسة إلى استكشاف وفهم كيفية استخدام المطورون لأساليب التطوير الرشيقة في تطوير تطبيقات المحمول، وإلقاء الضوء على التحديات التي تواجههم، ولتحقيق هذه الغاية أُجريت هذه الدراسة النوعية مع أربع شركات تختص في مجال تطوير تطبيقات المحمول.

توصلت هذه الدراسة إلى مجموعة من النتائج، أبرزها عدم قابلية جميع مبادئ أساليب التطوير الرشيقة للتطبيق في تطوير تطبيقات المحمول، إلى جانب وجود تحديات جدية أمام المطورين عند استخدام هذه الأساليب في تطوير تطبيقات المحمول كقيود أدوات التطوير الآلية (الاوتوماتيكية) ومتاجر التطبيقات عبر الإنترنت. وبناء عليه استحدثت هذه الدراسة أسلوب رشيق جديد قائم على أسلوب السكرم لمعالجة التحديات الموجودة في تطوير تطبيقات المحمول.

Acknowledgements

I extend my sincere gratitude and appreciation to all who helped me during my studies and in preparing this work. First and foremost, I am very grateful to my supervisor Dr. Samer Zain for his support, guidance, friendship, remarkable suggestions and constructive criticism enabled me to prepare my Master's thesis. I could not imagine the completion of this research without his dedication and provision of a large proportion of his time.

I would like to thank the committee members: Dr. Abdel Salam Sayyad and Dr. Adel Taweel. Their support and encouragement have helped me and their comments guide me. I am also grateful to Dr. Mamoun Nawahdah and Dr. Radi Jarrar for their review of case study protocol. My thanks also go to my friend Elias Khalil for assisting me in communicating with companies to enable me to collect the data. I am also thankful for the participants from the four companies who have participated in my case study.

I would like to thank my parents, my parents started me in the right direction in life and have been supporting me throughout each step. My father, without whom the breadth of my education wouldn't have been possible. My mother who has constantly supported and been a source of inspiration during this whole study period even when I had been far away from home. I would like to give my thanks to my brother and my

sisters for the emotional support, patience and help they provided.

To the most important people of my life, my wife, and my son I owe a special note of gratitude for their encouragement, support, and sacrifice. Without my wife optimism, patience, support, and encouragement, I don't think I could have garnered the energy to complete this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recently, mobile devices are no longer a luxury. In fact mobile devices are used by all types of people no matter their level of income to accomplish their daily tasks. The increasing popularity and adoption of mobile devices in people's lives is due to the ease of use and portability of these devices ; evolution of the computational power ; and rich mobile application (app) markets. Further, the use of mobile apps is no longer limited to the entertainment purpose and have become used in critical fields such as health, education, finance, marketing, etc.. Consequently, mobile devices have become the main target for software industry. As well as, mobile app development project number are increasing .

During the past two decades, the adoption of agile method as software development model is continuously on the rise [1]. The rise of the popularity of agile method is due to its ability to address some shortcomings of traditional methodologies [2]. Further, the increase in the adoption of agile method is a reason to the success of these methodologies and several studies have been carried out to document this phenomenon widely [1]. These methodologies have offered flexibility to work within constraints and ability to adapt to changing market conditions [3].

Previously, most of mobile apps were small or medium in size and scope and targeted entertainment sectors. Thus, one developer or two developed them at the most. Therefore, the development processes of mobile app were not given sufficient attention by developers or researchers and followed limited number of best software development practices [4]. However, the current proliferation of mobile apps requires mobile development methods to produce high level of quality apps.

This purpose can be achieved by adopting recent standards of software development methodologies in the app development life cycle process [4].

However, mobile app development are more complex than traditional development, thus traditional software engineering approaches are not directly applicable in the mobile industry [5]. Firstly, user interface (UI) and user experience (UX) in mobile device provide new features for the user interaction, which have not been previously explored in research yet [6, 7]. Secondly, the fragmentation of mobile platform and devices, each platform has different characteristics and constraints [8, 9]. Therefore, the app may be developed on a platform and used on a different one, or developed different versions of same app tailored for different platform [8, 9]. Thirdly, mobile industry undergoes short time to market requirements and fierce competition [10], so apps should be developed quickly with competitive cost to succeed in the competition of millions of apps on the market [8]. Obviously, Mobile app development is complex and fault prone [11]. In addition, it outpaced the traditional software engineering approaches [5]. Thus, developing and adopting appropriate software engineering approaches is necessary.

Literature proposes different software development methodologies dedicated for mobile app development [12]. Agile method has caught the attention of researchers and software engineers around the world and many studies consider the agile method as a natural fit for Mobile industry [9, 12, 13, 14, 15, 16].

More specifically, mobile apps should be developed quickly since they have a short time for market release, and agile practices help to achieve development because it focus on short development cycle [10]. However, researchers recommend that the development process should be tailored to fit with mobile app requirements [13]. In many cases, the shortcomings of agile method are fulfilled by merging the practice of the agile with practices from other frameworks [8].

Mobile app development is still largely immature and agile methods dedicated for mobile industry is still insufficient [10]. Further, no study has focused on identifying the advantages and issues when adopting agile development methods in mobile app development in real world industrial teams. While there is a study that has focused on traditional web and desktop app [17]. Consequently, this study aims to conduct a multiple-case study to explore and investigate issues and challenges of agile and incremental development methods for mobile apps in real world to understand how industrial teams approach mobile app agile development; explore the challenges they face; uncover implicit issues; and compare results with state-of-the-art.

Therefore, this study aims to achieve several goals. Firstly, conduct qualitative study in order to explore and investigate mobile app development teams that adopts agile method in real world scenarios. Secondly, reveal implicit challenges and needs of real industrial teams. Thirdly, compare what is applied in the real world with what is found in state-of-the-art. Such study can reveal more knowledge and perspectives about how development of mobile apps is being approached, and open door for new research.

1.1 Research Problem and Motivation

Agile method is the most popular software development method, which has achieved widespread success and address shortcoming in traditional approach [2]. However, studies reveal that mobile apps development environments are different from traditional web and desktop apps [5]. Consequently, it is highly recommended to tailor development process to suite for mobile app development.

Although there are several agile software engineering methods dedicated for mobile app development [12], but still, they are largely insufficient [10]. Further, no study has focused on identifying the advantages and issues when adopting agile development methods in mobile app development within real world context. Therefore, conducting qualitative study to explore and investigate mobile apps that adopts agile development in real world can reveal implicit challenges and needs of real industrial teams. Further, comparing what is applied in the real world with what exists state-of-the-art can help developers to solve issues that face them, since some issues may have been resolved by some researches. Such study can reveal more knowledge and perspectives about how development of mobile apps is being approached from industrial perspective and opens a door for new research.

Consequently, the following research questions are formulated:

- **RQ1)** How do the industrial teams use the Agile method in mobile app development?
- **RQ2)** What are the challenges and merits faced by these industrial teams compared with state-of-the-art?
- **RQ3)** How can the agile practices be adjusted to best serve the mobile app development contexts?

Data collection procedures were shaped based on these research questions by focusing on certain aspects and constructing specific interview questions in case study protocol (see Appendix A).

1.2 Theoretical propositions

Theoretical propositions is important for our case study to lead to our case study. The propositions will guide us through the data collection procedure and prioritize the analytic strategy [18]. Therefore, propositions helps to know which data should be focus on it and which data can be ignored. During literature review, we identified the following propositions:

1. Agile development methods are a natural fit for the mobile application development.

2. There are issues and challenges that have not been empirically identified before in adoption agile approach in mobile application development.

3. There is a need to tailor some agile practices to suit mobile application development.

1.3 Study Objectives

This study aims to help in understanding the adoption of agile development methods in the development of mobile apps in real world development teams. As well as, identifying the issues that needs to be addressed and advantages that can be gained and help in solving some issues by comparing what is applied in the mobile app industry with what is exist in state-of-the-art. Thus, the following objectives were formed for this study:

- Illustrate one way of implementing agile and incremental development practices in mobile apps.
- Provide an in-depth understanding of the merits and issues related to agile development.
- Increase the generalizability of existing findings by targeting new development environments.
- Develop an agile method for mobile app development.

1.4 Thesis Structure

This thesis is broken down into eight Chapters.

Chapter 2, Reviews the current state-of-the-art in mobile app development. This chapter provides an overview about general agile method, and the characteristic that makes agile method is fit for mobile app development. As well as, it digs into agile method dedicated to mobile app development; challenges of mobile app development.

Chapter 3, Describes the methodology used for carrying out the research justifying this thesis. This chapter describes the case study, multiple case design, data collection methods applied in this study, and considering data collection and analysis procedures.

Chapter 4, presents the results of the data collected using observations, interviews and focus groups. this chapter is divided into three sections, the first section gives a background of the selected case studies that have participated in this study. Thereafter, the second and third sections present the results of empirical research from companies that were the source of data collection following through the research questions.

Chapter 5, provides detailed analysis of this study finding, by discusses the results of this case study to answer the research questions, and compares them with the state-of-the-art.

Chapter 6, presents proposed agile-scrum method for mobile app development. this method was derived based on this study results and extensive study for researchers about the agile method and the challenges faced by mobile app developers.

Chapter 7, discuss the main threats to validity in this study. This chapter provides the tactics that were followed in this study to improve each validity.

Chapter 8, presents recommendations to improve the mobile app development process and concludes the study and outlines key areas for future research.

Chapter 2

Literature Review

This Chapter provides a critical literature for the stat-of-the-art on mobile apps development field. First, a background about agile method is provided based on literature. Second, studies that showed that the agile practices is a natural fit form mobile app development to overcome these challenges are reviewed. Third, proposed agile method for the development of mobile apps are reviewed. Finally, challenges that make mobile app development different from traditional development are discussed. The critical literature review is based on the methodology provided by Jesson et al. [19].

2.1 Agile Methods

Agile methods were first formally introduced as a software development methods formally in 2001 by providing agile manifesto [20]. Signatories of this manifesto emphasized that agile methods focus on individuals and interactions rather than processes and tools, focus on customer collaboration over follow-up work contract , focusing on efficient software development instead of comprehensive documentation and try to respond to change over following a plan [20]. Therefore, agility of the methodologies means quick delivery, dynamic, aggressively change and context specificity [21].

There are many studies that covered success of adoption of agile practices in software project development [22, 23, 24]. In addition, Asnawi et al. conducted a qualitative study to understand the issues faced when adopting Agile methods by early adopters in Malaysia [25], this study involved 13 member including: engineers, project managers, originators, and CEOs.

The results of this study showed that social and human perspectives are crucial when the adopters begin to use agile strategies. Further, Kumar et al. conducted an online survey to reveal the factors considered by professionals of programming field and the impacts of adopting agile practices on clients and business when practicing agile [26] . This study showed that adopting agile practices could increase the output of the business and raise the client satisfaction.

In terms of adopting agile failure, software development depends on lessons learned from specific project [27]. For example, some of the factors that cause failure of agile project were observed, these factors include lack of preparation and associated support, people's resistance, lack of experience in agile method [28]. Moreover, failure factors were classified into four groups, which are technical, process, people, and organizational [29].

2.2 Agile is a Natural Fit for Mobile App Development

The environment of mobile app development is highly competitive, dynamic and uncertain [13]. This is clear from the unique challenges facing the mobile app industry that will be disused in section 2.4. Therefore, agile method has been proposed as a natural fit for software development of mobile app to provide a solution for these challenges [9, 12, 13, 14, 15, 16]. However, studies carried out for app Agile methods to the development of mobile app indicates there is a need for tailor software development processes to suit the requirements of mobile apps [30]. Therefore, some authors fulfilled some of the shortcomings of the agile method by incorporating practices from other framework [8].

In 2003, the suitability of the agile method for mobile app development was discussed for the first time [14]. Then Abrahamsson mapped between agile home ground themes with mobile app development characters [31]. The mapping demonstrated why agile is most suitable method for mobile app development processes due to short development cycles, highly volatile environment, app level software, small teams, object-oriented environment and identifiable customer.

To evaluate adopting agile methods in mobile app development and how these approach improve the development process, Flora et al. [30] conducted an online survey with 130 participants were responded from mobile community including development team members, agile experts, researchers, and other stakeholders. The result of their study indicates that mobile has dynamic and incomplete requirement, which lead to build mobile app with limited set of features in the first release and update it in later by frequent interaction with clients, these make agile methods suitable for mobile app development as they short time, require flexibility, and reduces time to market. In the same way, Santos et al. conducted a survey to understand how agile practices support mobile app development [32].

With 20 student responded to this survey, most of them do not have experience with development methodologies. The result of their studies showed that agile method is suitable for mobile app development especially in project management and control and development time. As well as, the result shows there are some point still challenges mobile. Another survey was conducted on the mobile research and development community [10], with 130 responses from development team members, consultant and top-level manager. The participant indicated that agile practices should be adapted in mobile app development to mitigate challenges. However, the participant indicated that agile methods must be tailored to each team.

Scharff et al. evaluated the effectiveness of agile practices by empirical study [6]. A study conducted with scrum in classroom setting to assess the effectiveness app of scrum as agile method in mobile app development. The authors claim that Scrum helps mobile app development teams to accomplish their tasks in time and the Scrum is the key factor for the success of the project when the time is constraint. In the same context, Kaleel et al. provided a detailed analysis on scrum practices that suit mobile app development [4]. The author proposed including new practices, which is important for mobile app development life cycle such as market analysis and physical constraint analysis to mitigate uncertainty and technical risks if they are likely to get in early stages. Authors claimed the scrum practices such as adaptability to volatile requirements, daily meetings to make effective communication and technically strong development teams are best suit requirements of mobile app development.

Zein et al, conducted an exploratory multiple case in 2015 involving four software development companies [33]. Their study was the first study to investigate how mobile app development teams applied testing techniques and the challenges that they were facing. The results of their study revealed that there is a lack of industrial teams knowledge about mobile app testing, especially in mobile application life-cycle conformance, context-awareness, and integration testing. In addition, the researchers argued that the industrial mobile app team did not apply formal testing approaches.

Finally, proposed agile methods for mobile app development have also been criticized by Corral et al. [8] for utilization in a real world setting. Due to little attention paid to agile methods for mobile app by researchers to investigate if agile paradigm is adopted or not or if proposed method would help to solve mobile app development. In addition, despite proposed agile methods for mobile app development are encouraging but it is still theoretical [13].

2.3 Proposed Agile Practices for the Development of Mobile Software

There are some research articles proposed mobile software development framework based upon agile principles. These papers were surveyed and four approaches were found, which is Mobile-D, MASAM, Hybrid Methodology, and Scrum Lean Six Sigma. Some of these approaches combine agile with a non-agile method to build a new approach dedicated for mobile app development, the following presents these approaches.

2.3.1 Mobile-D

Mobile-D was introduced as the first attempt to adapt agile practices in mobile app development in 2004 to meet the specific demands of volatile environment [34]. This software development approach was drawn from Extreme Programming, Crystal Methodologies and Rational Unified Process. Mobile-D consists of five sequentially arranged phases as follows: Explore, Initialize, Productionize, Stabilize and System test and Fix. Each phase implies a sprint and has a number of associated stages, tasks, and practices. Despite the sequential organization of its phases, Mobile-D encourages iteration because its activities are based on principles that are in line with agile ground practices such as pair programming and test driven. It is recommended to use this approach by small co-located teams (at most 10 developers) working on short development cycle. However, this approach seems promising and plays an important role in theory, but the description provided by this approach is brief and incomplete to used literally in practice [13].

2.3.2 MASAM

In 2008, Mobile app Software Agile Methodology (MASAM) was proposed as a process for mobile apps development [35], which is based on extreme programming, agile unified process, rational unified process and the software and systems process engineering meta-model. MASAM's structure and detailed implementation show a strong tie with Mobile-D approach and introduces minor variation. This approach proposes simple development cycle comprised of four phases: preparation, embodiment, product development and commercialization. Each phase represent a segment of the development process. app development carried out by test-driven, pair programming and continuous integration. As well as, commercialization phase expands this approach to focus on selling the product. It is recommended to use this approach within small companies from mobile app development [13]. However, this approach is unclear and the author didn't provide any case study of actual implementation in real world [8, 30, 13].

2.3.3 HME

Hybrid Methodology Engineering (HME) was proposed as a new agile method that a hybrid Agile and risk-based methodology based on methodology engineering techniques [36]. This approach creates different development methodologies for different setup due to its belief that there is no single process that can fit in all situations. For achieving this, it combines agile methods, adaptive software development and new product development (NDP). It was created in four iterations, the first iteration contains practices commonly found in Agile methods, the second iteration focuses on introducing a new product to the market by including activity from NPD, the third iteration integrated ASD into the methodology, the final iteration includes prototyping to mitigate the potential risks related to technology. Although this approach is focused but it still ah high-level and have not been provided need details about its phase to apply it for mobile app development [12, 13].

2.3.4 SLeSS

In 2011, Scrum Lean Six Sigma (SLeSS) was proposed as an agile method that integrates scrum and Lean Six Sigma for the development of embedded software for mobile phones [37]. Scrum is a methodology for project management and software development [30]. While Lean Six sigma is a methodology that focuses on improving output and reducing defects and failures in a planned and objective way. This approach does not customize backlog only for development project, but it also use it for process improvement purpose based on statistics. Consequently, adopting SleSS approach foresees following an incremental approach with planned-based methodology, which guarantees achieving performance and quality goals of software development project with progressively improvement the development processes in a statistically basis. Although the author present case study of implementation for this approach but there is no report about its utilization [30].

2.4 Challenges in Mobile Software Development

Unlike the development of traditional apps, mobile app development poses additional issues. This is due to the fact that a technical constraints associated with mobile systems and rapidly changing business requirements [4]. In fact, mobile app development has become an industry with massive potential. This development due to increasing the complexity of mobile devices, increase the efficiency of wireless networks and rapid growth of mobile apps market [10]. With the increase in the number of mobile devices and apps, there is also an increase in the number of challenges it faces development team need to be addressed by practitioners in this field [8, 30, 13]. In this section, various challenges related to mobile hardware and software faced by developers while building mobile app were presented.

Platforms Fragmentation: There are a several of platforms available for mobile operating system, each platform needs apps with different specifications due to fragmentation in the hardware, tools and technology required to build apps on each platform [10, 32, 38]. Consequently, developers need put a lot of analysis and effort to make their app in such a way compatible with all platforms [10, 38].

If the company focus on a single platform as most businesses do, this will reduce the reach of their apps [38]. Develop app run efficiently across various platforms is challenging because there is a need to keep app updated-to-date with frequent update across platforms with limited resources [10, 32].

Hardware Fragmentation: Mobile hardware frequently changes in term of memory, speed, graphics processing, etc. As well as, the processing capability is restricted when compared with desktop computers in term of speed, Inadequate energy supply and memory and computational power [10, 38]. This is challenging, because while developing an app, some of the functions that are used by mobile apps must be disabled since they demand a large volume of memory and fast processor swiftness [38]. Therefore, mobile app development pose a challenge for developers to keep the app run efficiently in both high-end and low-end hardware devices [10].

Novice developer: Some mobile app developers suffer from the lack of knowledge and experience resources to develop an app for variety platform, because they come from traditional web and desktop development background [39]. While it is important to follow the style guideline and design patterns as well as the behavior of each mobile device. This is a challenge for a team that may not create a successful mobile app [10].

Insufficient and incomplete requirements: This is one of challenging aspect. Since the project can easily get off the track if the business is not clear about its requirements. It requires considerable effort and time at the beginning of the project to analyses and understand the customer's requirements. Thus, increasing the workload of development, integration and app testing [10, 38].

Budget and Schedule: The budget is a critical issue why mobile app projects fail, especially due to low funding [10, 38]. As well as, time factor limiting the commercial success of the mobile app, mainly in narrow time for release new release of an app [4]. Some claim that the development of mobile projects are small and therefore need a small budget. But this is a common mistake because mobile app development is a complex process with different stages [10, 38]. In addition, the rapid growth of mobile devices with a limited life span puts pressure on app developers to keep the app is adaptive [4].

When developer of the app has sufficient budget and time, he/she will work with a free mind, which makes him capable of building an effective app. In contrast, the developer who works under the pressure of the tight schedule and limited budget will be limited to two options, either delivering a poor quality app to meet the deadline, or losing the deadline for delivering a high-quality app [10].

User Experience (UX): Unlike desktop devices, mobile devices contain many features affecting the design of the app [10, 32]. One of the most important features are the interaction method with the device. Mobile devices have sensors like the accelerometer that responds to device movement, GPS to collect location data, touch screen treats with gestures and virtual keyboard [4, 10, 38]. These features need additional efforts like sensor handling, UI management and testing complexity [4].

Some developers treat mobile apps as another screen and simply expand an app without realizing user interaction and behavior that are different from traditional apps. This make the app little acceptance and low adoption among users [10, 38].

User Interface (UI): Although the design of the mobile interfaces inherits many of the ideas from traditional user interfaces [10, 38]. However, developers consider screen resolution and screen size [5]. Small screen means little data can be displayed, while user expects to find the information effortlessly. Consequently, developers spends lot of time and effort to design app to display the most relevant information for the user [10]. As well as, they concern of design patterns like touch, motion and location information, and virtual keyboard with small buttons, which are suited for number entry instead of text entry and. These challenging mobile developers to make an effective and acceptable app within small screen. Therefore, the design of the UI is more important than other activities in the development of mobile apps [10, 38].

Data Access: A mobile device is capable of accessing data, either by a web browser or a native app [10, 38]. However, backend integration still one of the major mobile concerns need more planning, research and actual development [4]. This is challenging to app developers since it depends on internet connection. Hence, the app should track disconnected network connection to return to last existing data, reconnect, and update the data as soon as possible [4].

Quality Assurance Issues: Mobile users expect high quality apps and this is a challenge because many companies can suffer if their customers are affected by the low quality of the app [10]. However, mobile testing is a complex process due to the presence of many operating systems and many of the devices and the app have to test on vast number of them [40]. As well as, there are several factors that mobile app should be tested against it, such as apps switching, wireless switching and VPN drop. In addition, tests should be performed on the actual devices to detect any failure in the hardware or network connection [10, 38], which is difficult to get number of devices to test the app running on them [40]. Consequently, there is a need for extensive testing and formal reviews to ensure that the app is high quality before used by customers [10].

Analyzing the Target Users: Developers prefer to identify target users and then develop a mobile app based on their interest. Therefore, the failure to identify the target user and analyze their preferences and interests leads to develop an app without value in the market [10, 38]. Identification of user expectations is a challenge in mobile app development [32].

Security and Privacy: Security is major challenging in the mobile industry is due to there exist many different devices and operating systems, open platform and malicious apps could be installed without detection these type of app [4]. Therefore, the least secure mobile apps may be vulnerable to unauthorized use such as stealing information [10]. It is the responsibility of mobile app developer to work on the issue of preserving, protecting, securing the mobile app, the data transmission and servers [38].

2.5 Classification of Mobile Apps

Literature of software engineering classifies mobile apps into different categories based on different criteria. Mobile applications can be categorized into consumer (End-user) and enterprise apps based on the type of services that app provides [41].

- **End-user apps:** These apps are designed for commercial purposes and it aims to improve and facilitate different aspects of people's lives. These apps are delivered to the users by downloading them frequently from different online stores such as Google Play and iOS App store such as social media apps, travel apps, etc. [41].
- **Enterprise apps:** These apps are designed to meet business needs and aim to increase the efficiency, productivity and satisfaction of company employees [41].

2.6 Discussion

Based upon literature review, with the growing adoption of mobile apps and the presence of an unlimited number of mobile apps, there are additional challenges of mobile apps development that are not common in the development of traditional software apps. Therefore, agile software development practice attracted the attention of software development teams and researchers in mobile app development.

Although some agile frameworks were proposed for mobile app development; these platforms remain theoretical and there is no evidence of how these work in real setting. In addition, there is still a lack of research initiatives to understand the real issues and challenges faced by developers when adopting agile practices in mobile apps development. In contrast, there are many studies carried out to identify the advantages and issues of adopting agile and incremental development methods in the development of traditional web and desktop apps [17].

There is a requisite to explore and investigate when adopting agile methods in mobile app development. Therefore, conducting a qualitative study to explore adopting the agile method in mobile app development can reveal implicit challenges and needs of real industrial teams. Subsequently, this study is aiming to conduct a multiple-case study to explore issues and challenges of adopting agile practices in mobile apps development to draw more general conclusions.

Chapter 3

Research Methodology

This chapter presents the case study research methodology as well as detailed information about the data collection and analysis that are used in this thesis.

3.1 Case Studies

A case study research method is an empirical study that investigates a contemporary phenomenon within its real-life context, especially when the boundary between the phenomenon and context cannot be identified clearly [42]. The qualitative nature of this methodology is concerned with the natural setting of the phenomenon under investigation to obtain detailed qualitative information [43]; which enables the researcher to obtain better answers from participants and get in-depth understanding of the context [44]. This type of research follows a systematic process for data collection, data analysis and generation of results [45].

Case studies are used for exploratory purposes, in addition they can be used for explanatory and descriptive purposes [42]. When exploratory case study research is applied, it is important to include industry-based cases because the context can play an important role in defining an emerging theme or theory [44, 45]. In addition, it is strongly recommended that the case studies are based on several sources of data and evidence [42].

Since the main objective of this study is to investigate issues and advantages of adopting agile methods for mobile apps in the real world. Therefore, the Case study is a perfect fit to this study objective. This will enable the researcher to investigate the adoption of agile and incremental development method in mobile app development industrial context.

3.2 Multiple Case Study Design

This study applies the multiple-case study research method. The results and evidence of multiple case designs are more convincing, which make this design more robust compared with a single case study design [42]. This study investigated and drew conclusions from several industrial cases. Each single case represents a mobile app software development company in Palestine. See Fig. 3.1 that show multiple case study design.

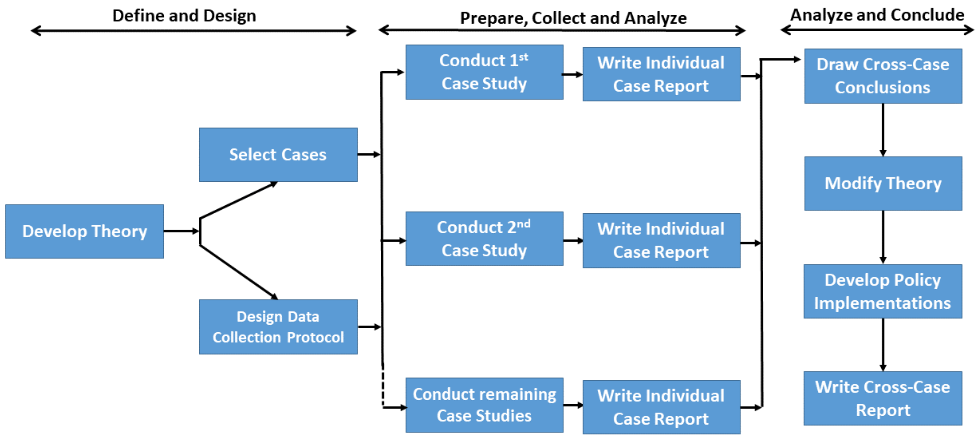

Define and Design
Prepare, Collect and Analyze
Analyze and Conclude
Develop Theory
Select Cases
Design Data Collection Protocol
Conduct 1st Case Study
Write Individual Case Report
Conduct 2nd Case Study
Write Individual Case Report
Conduct remaining Case Studies
Write Individual Case Report
Draw Cross-Case Conclusions
Modify Theory
Develop Policy Implementations
Write Cross-Case Report

FIGURE 3.1: Multiple Case Study Design [42]

The theoretical frame of reference should be defined to clarify the context of the study for whom conducting the research or who reviewing the results. However, in software engineering field theories are still under-developed, so theoretical frame of reference for software engineering studies can be expressed based on relevant work [44].

3.3 Case Selection and Context

In this study, the cases are software development companies in Palestine in the field of mobile app development. The unit of analysis is the development team that consist of developers and testers within these companies. The selection of the companies is based on the availability of team members and willingness of the company management.

3.4 Data Collection Methods

Interviews, observations and focus groups are applied as a data collection methods. This study applied data source triangulation to validate and crosscheck the findings of the study and increases the reliability of the data and the process of gathering it [42].

3.4.1 Observation

Observations are known as a frequent source of information in qualitative research. Unlike interviews, observations often provide more objective information related to the research topic [18]. Data collection through observations is done through taking field notes on the behaviors and activities of developers in mobile app development teams without participation [42] to investigate the challenges and issues of adopting agile method, activities, techniques applied in software development process.

In these field notes, the activities are recorded in unstructured or semi-structured way by the main researcher at the research site. All activities involved in the software development process for mobile apps are observed and investigated. Therefore the observed activities include:

- What is the agile method followed?
- All development activities (requirements elicitation, requirements analysis, design, development, and testing).
- Agile practices used in mobile app development.
- The usage of the IDE (integrated development environments), bug records, and bug management processes .
- Challenges and limitations of using Agile method in mobile app development.

3.4.2 Interviews

In this method, face-to-face interviews are conducted with software development team members. Interviews were semi-structured and open-ended since they are well suited for this kind of research [42] (see Appendix A). Predetermined and flexibly-worded questions are asked to collect tentative answers.

In addition, the researchers asked follow-up questions to probe more issues of interest more deeply. Using this approach encouraged interviewees to express openly and freely situation of the world from their own perspective [42].

3.4.3 Focus groups

These interviews are semi-structured and generally open-ended questions to gather views and opinions from group of participants. By interviewing individuals, you gain efficiency but with a loss in depth. Rather, group interviews allow participants to express themselves when they are part of a group than when they are the target of an interview [18].

3.5 Thematic Analysis

For data analysis, thematic analysis method was applied, which is reported in many qualitative Studies [42, 44, 45]. Thematic analysis is an iterative method for identifying common themes in qualitative data to understand specific phenomena in particular contexts [46]. This method classifies concepts that may affect studied phenomenon such as behaviors, events or activities [47]. These themes divide segments of data into groups, which can be used to provide results by examining the repetitions, similarities, differences or anomalies of groups [48]. It is recommended to maintain a chain of evidence from the findings to the original data [44]. To achieve this, special identification numbers were carefully assigned for each session of the interviews, focus groups, and observations. Later on, each sentence of these sessions was given a special number that is derived from the originating session number. Using this approach, a studied sentences can be easily link to their original session.

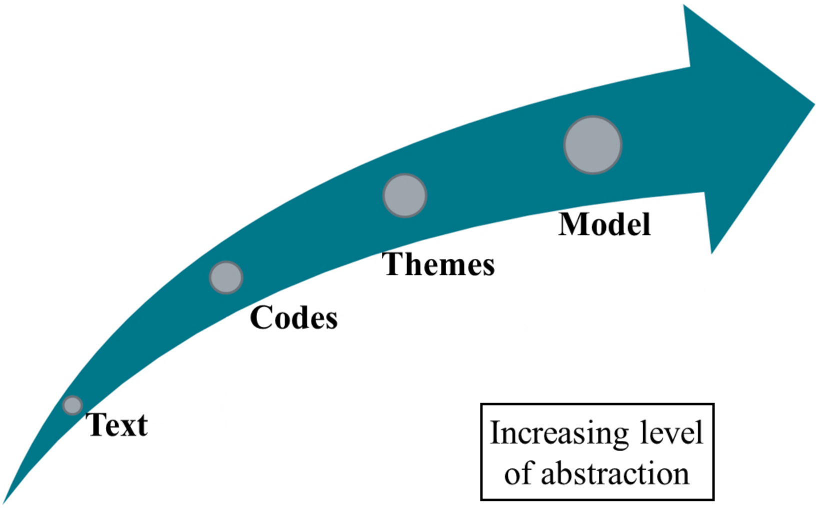


Figure 3.2: Levels of interpretation in thematic analysis [18]

The thematic data analysis process is based on guidelines provided by [46, 49]. First, researchers should be intimately familiar with their data by reading and re-reading it. Secondly, and after breaking down the transcripts into sentences, a set of codes are formulated by generating labels for each sentence. Thirdly, the researcher should search for themes, which are a meaningful pattern in the data relevant to the research question. In our case, themes were carefully selected for challenges and issues in adopting agile method to mobile app development. Due to the iterative nature of this method, the occurrences related to themes were discussed several times during the coding process. Fourthly, reviewing themes should be done by defining the nature of each theme, and the relationship between the themes to reflect story about the data. Finally, a detailed analysis of each theme is conducted, write it by identifying the essence of each theme, and give it a name. The writing should tell a coherent and persuasive story to the reader about the data. See Figure 3.2 that show levels of interpretation in thematic analysis.

3.6 Data Collection Procedures

The availability of team members and willingness of company management were the main factors in the form of data collection strategy in this study. Data collection lasted for a period of three months. In the first case study, the researcher carried out focus group interviews with members of mobile development in one meeting session. In the second case study, data collection was done through observations and face-to-face interviews. Thus, the data collection strategy in this case consisted of two parts: The first part was observation by the researcher for two months, the second part was one-to-one interviews with members of mobile development projects. In the third case, several one-to-one interviews were carried out . In the fourth case, focus group interviews were carried out with members of mobile development in one meeting session and one-to-one interview.

The diversity of data collection methods has helped to apply data triangulation in this research. Moreover, to maintain the chain of evidence [42, 44], the collected data was stored in a way that would easily be retrieved and tracked by another researcher. Therefore, all collected data were stored in documents and spreadsheets (see Appendix B), and all data records were numbered using a special number. Moreover, during thematic analysis coding, each sentence was given a special code and linked to its original document (see Appendix C).

Chapter 4

Results

This chapter presents the results of the data collected using observations, interviews and focus groups. The chapter first gives a background of the selected case studies that participated in this study in terms of the nature of its work and the tools, languages and techniques applied in their work. As well as, the participants that participated in the focus groups and interviews. The case studies are given codes as follows: C1, C2, C3 and C4, to refer to each case with its code to keep confidentiality. Thereafter, the second and third parts present the results of empirical research from companies that were the source of data collection following through the research questions.

4.1 Demographic Information

This section highlights the results from four different cases that are included in our study. The first case study was C1, a company that specializes in mobile app development field with two teams. Each team has two developers and a QA engineer applying agile development processes. They works on a taxi app that deals with web and smart phone clients. Their app is built to support both the Android and iOS platforms.

The second case, C2, is a well-established and large software development company developing various types of apps. This company provides business and software solutions and a wide range of IT services. They are mainly focus in developing mobile apps for enterprise level and apply Kanban (but feature per release) agile process in mobile app development. C2 have four mobile app development teams, each team consists of approximately four members, and offers mobile apps that supports Android, iOS, hybrid, and m-site apps.

The third case, C3, is a relatively old and well-established software development company building web, cloud-based, and mobile apps. The development process applied in this company is Scrum agile .This company has five mobile app team, each team consists of approximately five members. Most of their apps are social and business mobile apps and they support native apps (Android and iOS apps) and hybrid apps.

The forth case study, C4, is a startup software development company. They apply Scrum agile method. Moreover, they are applying best practices in building mobile app by specialized and skilled mobile specialists. They are building hotel and e-travel apps that deal with web and smart phone clients. Their mobile apps support Android and iOS platforms. Additionally, they apply hybrid mobile app framework to be used as a proof-of-concept. Each platform has special development team of five members. The demographic data for each case is shown in Table 4.1.

A set of tools and languages used by each team in developing mobile apps, as well as, the platforms used to produce

Case ID	C1	C2	C3	C4
Business type	Software house	Software house	Software house	Software house
Software house	National	International	International	National
Dev. Method	Scrum Agile	Kanban Agile	Scrum Agile	Scrum Agile
Mobile apps types	Games apps, Taxi apps	Business apps	Social apps, business apps	Hotel apps, travel apps
Single user/ Enterprise apps	Single user	Enterprise	Enterprise	Single user
Team size	2 teams, each 3 members	4 teams, each has 4 members	5 teams, each has 5 members	2 teams, each has 4 members

Table 4.1: Demographics of four cases

mobile apps and tools used by each case to recording the software bugs are shown in Table 4.2. During the three months of data collection through observations, interviews and focus groups, a total of 14 developers.

The total number of participants in the interviews and focus groups from the cases was ten participants. In addition, there is a team of four participants that the researcher has collected his observation data based on their work. Most of the participants in study interviews and focus groups were Project managers. Moreover, the majority of participants have more than 10 years' experience in software development. On the other hand, 50% of the participants in study interviews and

Case ID	C1	C2	C3	C4
Tools used	Eclipse, Android studio, Xcode, Corona, Unity, Visual studio, Notepad++, JUnit	Android studio, Xcode, Appium, Selendroid, Ionic, React native	Android studio, Xcode, AppCode PhoneGap, Junit	Eclipse, Android studio, Xcode, Sublime, React Native, Appcode, PhoneGap
Languages	Java,Objective C, Swift , Lua, C#	Java, Swift, Objective c, JavaScript	Java,Objective C, Swift, JavaScrip	Java, Objective C, Swift, JavaScrip
Platforms	Android, iOS	Android, iOS, Hybrid, M-site	Android, iOS, hybrid	Android, iOS, hybrid
Bug recording tools	MS Excel	Jira	Jira, Bugzilla	Jira

Table 4.2: Tools, languages and techniques applied

focus groups have less than three years' experience in mobile app development. All of them had experience in native mobile app development, while few of them have experience in hybrid and m-site mobile app development. The demographic data for each participant in interviews and focus groups is shown in Table 4.3.

4.2 Development process and Environment

This section provides a discussion of the development processes and their activities. As well as, environment used in

ID	Case ID	Role in the Team	Development Experience	Total Mobile App Development Experience	Experience in Each App Type			Previous Experience
					Native	Hybrid	M-Site	
P1	C1	Project Manager	11 years	7 years	7 years	3 years	5 years	Web development, back-end development & gaming
P2	C1	Team Leader	4 years	4 years	4 years	-	-	No previous experience
P3	C1	Developer	3 years	3 years	3 years	-	-	No previous experience
P4	C2	Team Leader	4 years	4 years	4 years	-	-	No previous experience
P5	C2	Project manager	12 years	8 years	8 years	1 year	2years	Web development & back-end development
P6	C3	Project manager	12 years	3 years	3 years	-	-	Project management & analysis
P7	C3	Project manager	12 years	1.5 years	3 years	-	-	Back-end development & QA engineer
P8	C4	Project manager	1.5 years	0.5 year	0.5 year	0.5 year	-	Communication & marketing
P9	C4	Project manager	10 years	8 years	5 years	2 years	2 years	Web services & enterprise development
P10	C4	Project manager	9 years	3 years	3 years	0.5 year	-	Back-end development

Table 4.3: Interviews and Focus Groups Participants Demographics

mobile app development. The quotations in this section and the following section represent the actual response that was obtained from the respondents' response.

Generally, four cases applied agile method as a development process in mobile app development, all of them applied Scrum method except for C2, in which they applied Agile Kanban. It was noticed that the mobile app development is inherently rapid in all cases, in which their teams had to offer new release about every two weeks. However, C2 provided a release per feature. One of the project manager in C4 said: *"We hope to do a release per feature which means higher frequency for the releases, but we can't do that before we reach 100% automation of testing"* – Project manager, C4.

It was observed that the road map and the features offered by competitors are the main sources of requirements for development teams. In general, the term road map refers to the initial project plan of decisions and what is likely to happen during the course of the project [50, 51]. Moreover, road map should answer a set of questions that are related to markets, products and technologies [52]. In addition, the features offered by competitors are an important source of requirements and the prioritization of requirements in single user mobile apps. On the other hand, in enterprise app development, the team depends on the customer as a basic source of requirements and prioritization. One of the participants mentioned: *“Most of our requirement could be from competitors, so we should know who our competitors are? To understand what is the existing app. And how to benefit from their experience? Because the users use it and familiar with the existence app. Moreover, we search for weaknesses in competitors such as complicated feature to provide this feature simpler to excellence on what exists”* – Project manager, C1.

In C3, one team reported that they divided app (Enterprise app) features between iterations, because the project had fixed cost with fixed time so often no change. Then they gave high priority for high risk features. In addition, they got feedback from the customer after each iteration and acceptance tests.

It was noticed that all teams give great care to the UX (User Experience) more than other aspects in single user mobile apps design. On the contrary, most participants admitted that they give higher attention to functionality than UX when developing enterprise mobile apps. However, one participant did not see this as true for enterprise apps and said: *"Sometimes you should focus in UX in enterprise app, because there are competitors for enterprise app. These competitors are indirect such as Facebook, Instagram, and these competitors imposes competition on you, because your app UX and performance should be similar to these apps"* - Product manager, C4. However, it is a different case in mobile game apps according to a team member from C1, he said: *"The games are a different story because game apps have more challenges in term of art, taste and physiology because of the service apps and often there is a need for such app, but in games often satisfying higher needs so you should make him enjoy in the game and there is no metric to measure the enjoyment. Thus we need to identify who are the target audience carefully (in some case your target could be male or female from the age X to Y years)."* - developer, C1.

Android and iOS platforms are supported by all cases as main development platforms. It was observed that the teams try to apply feature parity between these platforms, which means they they develop the same feature in iOS and Android platforms. The also applied uniform management and testing processes on both platforms. However, some companies use hybrid mobile frameworks to produce a proof-of-concept app to get fast feedback from the customer because it is a cross platform and does not need resources compared to native Android and iOS apps, *"We used hybrid in the beginning, mainly we used ionic as a proof-of-concept to test how the mobile app is important for our business"* – Project manager, C4 and *"The hybrid app usually used for a proof-of-concept. Because of its low investment cost and it is cross platform so no need to build the same app twice, so when we need to take feedback from clients we use hybrid app, but it has bad performance"* – Project manager, C3.

Further, it was clearly observed that there are many constraints on mobile app development. For instance, the diversity of users and the changes they need. As well as, the diversity of mobile devices, platforms, and the emergence of many new devices constantly adding further complexity. Thus, the teams should ensure that the apps work effectively on different devices. These constraints put more pressure on developers and QA engineers to develop and test the same mobile app for different platforms on various devices. *"mobile app needs a fast process to keep alive. Because mobile app always needs to be changed and updated because there are diversity in users and there are always changes in needs. Moreover, there are always new devices need to be compatible with your app."* – developer, C1.

Although the development process of mobile app is known to be rapid, it was noticed that existing tools do not support such constraint. For example, compilation and automation testing tools for mobile apps need more time compared to other web and desktop tools, *"The development isn't smooth as other platform, because the tools don't help you to make the development process fast as other platform as PHP for example. As well as, compilation time is long, so long waiting time and this factor is annoying, especially when the project becomes complex*

- this wastes a lot of time for developers-. So we compile the code on device especially in android to accelerate the build and compilation process." – Project manager, C4.

Moreover, it was observed that there is a lack of statistics about the devices and platforms used by the users. Such statistics are important for critical decision-making in mobile app development, For example, knowing which devices are more common in certain area of targeted users can help to identify which devices should be the focus in development and testing *"It is important to build our decision based on statistics. So we Search for statistics - which do not exist in Palestine - who are using mobile devices? What are the kinds of smart phones they are using in term of platform and type of device? There is no source for such statistics so we adopted our simple statistics"* - CTO, C1.

Regarding testing, it was observed that most cases teams applied manual testing (except for C2). Although participants in this study realized the importance of using the automation testing to get faster and accurate results; they claimed that the test automation tools need more investment in time and resources. *"Automation testing gives you accurate and fast result and it is useful in mobile app development because we are always adding features so we can run automated testing after adding any feature, but to build automation framework and resources you need to invest considerable time and resources, but in our case the scope was limited so it isn't useful to waste QA resources in term of cost and time"* – Project Manger, C3, *"You should innovate in using the automation testing in mobile app development because many things are not ready yet. This causes high cost under the large variation in mobile devices and platforms."* project manager, C4.This result is also came across by Zein et al. study [33].

From another aspect, it was observed that one of the main issues of teams is to obtain regular feedback from customers. They obtained feedback in several ways such as direct contact with users especially at the beginning of the project. They also send questionnaire to the users and sometimes allowing them to give their feedback through special designed form within the mobile app itself. However, C4 applied A/B testing as an indirect way to get user feedback. A/B or split testing is a testing techniques to compare one or more variations of single feature or element, to determine which alternative is better. The variations of the feature are distributed among users to collect quantitative data and compare the variants to each-other based on specific factors (Ex. number of clicks on an item or the time spent by the user on a specific page) [53, 54, 55]. More specifically, C4 implemented the same feature in two different versions and distributed them between users, then using analytic system that collects data about user behavior, they adopted one of these version and keep enhancing the adopted version using A/B testing by splitting it into two version again and so on, *"we follow A/B testing by offering the same feature in different way for the users and dividing these version between the users, then choosing the feature that is more accepted by the users, and enhance this version in future by make another two version from the original version"* – product manager, C4. It is different case in game apps where developers are interested in facial expressions of the user while playing the game to obtain user feedback, *"We record video for user face and screen of* *game without interrupting to monitor the user reaction during the playing to modify the game level based on user reaction"* – developer, C1.

Finally, an interesting result observed was the amount of stress faced by mobile app developers. This is also confirmed by most of the participants (except participants from C2). In addition to the constraints that were mentioned above, mobile app developers suffer from constant stress due to short releases, restrictions imposed by some stores on apps, and the long time to approve uploaded apps. Furthermore, there is a continued concern on the existence of bugs in the apps because customers of mobile apps are less tolerant than other users, *"mobile app development is stress for the programmer because you should fix these bugs quickly."* - developer, c1 and *"There is stress because the release is short and when uploading the release it could be rejected from online store so you should change some features to make the app acceptable by the store"* - project manager, C3.

4.3 Agile Principles

In this section we discuss agile principles that were mentioned by Petersen et al. [17] by focusing on the principles that are applied differently or not applied in our cases. Therefore, we examined whether the development teams apply the following agile principles, and if they tailored these principles to better suit mobile app development, how do they tailored them? the Agile Principles are shown in Table 4.4.

Principle	Definition
Iterations and Increments	The project are developed in several iteration. So, project parts are developed and tested in increments, each increment is independent of others and the output of each increment is used as an input for next one or to be delivered to the market sometimes [17, 56, 57, 58, 59].
Internal and External Releases	Instead of delivering of the increment of project development to the market, it could be used as an input for the next internally or externally used increment [17, 58, 59].
Time Boxing	Time Boxing means that each iteration of project has time limit (i.e. fix duration and deadline) [17, 56, 58, 59, 60].
No Change of Started Projects	When a feature of the project is selected and the implementation is realized, the feature has been started then it is finished [17, 56, 59].

Incremental Deliveries	The project is delivered in batches to the market through small increments each increment contains a chunk of functions. The highest priority functions are delivered first, user can use the project from the first delivery [61, 62]
On-site Customer	On-site customer means that including actual user within the development team and available full time to answer questions. Moreover, by applying this principle the developers can obtain immediate feedback and information by participating the customer in requirement definition and validation activates [63, 64].
Frequent Face-to-Face Interaction	Frequent Face-to-Face Interaction means that team members frequent communicate and meet in form of stand-up meeting as scrum [17, 58]. This practice helps to resolve potential misunderstandings and determine the basis for a smoother implementation of the day-to-day activities of the project [65]
Self-organizing Teams	The team members have authority and responsibility to manage their workload, assigning tasks to members based on need and best fit and making decision making. [66, 67]. Self-organizing team leads to motivating members to commit themselves to their responsibility, greater creativity and higher productivity and quality [68].
Empirical Process	Defined processes cannot be used alone to manage software projects effectively because software projects are complex and changeable during project development time. So, agile method adapts empirical process and encourages the continuous examination and adaptation of work and processes [69, 70]
Sustainable Discipline	Discipline is the foundation for any successful endeavor. As the software requires agility, it requires discipline, both concepts are a counterpart to each other, while the study allows adapt and react to new environment and invent, the discipline give strength and comfort when things are difficult [71].

Adaptive Planning	The planning in agile is less formal [72], and it is adapted for a release with short period and focus on factors that effect on increments delivery [73]. Moreover, Agile always welcomes change in technology, requirements or method itself [57].
Requirements Prioritization	Agile methods specify the most important requirements to implement earlier than others based on their business value [74, 75]. Since whole system functionality cannot be implemented in the same iteration [76]. Therefore, the requirement with less priority is implemented in the upcoming iteration [77].
Fast Decision Making	Agile development empowers members who have the power to make decisions that are not limited to a particular role [78]. However, fast decision making means fast response times, which helps to exploit opportunities. In the long term if the majority of decisions that are made are correct it means that the project is successful [79].
Frequent Integration	Team members' code should be integrated and tested frequently [80, 58, 59, 17]. This practice with small releases enable the team always to deliver working software because it guarantees a permanent availability of an executable system [80].
Simplicity of Design	Agile software processes maintains the simplicity of system design as much as possible at any moment in time [81], maintaining simplicity of design helps the team to work productively with minimal documentation outside the source code [82].
Refactoring	Change internal structure without changing the functionality of the system, Refactoring is applied to simplify the complexity of structure and improve the understandability, as well make the system modifying is cheaper and easier[83, 84, 85]
Team Code Ownership	All team members should understand and contribute to the code, this is important for sustainability because it helps to reduce the risk of losing knowledge when develops leave [86]

Table 4.4: Agile Principles

It has been observed that all cases followed iterations and incremental development principles. Further, it has been confirmed by the participants that the iterations and releases' time in mobile app development is shorter than other web and desktop domains (one week for each iteration and two weeks for each release), as the team member in C1 mentioned: *"The time boxing in mobile is less than other technologies to stay in competition, usually it two week release... But in two-week release there is a high risk because it's difficult to avoid bugs due to rapid development."*Developer, C1.

It was observed that the time-boxing was applied in all cases except for C2 because it follows release per branch. However, the participants from C1 and C4 aim to stop using time-boxing and achieve release per branch, because this approach will help to achieve higher frequent releases (as mentioned in the previous section), project manager of C1 said: *"We hope to reach release per feature, mobile app feature almost need shorter time than other technology, so when you make release per feature that mean you always compete"* - project manager, C1.

Additionally, it was noticed that C1 and C3 did not apply internal and external releases, participants from C3 confirmed that they did not apply this principle for reasons not related to the mobile app development. While a project manager at C1 said: *“We do not apply this principle because the time between the releases is short and it is not allowed to apply this principle.”* - project manager, C1.

Further, it has been observed that in the enterprise apps they violated *no change of started projects* principle, based on what the customers need. For single user apps, some of the developers at C3 believe that there is no need to violate this principle as long as they are following the right road map. Some of the participants believe that this principle should be violated sometimes, project manager of C1 mentioned: *”In some cases, competitors release a new feature in their app. This motivates us to introduce a new feature in our app that is not in our plan as soon as possible as a response to the feature offered by the competitor. Therefore, as a team we have to stop working on the feature we are working on and that was planned from the beginning of the sprint to develop the new feature.”* - project manager, C1.

The cases C2 and C3 focused on enterprise mobile apps and maintained on-site customer principle by involving product owner in the planning sessions. Whereas C4 tried to involve customers using indirect way in decision making by applying A/B testing as mentioned in previous section. Regarding the principle of code refactoring, all cases applied it, but the participants confirmed that refactoring in mobile app development is less than other technologies due to time limitation and small size of the app code.

Regarding the principle of requirements prioritization, it was observed that all cases applied this principle. It is done based on the customers view in the enterprise app. However, in C1, there is a trade-off between developing features on a platform before the other. This is because the number of users for this platform was more than the other, *"Sometimes there is a trade-off between iOS and Android, where do we need to implement feature first? You may decide to make a feature on a platform before the other because the number of users in this* *platform is more than the other."* - project manager, C1.

Chapter 5

Discussion

This Chapter discusses the results of this case study, answers the research questions, and compares them with state-of-the-art. The related work confirmed that agile method is a natural fit for mobile app development industry [9, 12, 14, 15, 16]. Further, the traditional software engineering approaches cannot be applied directly to the mobile industry because the mobile industry has gone beyond these approaches [5]. Therefore, researchers recommend that the development process should be tailored to better suit mobile apps peculiarities [13]. Further, there are several agile software engineering methods proposed for mobile app development [34, 35, 36, 37].

5.1 Software Development Method

Many insights can be concluded from the results. Firstly, based on the response of all interviewees and the result of observation, agile method has been applied in all study cases. This is clear since the studied teams apply iterative and incremental development, which help them to adapt quickly and easily to frequent changes in environment, requirements and technology changes. In addition, development teams are concerned with developing better UX more than other aspects. All students confirmed that the program was being modified All studied cases applied one used one type of agile method in mobile app development, three cases used Scrum and the fourth case used Kanban method. Moreover, All interviewees confirmed that their agile method was being tailored in different ways to suit the nature of mobile app development. However, there are some challenges still facing mobile app developers, all of them will be discussed in the next sections.

5.2 How do teams tailor the Agile method to better suit the mobile app development?

The iterations and incremental development in the agile methods are suitable for the nature of rapid nature of mobile app development. However, during this research, it was noticed that some agile principles were tailored to better suit the mobile app development:

- **Time boxing principal:** Six out of ten interviewees confirmed that time boxing was inappropriate for developing mobile apps and the teams applied "feature per release" instead. The term feature per release refers to the delivery of the feature to the end user once it is completed. Feature per release is more suitable for mobile app development because the features of mobile apps are small (sometimes need a few days to develop one feature) while applying time boxing means committing to a deadline for each iteration. Moreover, it is better to deliver every feature once it completed rather than making the user to wait a certain period of time.Furthermore, adopting feature per release will make users more satisfied since they find that there is continuous updates and additional features being developed continuously. Finally, feature per release can make teams more capable to keep pace with competitors.

- **Internal and external releases principal:** It was observed that this this principal does not perfectly fit with the mobile app development needs. This is because the mobile features are known to be small and do not require internal releases.
- **No change of started project principal:** Eight out of ten interviewees confirmed that the development teams cannot commit to this principle in case of fierce competition. it was observed during data observation that sometimes they should stop working on a feature because the competitors launched a new feature. Thus, the teams should develop a new feature as a response to the competitor's feature.
- **The on-site customer principal:** Since there are large

numbers of users for mobile apps, the on-site customer principal cannot be adopted in the mobile app development. However, it was observed during data observation and 6 out of ten interviewees from focus groups and Interviews participant confirmed that the development teams can get feedback from users using questionnaires, focus groups with sample of users, there are four interviewees said they using indirect ways such as the use of A/B testing.

5.3 The Advantages of Using Agile method in the Mobile App Development

The development teams benefit from the flexibility of agile method by adopting new techniques to address the challenges of mobile app development. For instance, it was noticed from the response of all participants in interviews and focus groups that the teams had to offer rapid releases (one week for each iteration and two weeks for each release) . This insight came across with other studies [40, 5, 13, 26, 33]. In addition, the flexibility of agile methods give all teams in this study the ability to address the challenges of the diversity of mobile devices and platforms by developing the same app for more than on platform (Android and iOS).

Further, the challenge of getting continuous feedback from users, teams applied questioners, direct contact with users, and using indirect contact by using A/B testing and analytics systems. On the other hand, in order to keep pace with the competitors and satisfy low tolerance app users, seven out of 10 interviewees admitted that they continuously analyze the features of other similar apps. This will help them in developing apps with better features. This also came across with the study by Flora et al. and Dehlinger [30, 5].

5.4 Issues of Mobile App Development When Adopting Agile Method

However, there are still some challenges in mobile app development that remain unsolved. Firstly, four out of ten interviewees confirmed that the development tools and IDEs do not support the rapid nature of mobile app development. For instance, the compilation tools take considerably long time to finish. Secondly, there is a lack of adoption of automation testing tools because these tools need more investment in time and resources, all cases use manual testing except one case (C2). This came across with the study by Zein et al. [33]. Thirdly, five out of ten interviewees emphasized that the diversity of mobile devices and platforms put more pressure on developers and QA engineers to develop and test the same mobile app for different platforms on various devices. Finally, five out of ten interviewees confirmed that they suffer from the lack of statistics about the devices and platforms used by the users according to certain categories, such as gender, geographical area, and age to mention a few.

Further, in addition to the known challenges in mobile app development in the literature, four out of ten interviewees they are suffering from work stress for several reasons. Firstly, the rapid development of mobile apps and the short release time. Secondly, the long time and the constraints imposed by some online stores to approve the uploaded apps. Thirdly, developers are highly concerned about discovering app bugs before users do, since that users of mobile apps are less tolerant than the users of other app.

Finally, four out of ten interviewees confirmed that adopting agile method when developing enterprise mobile apps is somehow different as compared to single end-user apps. In this type of apps, the developers give higher attention to functionality than UX. Moreover, they depend on the customer as a basic source of requirements and prioritization. Further, they apply on-site customer principle by involving product owner in the planning sessions as in traditional app.

Chapter 6

Proposing an Agile Method for Mobile App Development

This chapter presents a proposed agile method for mobile app development which was derived based on this study results and extensive study of research about agile method and the challenges faced by mobile app developers. Existing software development methodologies do not fit the development of mobile app because the nature of mobile apps is different from web and desktop apps as mentioned previously in this study. A new Agile-Scrum method for mobile app development includes vital practices that are essential for mobile app development.

The proposed method in this study is an improvement on the agile-scrum method for mobile app development proposed by Kaleel et al. [4]. Figure 6.1 shows the proposed agile-scrum method by Kaleel for mobile app development.

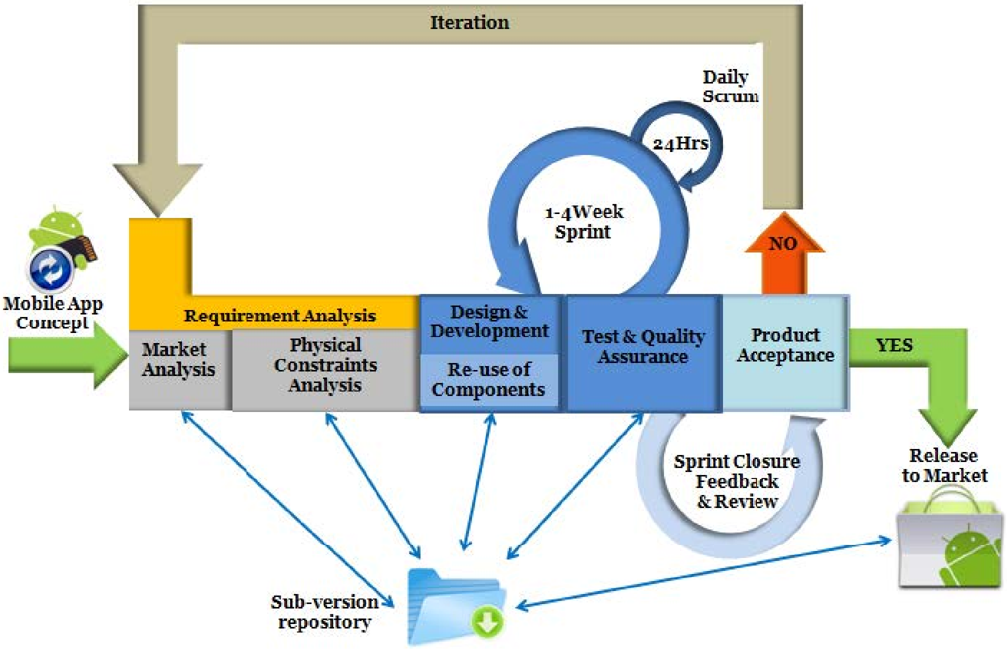


FIGURE 6.1: Proposed Agile-SCRUM Method by Kaleel for mobile App development [4]

The proposed method by this study consists mainly of eight different phases namely: Phase 1: Product Backlog, phase 2: Sprint Backlog, phase 3: Requirements Analysis, phase 4: Design and Development, phase 5: Test and QA, phase 6:

Product acceptance, phase 7: Sprint Closure Feedback and Review and phase 8: Release to Market. As well as, A/B testing across all the life cycle of software development, starting from requirement elicitation and continuing through design and development until testing and QA. The output of phase 1 is a prioritized list of all gathered requirements. In addition, Sprint Backlog consists of a short list of Product Backlog that the development team plans to cover in the sprint. Figure 6.2 shows the proposed agile-scrum method by this study for mobile app development.

Further, Kaleel et al. have incorporated market analysis and physical constraint analysis for smartphone devices in Requirements Analysis phase. Where market analysis enables the development team to answer ”Would it be useful to develop this story?” and enables the development team to know how they will develop the feature by knowing what competitors offer to offer the feature in a better way. Moreover, identifying what the famous apps offer and effect on the UX, such as the Facebook app notification, so that the team can use the same way to implement the notifications feature in their app. Therefore,

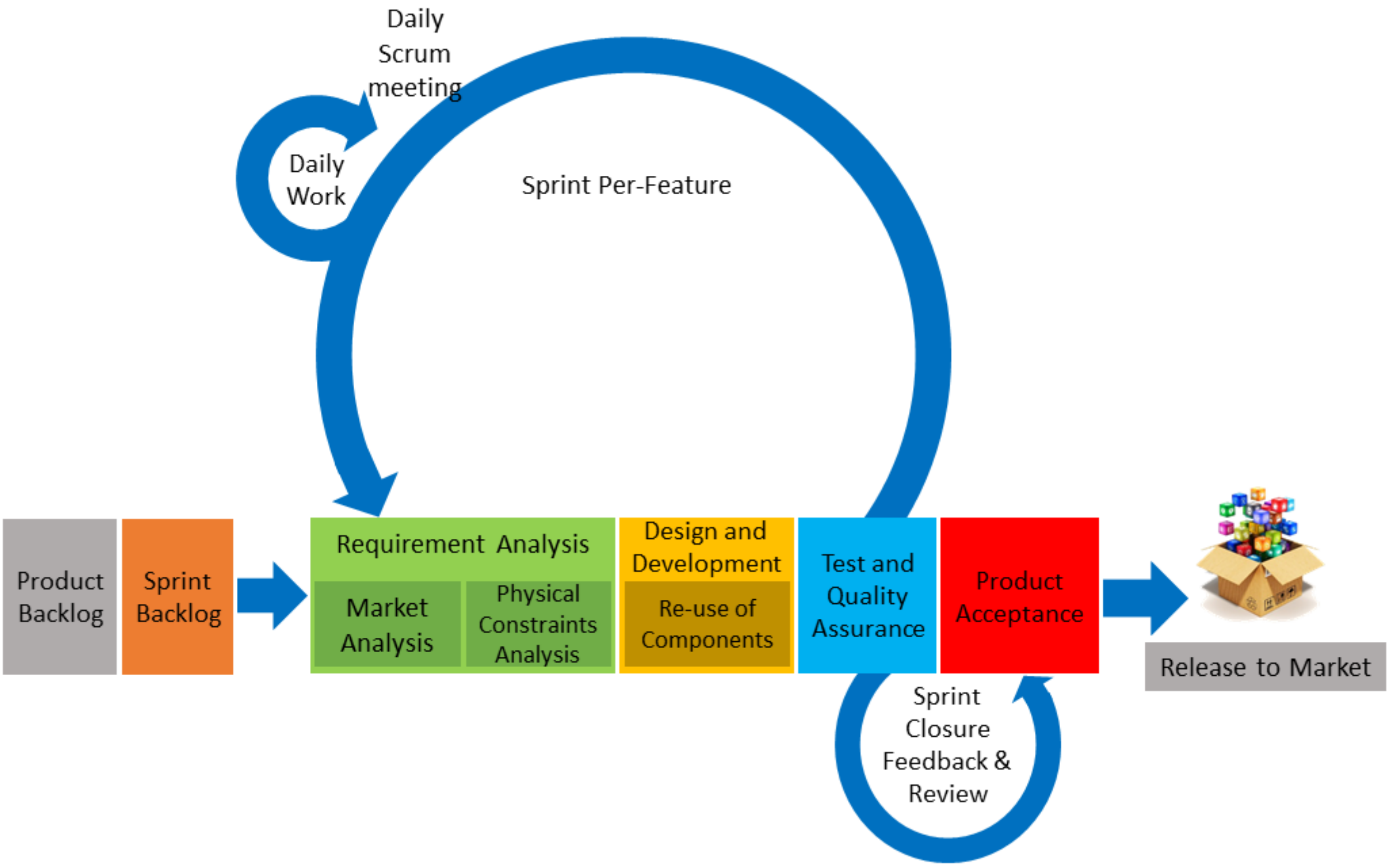


Figure 6.2: Proposed Agile-SCRUM Method for Mobile App Development

this activity will mitigate the challenge of fierce competition in mobile app development. In addition, physical constraint analysis helps the teams to identify new device versions and device capability that used by the users. Including physical constraint analysis in the early stages of development helps in mitigating uncertainties and technical risks [4], especially will reduce the impact of platforms and devices fragmentation.

In addition, the Re-use concept in software development was incorporated into the Design and Development phase, Re-use components is an important concept in the development of mobile apps, which helps speed up the development process and align with the nature of rapid mobile app development. Moreover, sprint in the proposed method lasts until finishing development of mobile feature, at the end of each sprint there is a release to the mobile app store after Product acceptance test. It is known that the development of the mobile features need short time (sometimes need a few days to develop one feature) and are simple compared to the traditional apps. Therefore, the adopting release per-feature means that the app on the store is constantly updated, which means increasing user satisfaction and increasing the app's ability to compete. The daily SCRUM meeting is part of the sprint, this meeting is used as a mean to share updates about tasks that have been completed between team members and the next action plan and expect various issues to address it.

The closure of each sprint phase, the teams held effective feedback and retrospective meetings to document the best practices and shortcomings encountered by the development team in the current sprint. Finally, before the release the app to the official market, the product's acceptance is validated.

As mentioned previously, A/B testing is across all the life cycle of software development in our proposed method, starting from requirement analyzing and continuing through design and development until testing and QA. Recently, the popularity of A/B testing has grown due to the success stories of the companies that used this method such as Microsoft, Google, Amazon and others [87]. A/B or split testing is a controlled experiment [87], used to optimize any new design of the app by improving the conversion rates [88]. In this method, two versions of feature design are assigned randomly to a selected sample of users, each user sees only one version. Key success such as conversion is measured statistically for the two version to determine the preferred version [87, 88].

A/B testing enables the team to improve their app in an incremental way. Therefore, it requires continuous periodic review of the app to improve the user experience of the app [89]. In addition, there is always the possibility of further improvement because a recently completed A/B testing can help the teams to identify deeply what the users like, how they react and their needs which means a new way to improve the app. Figure 6.3 shows A/B testing iterative process.

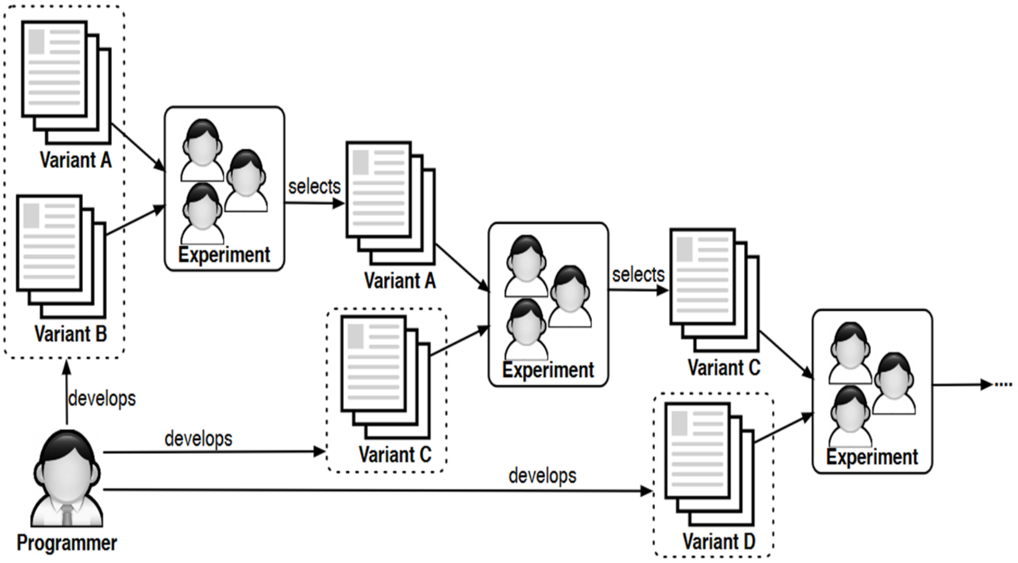


FIGURE 6.3: A/B testing iterative process [89]

Mobile app UX is the most important aspect of mobile app development. Therefore, the UX should be optimized for the desired app. but there is no magic way to know the best performance or design of the app. However, A/B testing is used to make an improvement to the UX and the team can test virtually anything and more than one thing at a time. Moreover, this method shifts the team conversations from ”we think” to ”we know” because it enables data-based decision making. Therefore, the team can make a careful changes the UX while collecting data on the results. Consequently, using A/B testing in mobile app development is not a luxury because it helps the team to improve the app UX based on data from real users.

6.1 Comparative Analysis

Corral et al. performed a comparative analysis of agile methods that dedicated for mobile application methodologies [12]. These methods was included in Literature Review chapter of this study which is: Mobile-D, MASAM, HME and SLeSS. Corral took generic software development life cycle as baseline, which is consists of: analysis, design, implementation, test and deployment phases [12]. However, this section adds the proposed method in this study to this comparison.

6.1.1 Analysis

All methods contain a methodological practice to explore the working setting, identify the user needs and initialize the work settings. In Mobile-D, MASAM and HME focus on creating self-organized teams, promoting communication between roles and customers and conduct the assignment of roles and responsibilities in this early phase [12]. Generally, requirements gathering and planning are simple in agile methods, its focus on establishing an initial set of ideas that welcome and respond to change. on the other hand, Scrum-oriented methodologies like SLeSS and the proposed method follow an iteration-based planning, which is updated and specified based on the outcomes of previous iterations [12].

However, Scrum approach is fit with volatile mobile app environment and changeable requirements. Therefore, each iteration there is a new plan for the next plan based on the current situation of the app in term of requirement and the output of the previous iteration [12]. Moreover, the proposed method embed two activity in this phase: physical constraint analysis and Market analysis, these activities are important for mobile app development to understand the competitors feature and the diversity of devices and platforms to mitigate two important challenges in mobile app development.

6.1.2 Design

Despite Agile methods allow flexible design, but it should be implemented carefully because good and simple design facilitates the quick reaction and reduces the impact performing eventual changes [12]. Mobile-D, MASAM and HME underline architectural design and propose adequate guidance for applying incremental and product line design. Scrum-based methods use the design phase to establish a perspective to process the existing Product Backlog [12]. In addition, SLeSS identifies the attributes that Critical to Quality (CTQ), which represent the most value from the customer perspective to determine the impact of the product output on the customer's goals [12]. In the proposed method, it is recommended to adopt reuse component to speed up the development process, which is required in mobile app development.

6.1.3 Implementation

The goal of this phase is to deliver working software product to the customer based on collaboration between teams and close communication with the customer [12]. However, this phase does not include specific activities in any agile methods. Mobile-D and MASAM follow typical XP techniques [12]. In HME, this phase carried out within Development Engine to create reusable software components, using Test Driven Development [12], which is similar to the proposed method in term of follows reuse component technique. SLeSS proposes carrying out an improvement phase to apply a solution that addresses the issues found in the Measure phase.

6.1.4 Testing

In Mobile-D, MASAM and HME there is a test effort (e.g., a test case or test scenario) for each implemented feature implemented [12]. On the other hand, in SLeSS there are comprehensive test activities as part of the validation of the improvement phase [12]. In the proposed method the test efforts should exist in every sprint, which means for every implemented feature there is a test effort since it follows sprint per feature. Moreover, A/B testing is used not only in this Phase but extends to all phases to improve app UX, as development teams focus on UX more than any other aspect of mobile app development.

Chapter 7

Threats to Validity

Ensuring validity and reliability in a case study provides trustworthiness, credibility and confirmability of the study findings as well as it provides data dependability [42]. To establish the quality of any empirical social research there are four aspects of the validity have been commonly used, which are construct, internal and external validity, and reliability [44]. The case study is subject to validity threats because they are form of empirical social research. There are many tactics that must be used to address the validity that should be applied during all case study phases [44]. However, this study did not consider the internal validity because it is exploratory in nature

[42]. The following describes these validity and tactics that have been followed in this study to address them based on Yin [42] suggestion:

- **Construct validity:** This aspect of validity is concerned with having the correct operational measures of what is being studied during the case study research [42]. To increase construct validity, this study applied the following tactics: First, this study used multiple source of evidence during data collection. Thus the data was collected using observations, interviews and focus groups to ensure case study construct validity. Second, the chain of evidence was established during data collection and data analysis. This tactic was applied by storing the collected data in documents and spreadsheets and each document was given a special number. In addition, during data analysis, each sentence was given unique code and linked to its original document.
- **External validity:** This aspect of validity is concerned

with the extent to which the findings of the study can be generalized. The intention in case study research is to enable analytical generalization to extend the findings of study to cases that have common characteristics [42]. Thus the findings of the study are relevant to these cases [44]. To address external validity Yin proposed using replication [42]. In this study the four mobile development companies represent a replication for understanding the issues and advantages of adopting agile method in mobile app development.

- **Reliability:** This aspect of validity is concerned with the extent to which data collection and analysis can be repeated with the same findings and conclusions. The goal of this validity is to ensure that errors and biases are minimized in the study [42]. To achieve reliability in this study, the procedures that were followed by the researcher were documented in the case study protocol. Moreover, the case study database was developed through storing the data that was collected during observations, interviews and focus groups from the cases and researcher reflections in documents and spreadsheets.

Chapter 8

Recommendations And Conclusion

This chapter presents some important recommendations to participants in the mobile app development industry, whether they are researchers, developers or companies. These recommendations aim to improve and speed up mobile app development. At the end of this chapter, the conclusion and future work section concludes the study and outlines key areas for future research

8.1 Recommendations

In this section, we present our recommendations for the mobile app development process. The recommendations are based on the results presented above and reviewed literature.

Firstly, there is a need for statistics about devices and platforms used by the users. Such statistics can help development teams when developing a new app to know which devices and platforms are used by target users who will use the app. Knowing this information will help developers to take into account the characteristics of the devices and platforms for the app users. In addition, it will help QA engineers to determine the devices and platforms that will use to test the app. As a result, the bugs associated with the characteristics of different devices and platforms will be reduced.

Secondly, the findings of this research study showed that the capabilities of mobile app development tools are limited when compared to traditional web and desktop application tools, and need to be improved. More specifically, the compilation and debugging tools for mobile apps are slow and relatively need more time to execute. Thus, speeding up these tools will help the developers to focus on coding . Further, the test automation tools for mobile app development are still cumbersome and need further enhancements.

Thirdly, There is no doubt that mobile automation test needs to invest in resources and time at the beginning of the project. However, using automation test will be helpful for mobile development team. Especially in advanced release when the number of app features increases and there is not enough time to test all app features manually under short releases. Therefore, using automation test will give the team quick and accurate results about the status of the application and they can repair the bugs before uploading the new version of the app on the app store.

Finally, the restrictions imposed by online stores such as Google Play and App Store add additional overhead on developers. Such restrictions are important to maintain the quality of uploaded apps. However, these restrictions consume valuable time and effort from development teams. Therefore, online stores should mitigate these restrictions as much as possible to speed up delivery of releases to the users.

8.2 Conclusion and Future Work

In recent years, the agile method has attracted the attention of developers and researchers as a natural fit for mobile app development industry. However, mobile app development is different and more complex that traditional web and desktop contexts. Therefore, Agile methods have to be tailored in order to be adopted for mobile app development. Thus, Several agile methods were proposed for mobile app development, but the effectiveness of these methods was not verified in industrial context.

This study conducted an exploratory multiple case-studies that aim to provide in-depth understanding of how mobile app industrial teams approach agile development and the challenges they are facing. The results will help in better understanding of how agile principals are applicable to mobile app development contexts and highlight particular challenges faced by the developers.

This study argues that not all agile development principals can be directly applied at mobile app development projects. Further, mobile app development teams face additional challenges such as stress, inappropriate automation tools, and very tide time-to-market. Consequently, agile-scrum method for mobile app development is designed based on this study results and extensive study for researches about agile method and the challenges faced by mobile app developers. This method includes vital practices that essential for mobile app development. This study argues that the proposed method will address various challenges involved in mobile app development.

Future work intend to validate the effectiveness of proposed method in the real world context. In addition, study how mobile app development teams implement task estimation techniques in agile development. As well as, study the challenges of mobile app security in detail within the mobile app development industry.

Appendices

Appendix A

Case Study Protocol

A.1 Background

There has been a tremendous shift in the use of smart phones and mobile devices recently. Billions of users have plethora of mobile apps available at online stores. Recent estimates show that there are thousands of mobile app developers. We conducted a critical literature review in the area of agile development methods for mobile apps at industrial contexts. During the literature review, research gaps were identified in the body of knowledge. More specifically, little is known about how agile industrial teams in real world development companies approach mobile app development and what are the challenges they face.

We intend to conduct a qualitative study to explore how agile industrial teams develop their apps and what are the challenges and merits they face. The main aim of our study is to explore and increase our knowledge about agile development methods in the context of real-world mobile app development and to justify a set or propositions (hypothesis) we generated based on our literature review. This study is based on the multiple case-study approach and each case represents a typical software development company in the industry. This document serves as the case study protocol. According to Yin [42], it is very important to develop the case study protocol before actual data collection to ensure case-study reliability and validity.

A.2 Study Objectives and Research Questions

There is a lack of research initiatives that aim to understand real issues and challenges faced by mobile application developers when adopting agile practices development process.

In contrast, there are many studies carried out to identify the real advantages and issues of adopting agile and incremental development methods in the development process of traditional web and desktop applications. Therefore, there is a need to explore and investigate when adopting agile practices in mobile application development. The main objective of this case-study is to expand our understanding about how agile development methods implement in mobile applications development to explore problems and advantage of it. Therefore, this study will try to answer the following questions:

- **RQ1)** How do agile teams approach mobile application development?
- **RQ2)** What are the challenges and merits faced by these industrial teams compared with state-of-the-art?
- **RQ3)** How can the agile practices be adjusted to best serve the mobile app development contexts?

Theoretical propositions is important for our case study to lead to our case study. The propositions will guide us through the data collection procedure and prioritize the analytic strategy [18]. Therefore, propositions helps to know which data should be focus on it and which data can be ignored. During literature review, we identified the following propositions:

1. Agile development methods are a natural fit for the mobile application development.

2. There are issues and challenges that have not been empirically identified before in adoption agile approach in mobile application development and there is a need to tailor some agile practices to suit mobile application development.

A.3 Case Study Design

Results from multiple cases are often more convincing and robust than one single case study [45]. Therefore, we will apply multiple-case design as show in Figure A.1; each case is a software development Palestinian company in the field of mobile applications.

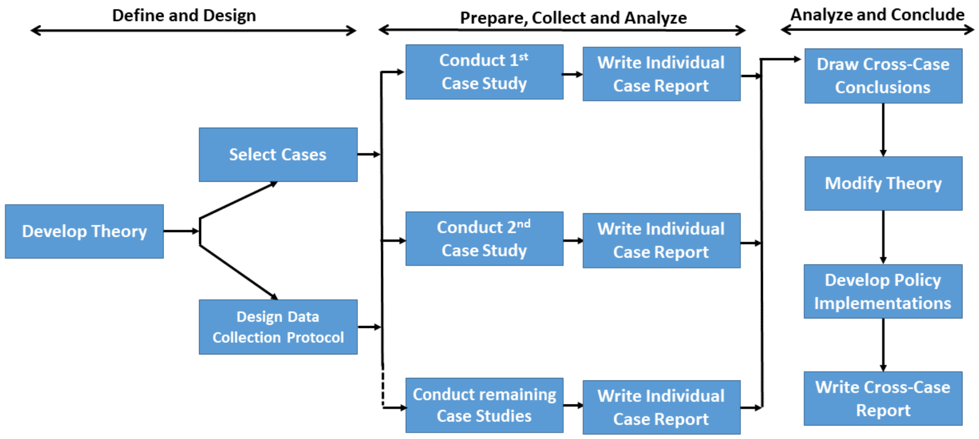

Define and Design
Prepare, Collect and Analyze
Analyze and Conclude
Develop Theory
Select Cases
Design Data Collection Protocol
Conduct 1st Case Study
Conduct 2nd Case Study
Conduct remaining Case Studies
Write Individual Case Report
Write Individual Case Report
Write Individual Case Report
Draw Cross-Case Conclusions
Modify Theory
Develop Policy Implementations
Write Cross-Case Report

Figure A.1: Multiple-Case Study approach according to Yin [42]

A.3.1 Selection of cases

The main criteria that we will follow for selecting cases in this study are based on the availability of members and willingness of company management.

A.3.2 General Procedures

In each interview, the interviewees will be asked a set of questions in the same order in the target companies during normal working periods. These questions will aim to investigate current agile practices applied in mobile applications development and how effective they are. Therefore, the interviewee should have experience in mobile application development through his or her prior involvement in the development of mobile applications.

The main researcher will collect data by interviews, focus groups and observations as part of his master thesis. The second researcher (Master thesis supervisor) will review the protocol and data collected in order to control the validity and reliability of the study.

A.4 Data Collection Methods

Our case study will be based on collecting data through interviews, observations and focus groups. Having three different sources of data collection will ensure case study construct validity through data triangulation [18].

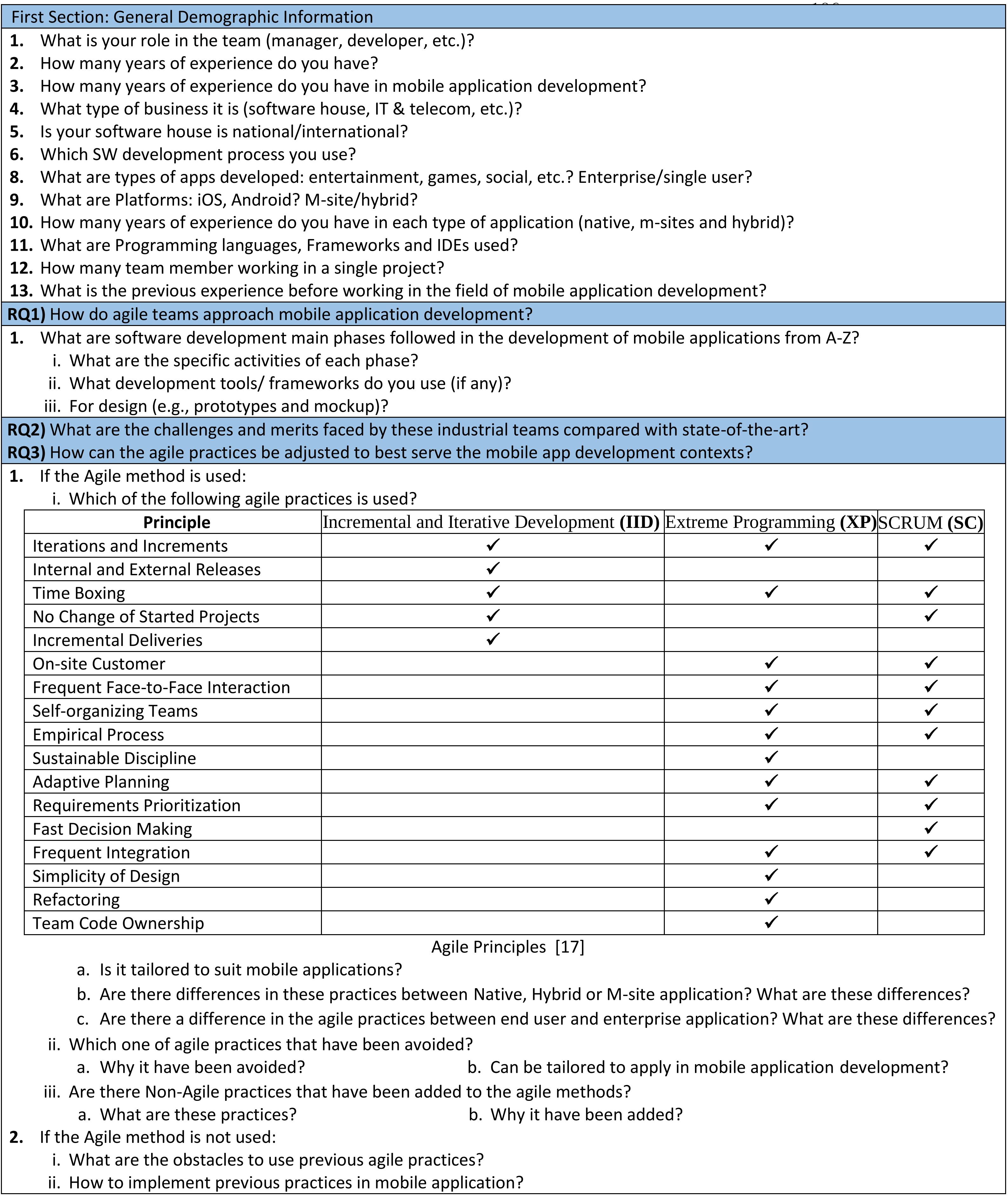
A.4.1 Observation

Observations are a source of information that is frequently used in qualitative research because it provides more objective information about the research topic than interviews [18]. The data is collected through recording field notes by the researcher in unstructured or semi-structured activities in the research site about the behaviors and activities of the mobile application development team. Observations are done without the intervention of the researcher in the team work to investigate the challenges and issues facing developers in adopting agile practices in the development of mobile applications.

The researcher will focus on investigating all activities that involved in software development process for mobile applications. This includes all development activities including requirements, design, development and testing to identify what is the development model is followed. As well as, Identify the tools used in mobile application development such as IDE, bug records, bug management processes, etc. These will help the researcher to reveal the Challenges and limitations of agile practices and techniques applied in mobile application development.

A.4.2 Interviews

Semi-structured open-ended interviews are the most fit for this kind of research [18]. Therefore, the researcher will conduct face-to-face interviews with developers and asking them predetermined and flexibly worded questions, during the interview the researcher will take interview notes. Moreover, the researcher will ask follow-up questions to investigate further issues in depth. This approach helps interviewees to express themselves freely and openly to learn more about issues related to research topic from their own perspective [42]. next page shows interview question.



First Section: General Demographic Information

1. What is your role in the team (manager, developer, etc.)?
2. How many years of experience do you have?
3. How many years of experience do you have in mobile application development?
4. What type of business it is (software house, IT & telecom, etc.)?
5. Is your software house is national/international?
6. Which SW development process you use?
8. What are types of apps developed: entertainment, games, social, etc.? Enterprise/single user?
9. What are Platforms: iOS, Android? M-site/hybrid?
10. How many years of experience do you have in each type of application (native, m-sites and hybrid)?
11. What are Programming languages, Frameworks and IDEs used?
12. How many team member working in a single project?
13. What is the previous experience before working in the field of mobile application development?

RQ1) How do agile teams approach mobile application development?

1. What are software development main phases followed in the development of mobile applications from A-Z?
  i. What are the specific activities of each phase?
  ii. What development tools/ frameworks do you use (if any)?
  iii. For design (e.g., prototypes and mockup)?

RQ2) What are the challenges and merits faced by these industrial teams compared with state-of-the-art?
RQ3) How can the agile practices be adjusted to best serve the mobile app development contexts?

1. If the Agile method is used:
  i. Which of the following agile practices is used?

Principle	Incremental and Iterative Development **(IID)**	Extreme Programming **(XP)**	SCRUM **(SC)**
Iterations and Increments	✓	✓	✓
Internal and External Releases	✓		
Time Boxing	✓	✓	✓
No Change of Started Projects	✓		✓
Incremental Deliveries	✓		
On-site Customer		✓	✓
Frequent Face-to-Face Interaction		✓	✓
Self-organizing Teams		✓	✓
Empirical Process		✓	✓
Sustainable Discipline		✓	
Adaptive Planning		✓	✓
Requirements Prioritization		✓	✓
Fast Decision Making			✓
Frequent Integration		✓	✓
Simplicity of Design		✓	
Refactoring		✓	
Team Code Ownership		✓	

Agile Principles [17]

    a. Is it tailored to suit mobile applications?
    b. Are there differences in these practices between Native, Hybrid or M-site application? What are these differences?
    c. Are there a difference in the agile practices between end user and enterprise application? What are these differences?
  ii. Which one of agile practices that have been avoided?
    a. Why it have been avoided?
    b. Can be tailored to apply in mobile application development?
  iii. Are there Non-Agile practices that have been added to the agile methods?
    a. What are these practices?
    b. Why it have been added?

2. If the Agile method is not used:
  i. What are the obstacles to use previous agile practices?
  ii. How to implement previous practices in mobile application?

Interview questions

A.4.3 Focus groups

This approach will follow the same interview questions style (semi-structured and open-ended) to collect views and opinions from participants. This kind of data collection Helps inter interviewees to support each other in their answer and helps the researcher to understand mobile application development from different perspectives [42].

A.4.4 General Analytic Strategy

It is highly recommended to follow thematic analysis approach to examining rival explanations in general analytic strategy. In this approach, direct rivals are selected from different rivals. Whereas the direct rivals are an intervention other than the target intervention for the results calculation [42]. The details of analysis process was inspired in [44]:

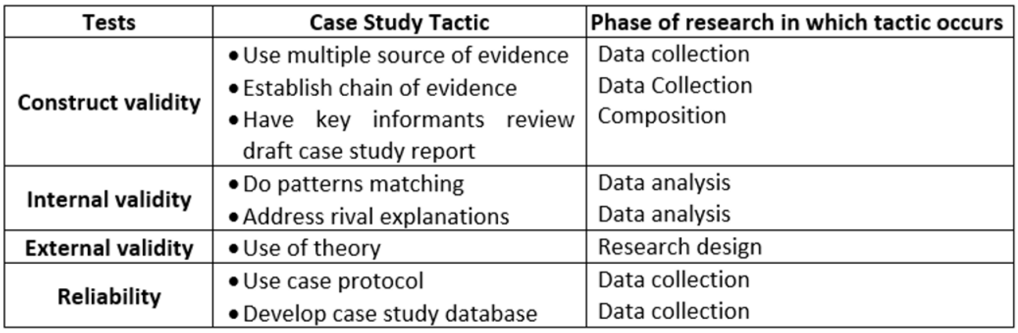
1. Review and study the collected data by the main researcher.
2. Based on the previous step, the main researcher will formulate a set of codes. This step will be done by done

after these codes discussed with the second researcher to increase the validity of study result.

3. Read the collected material again in order to assign codes to contents. This step is iterative and may require reformulating the code that was formed in the previous step.
4. Draw conclusions based on coded data.

A.5 Case-Study Validity Tactics

We will apply four tests have been commonly used in most of the empirical social research to strengthen the quality of the study, It is highly recommended to follow it [42]. Table A.1 shows these tests with the corresponding tactic to ensure theses test.



Tests	Case Study Tactic	Phase of research in which tactic occurs
Construct validity	• Use multiple source of evidence • Establish chain of evidence • Have key informants review draft case study report	Data collection Data Collection Composition
Internal validity	• Do patterns matching • Address rival explanations	Data analysis Data analysis
External validity	• Use of theory	Research design
Reliability	• Use case protocol • Develop case study database	Data collection Data collection

Table A.1: Case Study Tactics for Four Design Tests
[42]

Appendix B

Data Collected

This appendix presents the collected data from the studied cases. The data was stored in documents and each document was given a reference ID. This appendix displays each document separately.

Name:			
Company:			
Date:	30 /10/2017		
Type of interview:	Focus group		
REF ID:	ART1		
Case ID:	C1		

First Section: General Demographic Information

1. What is your role in the team (manager, developer, etc.)?
Project Manager

2. How many years of experience do you have?
11 years

3. How many years of experience do you have in mobile application development?
7 years

4. What type of business it is (software house, IT & telecom, etc.)?
software house /service

5. Is your software house is national/international?
National

6. Which SW development process you use?
Scrum Agile

7. What are types of apps developed: entertainment, games, social, etc.? Enterprise/single user?
Games, Taxi service apps
single user

8. What are Platforms: iOS, Android? M-site/hybrid?
IOS, Android

9. How many years of experience do you have in each type of application (native, m-sites and hybrid)?
7 years Native, 3 years Hybrid, 5 years m-site

10. What are Programming languages, Frameworks and IDEs used?
Android, Objective c, corona, Unity, Visual studio, eclipse android studio, Notepad++, lua for corona and MS excel for bug tracking, JUnit

11. How many team member working in a single project?
Depends on project (we have two teams 5 members (2 for android and 2 for IOS)

12. What is the previous experience before working in the field of mobile application development?
Web development, and backend development, gaming

1. What is your role in the team (manager, developer, etc.)?
   Team Leader
2. How many years of experience do you have?
   4 years
3. How many years of experience do you have in mobile application development?
   4 years
4. What type of business it is (software house, IT & telecom, etc.)?
   software house /service
5. Is your software house is national/international?
   National
6. Which SW development process you use?
   Scrum Agile
7. What are types of apps developed: entertainment, games, social, etc.? Enterprise/single user?
   Games, service
   single user
8. What are Platforms: iOS, Android? M-site/hybrid?
   IOS, Android
9. How many years of experience do you have in each type of application (native, m-sites and hybrid)?
   4 years for IOS and Android
10. What are Programming languages, Frameworks and IDEs used?
   Android, Unity, android studio, swift, JUnit ,Xcode, and Excel for bug management
11. How many team member working in a single project?
   5 members
12. What is the previous experience before working in the field of mobile application development?
   No previous experience

---

1. What is your role in the team (manager, developer, etc.)?
   Developer
2. How many years of experience do you have?
   3 years
3. How many years of experience do you have in mobile application development?
   3 years
4. What type of business it is (software house, IT & telecom, etc.)?
   software house /service
5. Is your software house is national/international?
   National
6. Which SW development process you use?
   Scrum Agile
7. What are types of apps developed: entertainment, games, social, etc.? Enterprise/single user?
   service
   Single user
8. What are Platforms: iOS, Android? M-site/hybrid?
   Android and IOS
9. How many years of experience do you have in each type of application (native, m-sites and hybrid)?
   Native

---

10. What are Programming languages, Frameworks and IDEs used?
Android, Java, eclipse and android studio,

11. How many team member working in a single project?
we have two teams , 5 members for each team

12. What is the previous experience before working in the field of mobile application development?
No previous Experience

RQ1) How do agile teams approach mobile application development?

1. What are software development main phases followed in the development of mobile applications from A-Z?
i. What are the specific activities of each phase?
ii. What development tools/ frameworks do you use (if any)?
iii. For design (e.g., prototypes and mockup)?

Our requirement from:

- Business plan
- Team members
- Most of our requirement could be from competitors, so we should know who are our competitors? To understand what the existing. And how to benefit from their experience? Because the users use it and familiar with the existence application. Moreover we search for weaknesses in competitors such as complicated feature to provide this feature more simpler to excellence on what exists

We decide whether we will accept the new requirement or not after check the validity of the idea versus it value and the challenges that will be faced in business, culture and mobile technology such as in Palestine there is no 3g. However, the user care about their business (how your app will satisfy their needs) not about your technology, So we thinking in the new culture that will introduce to the user and their resistance for the new idea.
It is important to build our decision based on statistics. So we Search for statistics (almost not exist in Palestine) who are use mobile device? What are the kind of smartphone they use in term of platform and type of device? (There is no source for such statistics so we adopted or simple statistics).

If the requirement is accepted :
We sketch the UI as a wireframe using UXPin tool. Actually, The UX is more important than technical side to gain user satisfaction. There is often a discussion between the UX specialist and product manager about the UI. They take into consideration competitors' designs and when our application will be used (during driving, running or setting …) to customize our application to be suited with the user situation in term of interface for example if the application will be used during running the button should be large because the user will use the application without full concentration

The features and UI of the feature are listed in the backlog using Bugzilla and It is given priority according to business plan and sometimes according to its importance compared to what the competitor offers. At the beginning of the iteration, the feature is chosen based on its priority. The feature is broken down into subtasks and distributed among developers. The feature/features is developed and then tested manually throughout the iteration (two weeks). After every two weeks we have a release.

We take the feedback from the user (almost using tracking tools, in some time we have to contact directly with user to take feedback specially in begging of the application because there is no large number of user in some case we ask user to use our application and take the feedback Immediately). Sometimes if there is a need we redesign the feature based on the feedback we get.

In Enterprise we focus more on functional meanwhile in single user we focus on UX and usability

In mobile game we don't take feedback directly from user, so we record video for user face and screen of game without interrupting to monitor the user reaction during the playing to modify the game level based on user reaction

Game is a different story because game applications have more challenges in term of Art, taste and physiology because in service application often there is a need for such application but in games often satisfy high needs so you should make him enjoy in the game and there is no matrix to measure the Enjoyment, so we need to identify who are the target audience carefully (in some case your target could be male or female from the age X to Y years).

Application that need to be runnable in diverse platform but there is no sufficient resource we use hybrid for this application

Mobile product needs fast process to keep alive

Mobile always need change and update because there are diversity in users and there always changes in needs, Moreover, there are always new devices need to be compatible with your application

RQ2) What are the challenges and merits faced by these industrial teams compared with state-of-the-art?
RQ3) How can the agile practices be adjusted to best serve the mobile app development contexts?

1. If the Agile method is used:
   i. Which of the following agile practices is used?

Principle	
Iterations and Increments	Yes, project divided into release based on list of feature Some times in release that contain major changes (in DB or backend), you should to make sure that release that should work compatible with each other, so you should remain old release until new release become stable and all customer are satisfy with new release so you will drop new release
Internal and External Releases	We do not apply this principle because the time between the releases is short and it is not allowed to apply this principle. It can be applied in case we agree with a group of users to use internal release before we deliver external release, but this can be costly
Time Boxing	in mobile is less than other to stay in competition, usually it two week release and one feature with small enhancement and bug fixing, But at the two-week release there is a high risk because it's difficult to avoid bugs due to rapid development. So mobile application development is stress for the programmer because you should fix these bugs quickly. Moreover, some stores (such as AP store) need a long time (four days) to approve on a new release we hope to reach release per feature, mobile app feature almost need shorter time than other technology, so when you make release per feature that mean you always compete
No Change of Started Projects	Yes, In some cases, competitors release a new feature in their app. This motivates us to introduce a new feature in our app that is not in our plan as soon as possible as a response to the feature offered by the competitor. Therefore, as a team we have to

	stop working on the feature we are working on and that was planned from the beginning of the sprint to develop the new feature
Incremental Deliveries	Yes as traditional development
On-site Customer	No, diverse customer
Frequent Face-to-Face Interaction	Yes as traditional development
Self-organizing Teams	Yes as traditional development
Empirical Process	Yes as traditional development
Sustainable Discipline	Yes as traditional development
Adaptive Planning	Yes as traditional development
Requirements Prioritization	Sometimes there is a trade-off between IOS and Android, where do we need to implement feature first? You may decide to make a feature on a platform before the other because the number of users in this platform is more than the other. User requirement (from feedback) some times
Fast Decision Making	Yes, sometimes we make campaign (new feature) not in plan as response for competitor new feature
Frequent Integration	Yes as traditional development
Simplicity of Design	Yes as traditional development
Refactoring	Refactoring in mobile is little compared to other applications because there is no time
Team Code Ownership	Yes as traditional development

a. Is it tailored to suit mobile applications?
b. Are there differences in these practices between Native, Hybrid or M-site application? What are these differences?
c. Are there a difference in the agile practices between end user and enterprise application? What are these differences?

ii. Which one of agile practices that have been avoided?
a. Why it have been avoided? b. Can be tailored to apply in mobile application development?

iii. Are there Non-Agile practices that have been added to the agile methods?
a. What are these practices? b. Why it have been added?

2. If the Agile method is not used:
i. What are the obstacles to use previous agile practices?
ii. How to implement previous practices in mobile application?

Name:	
Company:	
Date:	18/10/2017
Type of interview:	Single interview
REF ID:	ET1
Case ID:	C2

First Section: General Demographic Information

1. What is your role in the team (manager, developer, etc.)?
   Team Leader
2. How many years of experience do you have?
   4 years
3. How many years of experience do you have in mobile application development?
   4 years
4. What type of business it is (software house, IT & telecom, etc.)?
   Software house
5. Is your software house is national/international?
   International
6. Which SW development process you use?
   Agile (Kanban)
7. What are types of apps developed: entertainment, games, social, etc.? Enterprise/single user?
   Business application
   Enterprise
8. What are Platforms: iOS, Android? M-site/hybrid?
   IOS and android
9. How many years of experience do you have in each type of application (native, m-sites and hybrid)?
   4 in android and 2years in IOS
10. What are Programming languages, Frameworks and IDEs used?
    Java, android studio swift, JavaScrip, Xcode, Appium, Jira
11. How many team member working in a single project?
    4 teams in , 4 members (3devlopers and 1 QA)
12. What is the previous experience before working in the field of mobile application development?
    No Previous Experience

RQ1) How do agile teams approach mobile application development?

1. What are software development main phases followed in the development of mobile applications from A-Z?
   i. What are the specific activities of each phase?
   ii. What development tools/ frameworks do you use (if any)?
   iii. For design (e.g., prototypes and mockup)?

   We work as outsourcing, so we take ready requirement with design and UX for each feature from client, Sometimes we intervene if the requirement or the UX is illogical or does not fit the current application to reach a logical solution with the client. Each two weeks we make a release, we input the requirement in to Jira, each sprint we make breakdown for each story and make a planning and estimation then we distribute it between developers after develop it we testing it and upload it on the store

Development in android for UI is faster than IOS because there are many library support that (UI is the half of feature in mobile)

RQ2) What are the challenges and merits faced by these industrial teams compared with state-of-the-art?
RQ3) How can the agile practices be adjusted to best serve the mobile app development contexts?

1. If the Agile method is used:

i. Which of the following agile practices is used?

Principle	
Iterations and Increments	Each release take two weeks. At end of each release there are two days for code freeze. We make the same feature in IOS and Android if on feature doesn't ready on one platform we make release on another platform and doesn't make it in another platform
Internal and External Releases	Each one week we make beta release for internal testing and using
Time Boxing	yes
No Change of Started Projects	If client request an new feature during the release we stop working on the current feature and we start on new feature
Incremental Deliveries	Yes
On-site Customer	Yes
Frequent Face-to-Face Interaction	Daily standup meeting
Self-organizing Teams	yes
Empirical Process	yes
Sustainable Discipline	As scrum empirical process
Adaptive Planning	Yes it require as another platforms
Requirements Prioritization	Client request has high priority.
Fast Decision Making	Yes
Frequent Integration	We use to testing technique , one for each feature if the feature is accepted the we integrate it with the project and make integration testing using manual testing
Simplicity of Design	Yes
Refactoring	Yes,
Team Code Ownership	yes

a. Is it tailored to suit mobile applications?

b. Are there differences in these practices between Native, Hybrid or M-site application? What are these differences?

c. Are there a difference in the agile practices between end user and enterprise application? What are these differences?

ii. Which one of agile practices that have been avoided?

a. Why it have been avoided?

b. Can be tailored to apply in mobile application development?

iii. Are there Non-Agile practices that have been added to the agile methods?

a. What are these practices?

b. Why it have been added?

2. If the Agile method is not used:

i. What are the obstacles to use previous agile practices?

ii. How to implement previous practices in mobile application?

Name:	
Company:	
Date:	26/10/ 2017
Type of interview:	Single Interview
REF ID:	ET2
Case ID:	C2

First Section: General Demographic Information

1. What is your role in the team (manager, developer, etc.)?
Project manager

2. How many years of experience do you have?
12 years

3. How many years of experience do you have in mobile application development?
8 years

4. What type of business it is (software house, IT & telecom, etc.)?
Software House

5. Is your software house is national/international?
International

6. Which SW development process you use?
Kanban Agile

7. What are types of apps developed: entertainment, games, social, etc.? Enterprise/single user?
Business application
Enterprise

8. What are Platforms: iOS, Android? M-site/hybrid?
IOs, Android, M-site, Hybrid

9. How many years of experience do you have in each type of application (native, m-sites and hybrid)?
8years Native, 1 year Hybrid and 2 years m-site

10. What are Programming languages, Frameworks and IDEs used?
Java Script, Objective c , swift, android native, Xcode, react native, ionic, Appium and Selendroid, Jira

11. How many team member working in a single project?
4 teams , Often 4 members

12. What is the previous experience before working in the field of mobile application development?
Web application, and backend services

RQ1) How do agile teams approach mobile application development?

1. What are software development main phases followed in the development of mobile applications from A-Z?
i. What are the specific activities of each phase?
ii. What development tools/ frameworks do you use (if any)?
iii. For design (e.g., prototypes and mockup)?

We receive requirement with mockups and break it down to task and put estimate on it and put it on Jira and the developer start develop on these requirement and then test it using manual test, if user need and paid we use automation for UI and feature

RQ2) What are the challenges and merits faced by these industrial teams compared with state-of-the-art?
RQ3) How can the agile practices be adjusted to best serve the mobile app development contexts?

1. If the Agile method is used:

i. Which of the following agile practices is used?

Principle	
Iterations and Increments	Yes 2 weeks for each sprint and release
Internal and External Releases	Yes we have special group for internal release
Time Boxing	yes
No Change of Started Projects	Yes
Incremental Deliveries	Yes its agile
On-site Customer	Yes product owner always exist
Frequent Face-to-Face Interaction	There is standup meeting
Self-organizing Teams	Yes
Empirical Process	As scrum
Sustainable Discipline	Yes
Adaptive Planning	Yes
Requirements Prioritization	Based on client plan
Fast Decision Making	Yes
Frequent Integration	Yes
Simplicity of Design	Yes
Refactoring	Often there are no factoring because there is time restriction so we follow dimple design from the beginning
Team Code Ownership	yes

a. Is it tailored to suit mobile applications?

b. Are there differences in these practices between Native, Hybrid or M-site application? What are these differences?

c. Are there a difference in the agile practices between end user and enterprise application? What are these differences?

ii. Which one of agile practices that have been avoided?

a. Why it have been avoided? b. Can be tailored to apply in mobile application development?

iii. Are there Non-Agile practices that have been added to the agile methods?

a. What are these practices? b. Why it have been added?

2. If the Agile method is not used:

i. What are the obstacles to use previous agile practices?

ii. How to implement previous practices in mobile application?

Name:	
Company:	
Date:	28/11/2017
Type of interview:	Single interview
REF ID:	AT1
Case ID:	C3

First Section: General Demographic Information

1. What is your role in the team (manager, developer, etc.)?
   Project Manager
2. How many years of experience do you have?
   12 years
3. How many years of experience do you have in mobile application development?
   3 years
4. What type of business it is (software house, IT & telecom, etc.)?
   Software house
5. Is your software house is national/international?
   international
6. Which SW development process you use?
   Scrum Agile
7. What are types of apps developed: entertainment, games, social, etc.? Enterprise/single user?
   Social apps
   Enterprise
8. What are Platforms: iOS, Android? M-site/hybrid?
   IOS, Android and hybrid
9. How many years of experience do you have in each type of application (native, m-sites and hybrid)?
   3 years in IOS and Android
10. What are Programming languages, Frameworks and IDEs used?
    Android studio, Xcode, AppCode, phonegap, java, Android, Swift and objective c, and Jira for bug tracking, JUnit
11. How many team member working in a single project?
    We have 5 teams, In average 5 persons in each team
12. What is the previous experience before working in the field of mobile application development?
    Project management and analysis

RQ1) How do agile teams approach mobile application development?

1. What are software development main phases followed in the development of mobile applications from A-Z?
   i. What are the specific activities of each phase?
   ii. What development tools/ frameworks do you use (if any)?
   iii. For design (e.g., prototypes and mockup)?

   We start with scrum 0 (the first release take from 3 weeks to 1 month), we analysis the requirement and identification for the core feature in the project and ask client if there any unclear issue. After that, we take a look into the requirement and almost it clear and divide it into user story and explain it for the client, then we estimate how each sprint needs time, then we try to put time-frame for the project.

Then we start development in releases and manage the backlog using Jira, and the user acceptance test at the end of each sprint and we take the feedback from user, these feedback could be done in the next sprint or treat it as ad hook and fix it within current sprint if it was simple bug. At each sprint there are retrospective with client. the first development release could take 1 month then the next releases should be 2 weeks

We used wireframe which sketched by UXPin for UI we add them in sprint planning

The hybrid application usually used for a proof of concept. Because of its low investment cost and it is cross platform so no need to build the same application twice, so when we need to take feedback from clients we use hybrid application, but it has bad performance

RQ2) What are the challenges and merits faced by these industrial teams compared with state-of-the-art?
RQ3) How can the agile practices be adjusted to best serve the mobile app development contexts?

1. If the Agile method is used:

i. Which of the following agile practices is used?

Principle	
Iterations and Increments	2 weeks release because we need more regular feedback and the mobile app feature is simpler than other platforms
Internal and External Releases	Based on project, this may cost process
Time Boxing	Yes
No Change of Started Projects	This depend on client (because we outsourcing)
Incremental Deliveries	Yes
On-site Customer	Yes, in enterprise the product owner often exist in planning, but in single end user sometimes we use focus group with influencers and take feedback
Frequent Face-to-Face Interaction	Yes,
Self-organizing Teams	Yes
Empirical Process	Yes, as it in scrum
Sustainable Discipline	yes
Adaptive Planning	Yes
Requirements Prioritization	Based on client
Fast Decision Making	As another platform
Frequent Integration	Yes, we use and manual testing and stress test
Simplicity of Design	Yes
Refactoring	Yes, but less than another platform
Team Code Ownership	yes

a. Is it tailored to suit mobile applications?

b. Are there differences in these practices between Native, Hybrid or M-site application? What are these differences?

c. Are there a difference in the agile practices between end user and enterprise application? What are these differences?

ii. Which one of agile practices that have been avoided?

a. Why it have been avoided? b. Can be tailored to apply in mobile application development?

iii. Are there Non-Agile practices that have been added to the agile methods?

a. What are these practices? b. Why it have been added?

2. If the Agile method is not used:

i. What are the obstacles to use previous agile practices?

ii. How to implement previous practices in mobile application?

Name:	
Company:	
Date: :	28/11/2017
Type of interview:	Single Interview
REF ID:	AT2
Case ID:	C3

First Section: General Demographic Information

1. What is your role in the team (manager, developer, etc.)?
   Project Manager
2. How many years of experience do you have?
   12 years
3. How many years of experience do you have in mobile application development?
   1.5 years
4. What type of business it is (software house, IT & telecom, etc.)?
   Software house
5. Is your software house is national/international?
   international
6. Which SW development process you use?
   Agile (scrum)
7. What are types of apps developed: entertainment, games, social, etc.? Enterprise/single user?
   Social app and business apps
   Enterprise
8. What are Platforms: iOS, Android? M-site/hybrid?
   IOS, Android
9. How many years of experience do you have in each type of application (native, m-sites and hybrid)?
   1.5 in iOS and android
10. What are Programming languages, Frameworks and IDEs used?
    Android Studio, Android, Java, Xcode, Swift, JavaScrip and Bugzilla for bug tracking, JUnit
11. How many team member working in a single project?
    We have 5 mobile application teams,5 members
12. What is the previous experience before working in the field of mobile application development?
    QA engineer and backend developer

RQ1) How do agile teams approach mobile application development?

1. What are software development main phases followed in the development of mobile applications from A-Z?
   i. What are the specific activities of each phase?
   ii. What development tools/ frameworks do you use (if any)?
   iii. For design (e.g., prototypes and mockup)?

The project was fixed cost with fixed time so often no change (we know the backlog and the features) so we divide the iteration to 12 iterations each one 2 weeks and we divided the feature between these feature. In each iteration we finished testing and development (manual testing).
Each bug we found we fixed it in the next iteration

Automation give you accuracy and fast and it useful to run automation testing in mobile because we always add features so we can run automation testing, but to build automation framework and resource you need to invest in time and resources but you can use it for long time but in our case the scope was limited so it isn't useful to waste QA resources in term of cost and time

We build full wireframe for the project it's important to be exist like this wireframe using UXPin and take the client approval on it and to make accurate time and cost estimation. then we divide these feature between iteration, then we give high priority for high risky features and we finish it and get feedback from the client on each feature after each iteration and user acceptance test, we used Bugzilla

There are stress because the release is short and when upload the release it could be rejected from sore so you should change some feature to make the application acceptable by Store

RQ2) What are the challenges and merits faced by these industrial teams compared with state-of-the-art?
RQ3) How can the agile practices be adjusted to best serve the mobile app development contexts?

1. If the Agile method is used:

i. Which of the following agile practices is used?

Principle	
Iterations and Increments	Each sprint is release about (2 weeks) because it small and simple feature And every two weeks there is a new release that will positively affect the users who find that there is an update every two weeks on the store and feel that the developers of this application are interested in encouraging them to use it
Internal and External Releases	No, it costly and needs time
Time Boxing	Two weeks
No Change of Started Projects	Yes,
Incremental Deliveries	Yes
On-site Customer	we was participate with product owner
Frequent Face-to-Face Interaction	Yes
Self-organizing Teams	yes
Empirical Process	Scrum
Sustainable Discipline	yes
Adaptive Planning	Yes, there are many requirement changes
Requirements Prioritization	Based on client needs
Fast Decision Making	Yes
Frequent Integration	Yes
Simplicity of Design	Yes
Refactoring	Code size is small so it less than other
Team Code Ownership	Yes

a. Is it tailored to suit mobile applications?

b. Are there differences in these practices between Native, Hybrid or M-site application? What are these differences?

c. Are there a difference in the agile practices between end user and enterprise application? What are these differences?

ii. Which one of agile practices that have been avoided?

a. Why it have been avoided?

b. Can be tailored to apply in mobile application development?

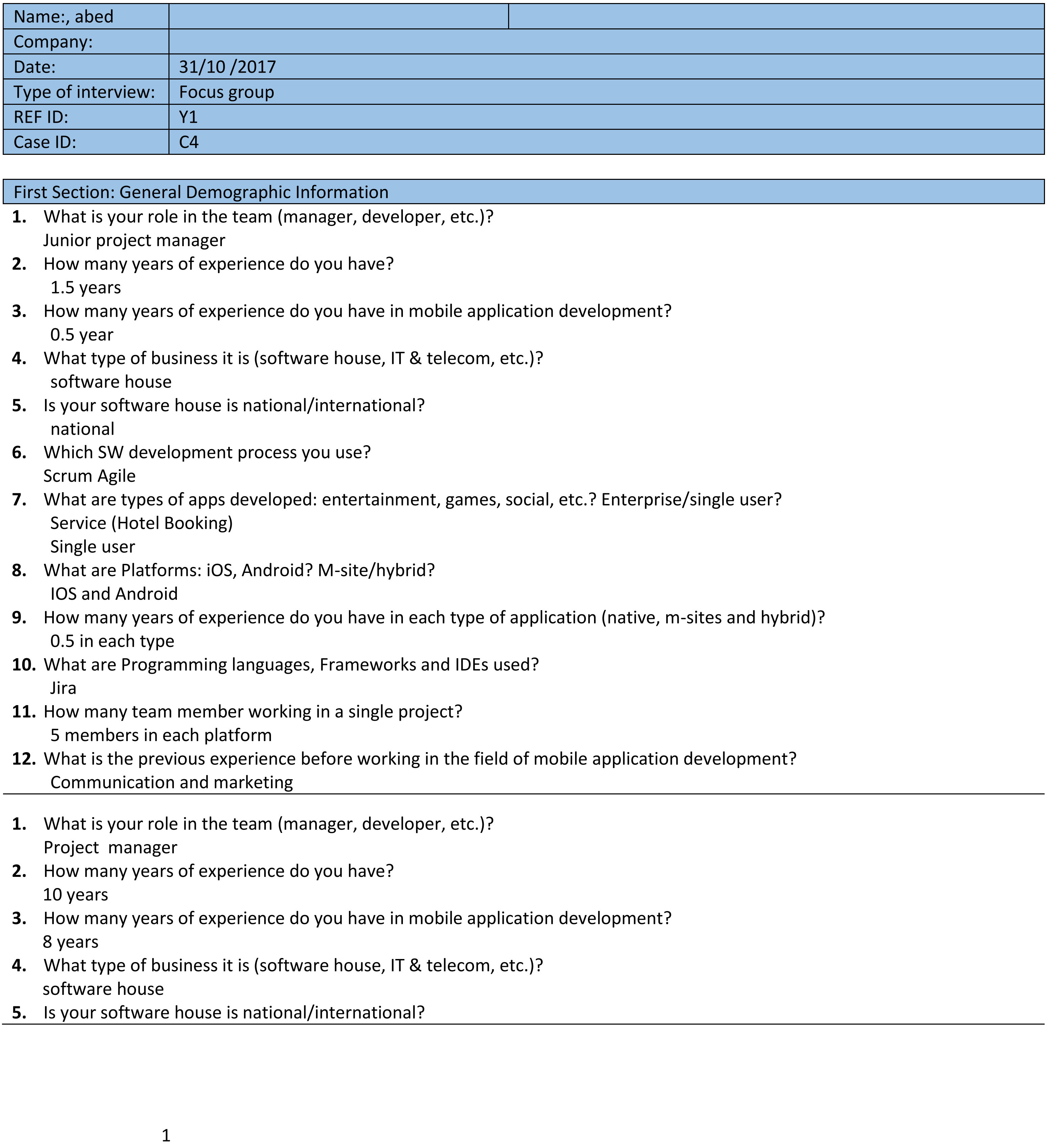
iii. Are there Non-Agile practices that have been added to the agile methods?

a. What are these practices? b. Why it have been added?

2. If the Agile method is not used:

i. What are the obstacles to use previous agile practices?

ii. How to implement previous practices in mobile application?



Name:, abed		
Company:		
Date:	31/10 /2017	
Type of interview:	Focus group	
REF ID:	Y1	
Case ID:	C4	

First Section: General Demographic Information

1. What is your role in the team (manager, developer, etc.)?
Junior project manager

2. How many years of experience do you have?
1.5 years

3. How many years of experience do you have in mobile application development?
0.5 year

4. What type of business it is (software house, IT & telecom, etc.)?
software house

5. Is your software house is national/international?
national

6. Which SW development process you use?
Scrum Agile

7. What are types of apps developed: entertainment, games, social, etc.? Enterprise/single user?
Service (Hotel Booking)
Single user

8. What are Platforms: iOS, Android? M-site/hybrid?
IOS and Android

9. How many years of experience do you have in each type of application (native, m-sites and hybrid)?
0.5 in each type

10. What are Programming languages, Frameworks and IDEs used?
Jira

11. How many team member working in a single project?
5 members in each platform

12. What is the previous experience before working in the field of mobile application development?
Communication and marketing

---

1. What is your role in the team (manager, developer, etc.)?
Project manager

2. How many years of experience do you have?
10 years

3. How many years of experience do you have in mobile application development?
8 years

4. What type of business it is (software house, IT & telecom, etc.)?
software house

5. Is your software house is national/international?

national

6. Which SW development process you use?
Scrum Agile

7. What are types of apps developed: entertainment, games, social, etc.? Enterprise/single user?
Service (hotel booking)
Single User

8. What are Platforms: iOS, Android? M-site/hybrid?
IOS, Android, Hybrid

9. How many years of experience do you have in each type of application (native, m-sites and hybrid)?
5 years Native, 2 years m-site and 2 years hybrid

10. What are Programming languages, Frameworks and IDEs used?
JavaScrip, Java, Objective C, Swift, Sublime, Android studio, Xcode, Phonegap, Jira

11. How many team member working in a single project?
Around 5 members for each platform

12. What is the previous experience before working in the field of mobile application development?
Web services and enterprise development

RQ1) How do agile teams approach mobile application development?

1. What are software development main phases followed in the development of mobile applications from A-Z?
i. What are the specific activities of each phase?
ii. What development tools/ frameworks do you use (if any)?
iii. For design (e.g., prototypes and mockup)?

Three type of requirement

1- Simple requirement (change text, image position) from company request by slack using feature request channel, the follow up question is asked for this feature to explain it, then it transformed into Jira
2- By email from product team based on checkout funnel (because this is an e-commerce project) so this type of requirement is reaction for data from client.
3- Strategy of company based on road map or company strategy.

There are design system (design pattern) used to cover the diversity of mobile devices, it was defined for mobile application

On Jira -> needs grooming, prioritize requirement -> grooming-> design ->development -> testing

Requirement gathering is different between Enterprise and end user, so prioritization is different, so you will listen to end user and they are responsible to find bugs and feature

More over the constraint (specially in term of time) in end user maybe less than enterprise, since in enterprise there is signed contract that force you to commitment on specific date

Sometimes you should focus in UX in enterprise, because even in enterprise project there are competitors maybe these competitors are indirect such as Facebook, Instagram, but these competitors Imposes competition on you, because your application UX and performance should be similar to these APPs

form factor (size screen) and different variation and flavor of the same platform is the main challenges in mobile application, so as a developer when you will be confident your feature is stable in different platform to give it to QA team. These challenges Look more clearly in QA there are many devices should be test for this app.

Our application follow feature parity so we always try to uniform management process and testing process (manual and automation)on two platform

There is stress especially in IOS, because when you finish new release you don't know when his release will deliver to the end user. Moreover, when new release is available on store, the adaption rate is low from user. And if there are critical bug is uploaded with app to store, so due to constraint on sore and adaption rate this make stress to developer and QA

RQ2) What are the challenges and merits faced by these industrial teams compared with state-of-the-art?
RQ3) How can the agile practices be adjusted to best serve the mobile app development contexts?

1. If the Agile method is used:
   i. Which of the following agile practices is used?

Principle	
Iterations and Increments	Yes,
Internal and External Releases	Yes, Beta virgin is given for QA and any employee in the company using beta distribution platform for mobile apps
Time Boxing	yes we apply time boxing, but we hope adopting release per feature
No Change of Started Projects	yes
Incremental Deliveries	yes
On-site Customer	• It rarely adopted because there are millions of users and millions of session that used by user, we follow AB testing by offering the same feature in different way for the users and dividing these version between the users, then choosing the feature is more accepted by the users, and enhance this version in future by make another two version from the original version • There are analytic system used to collect data about user behavior. So we emphasize on macro behavior which is related to checkout funnel • There are reporting issue section in the app enable the user to provide reports about bug, issues, and suggestion
Frequent Face-to-Face Interaction	yes
Self-organizing Teams	yes
Empirical Process	Scrums empirical process
Sustainable Discipline	yes
Adaptive Planning	yes
Requirements Prioritization	Based on business roadmap and objective we do it
Fast Decision Making	Yes, as it
Frequent Integration	Automation for API and mainly manual testing for mobile still there is no automation
Simplicity of Design	yes
Refactoring	yes
Team Code Ownership	yes

   a. Is it tailored to suit mobile applications?
   b. Are there differences in these practices between Native, Hybrid or M-site application? What are these differences?
   c. Are there a difference in the agile practices between end user and enterprise application? What are these differences?

   ii. Which one of agile practices that have been avoided?
   a. Why it have been avoided? b. Can be tailored to apply in mobile application development?

   iii. Are there Non-Agile practices that have been added to the agile methods?
   a. What are these practices? b. Why it have been added?

2. If the Agile method is not used:
   i. What are the obstacles to use previous agile practices?
   ii. How to implement previous practices in mobile application?

Name:	
Company:	
Date:	8/11/2017
Type of interview:	Single Interview
REF ID:	Y2
Case ID:	C4

First Section: General Demographic Information

1. What is your role in the team (manager, developer, etc.)?
   Project manager
2. How many years of experience do you have?
   9 years
3. How many years of experience do you have in mobile application development?
   3 years
4. What type of business it is (software house, IT & telecom, etc.)?
   Software house (service)
5. Is your software house is national/international?
   national
6. Which SW development process you use?
   Scrum Agile
   Single User
7. What are types of apps developed: entertainment, games, social, etc.? Enterprise/single user?
   Service (Hotel booking)
8. What are Platforms: iOS, Android? M-site/hybrid?
   IOS (swift), Android and Hybrid
9. How many years of experience do you have in each type of application (native, m-sites and hybrid)?
   3 years in native and 6 month hybrid
10. What are Programming languages, Frameworks and IDEs used?
   Java, JavaScrip, Objective C, Swift, Sublime, Xcode, Appcode,Eclipse, Andriod, Phonegap and React Native, Jira
11. How many team member working in a single project?
   5 developers in each teams (team for Android and Team for IOS)

4 What is the previous experience before working in the field of mobile application development?
   Backend development

RQ1) How do agile teams approach mobile application development?

1. What are software development main phases followed in the development of mobile applications from A-Z?
   i. What are the specific activities of each phase?
   ii. What development tools/ frameworks do you use (if any)?
   iii. For design (e.g., prototypes and mockup)?

In mobile there are challenges don't exist in other technologies (these was the reason why we start in hybrid at the beginning)
You build the same feature for two platforms (so you replica the resource)

The development isn’t smooth as other platform , because the tools don’t help you to make the development process fast as other platform as php for example, moreover compilation time is long so long waiting time this factor is effect specially when the project become complex (this waste a lot time for developers), so we compile the code on device specially in android to accelerate build and compilation process

At the beginning we make analysis to find out which devices and platform are most used in our target, the we make sure we have these devices to test the app on it, and we make sure once a new device will release we get it

Anyone in the company can input a new requirement ad there reasons for this requirement these requirement entered in pipeline for product management, then the project manager analysis and validate these requirement with the requester if the project manager approve on these requirement the measure the importance of these requirement for the product and how many time it needs and put descriptive description for it and make the design with the UX experts if it needs (sometimes there are many variation, then UX design and project manager try to reach to the best variation most often there are two form for the feature)for it then put it in the backlog, when sprint started they pick it (sprint take two weeks) sometimes we develop feature in two form if the UX supply more than one UI design

Then test it then we put the these feature online and divide it between users randomly, each feature have goals and based on conversion rate for the user we adopt one of feature variation then we keep enhancement on the picked feature sometimes we make more variations for the picked variation (using ab test) so the same feature is done in different type and always there are suggestion to enhance it based on the user conversion.

If the feature isn’t platform related we divide different feature between iOS and android user but if it platform related we divide variation between the same platform user

There are stress in mobile application development because if there are bug in custom version and user doesn’t update new version which is contain the fixed for this bug that mean you loss this user so we always try to add configuration out of mobile application to fix any bug quickly

We use hybrid in the beginning, mainly we used ionic as a proof of concept to test how the mobile app is important for our business

RQ2) What are the challenges and merits faced by these industrial teams compared with state-of-the-art?
RQ3) How can the agile practices be adjusted to best serve the mobile app development contexts?

1. If the Agile method is used:
   i. Which of the following agile practices is used?

Principle	
Iterations and Increments	Yes each two weeks we have release. Current each sprint is release but we hope do release per feature which means higher frequency for the releases, but we can’t do that before we reach 100% automation of testing so you should covering your platforms in automation testing which means you don’t need human interruption to make sure your app have bug or not I think automation is a ghost in mobile application development because the development process itself is not smooth. So you should innovate in using the automation mobile application development because many things are not yet ready. These are expensive under the large variation in mobile devices and platforms. There are continuous communication between feature requester, project manager, developer and QA to make sure the feature is developed and tested as expected After two week there are three or four days to make integration testing with bug fixing , some time we postpone the release to make sure we released correct version

	Because there are concerns if the version has bugs especially in iOS because there is a long review time and update rate less than android
Internal and External Releases	yes
Time Boxing	yes
No Change of Started Projects	Yes, because always we have clear road map
Incremental Deliveries	yes
On-site Customer	It is difficult to apply as in traditional technology, but we try to collect feedback from users using the questionnaires, but users are less interested in these types of questionnaires, so we use the AB testing
Frequent Face-to-Face Interaction	Yes we have every day stand up meeting with product manager
Self-organizing Teams	yes
Empirical Process	We use scrum empirical process
Sustainable Discipline	yes
Adaptive Planning	yes
Requirements Prioritization	Based on business strategy
Fast Decision Making	Yes
Frequent Integration	Yes
Simplicity of Design	yes
Refactoring	yes
Team Code Ownership	yes

a. Is it tailored to suit mobile applications?

b. Are there differences in these practices between Native, Hybrid or M-site application? What are these differences?

c. Are there a difference in the agile practices between end user and enterprise application? What are these differences?

ii. Which one of agile practices that have been avoided?

a. Why it have been avoided? b. Can be tailored to apply in mobile application development?

iii. Are there Non-Agile practices that have been added to the agile methods?

a. What are these practices? b. Why it have been added?

2. If the Agile method is not used:

i. What are the obstacles to use previous agile practices?

ii. How to implement previous practices in mobile application?

Appendix C

Data Analysis

This appendix presents the spreadsheets that used in data analysis for each case. by applying thematic coding.

RefID	ID	Sentence	Code1	Code2	Theme
ART1	31	So mobile application development is stress for the programmer because you should fix these bugs quickly.	mobile_app_development_stress		There is a stress in mobile application development because you need to solve the bugs as soon as possible in order to avoid loss of users. In addition, some store take a long time to add a new releas of application such as App store needs 4 days to
ART1	32	Moreover, some stores (such as AP store) need a long time (four days) to approve on a new release	mobile_app_development_stress		
ART1	21	Mobile product needs fast process to keep alive	mobile_app_development_challenges		In general, mobile applications need a fast software development process. That's because there is a diversity of users and there is a change in their needs, to achieve user satisfaction there needs to be a change and update of applications. Moreover, there are always new mobile devices so the application should be compatible with it.
ART1	22	Mobile always need change and update because there are diversity in users and there always changes in needs	mobile_app_development_challenges		
ART1	23	Moreover, there are always new devices need to be compatible with your application	mobile_app_development_challenges		
ART1	20	Application that need to be runnable in diverse platform but there is no sufficient resource we use hybrid for this application	hypride_application		We use hybrid apps when we want to develop an application that run on more than one platform and the company does not have enough resources to make a native applications
ART1	6	It is important to build our decision based on statistics. So we Search for statistics (almost not exist in Palestine) who are use mobile device? What are the kind of smartphone they use in term of platform and type of device? (There is no source for such statistics so we adopted or simple statistics).	development_process_requirment	challenges	Part of the new requirements will be according to our business plan, and an important part from the competitor to take advantage of the feature they offer, by offering similar feature or superiority over it, or to take advantage of the defects they have. we check the validity of new requirements against challenges that will be faced in business, culture and mobile technology . It is important to have statistics to know the most used machines and platforms.
ART1	1	our business plan is our source of our requiremnt	development_process_requirment	requirement_source	
ART1	2	Team members could be another source for our requiremnt	development_process_requirment	requirement_source	
ART1	3	Most of our requirement could be from competitors, so we should know who are our competitors? To understand what the existing. And how to benefit from their experience? Because the users use it and familiar with the existence application. Moreover we search for weaknesses in competitors such as complicated feature to provide this feature more simpler to excellence on what exists	development_process_requirment	requirement_source	
ART1	4	We decide whether we will accept the new requirement or not after check the validity of the idea versus it value and the challenges that will be faced in business, culture and mobile technology such as in Palestine there is no 3g.	development_process_requirment		
ART1	5	However, the user care about their business (how your app will satisfy their needs) not about your technology, So we thinking in the new culture that will introduce to the user and their resistance for the new idea.	development_process_requirment		
ART1	11	The features and UI of the feature are listed in the backlog using Bugzilla and It is given priority according to business plan and sometimes according to its importance compared to what the competitor offers.	development_process_prioritization	prioritization_criteria	The requirements are prioritized based on their value in the business plan and based on their value against what competitors are offering
ART1	12	At the beginning of the iteration, the feature is chosen based on its priority.	development_process_prioritization		
ART1	16	Sometimes if there is a need we redesign the feature based on the feedback we get.	development_process_maintenance		
ART1	18	In mobile game we don't take feedback directly from user, so we record video for user face and screen of game without interrupting to monitor the user reaction during the playing to modify the game level based on user reaction	development_process_feedabck	game_appilcation	We take feedback directly from users or by using tracking tool. The case is different in game application, we try to record video for the face of the player and the game to monitor the facial expressions of the player and modify the game based on user reactions
ART1	15	We take the feedback from the user (almost using tracking tools, in some time we have to contact directly with user to take feedback specially in begging of the application because there is no large number of user in some case we ask user to use our application and take the feedback Immediately).	development_process_feedabck		
ART1	14	The feature/features is developed and then tested manually throughout the iteration (two weeks). After every two weeks we have a release.	development_process_devlopment	testing_techniques	Features Test manually each itteration
ART1	13	The feature is broken down into subtasks and distributed among developers.	development_process_devlopment		
ART1	10	They take into consideration competitors' designs and when our application will be used (during driving, running or setting …) to customize our application to be suited with the user situation in term of interface for example if the application will be used during running the button should be large because the user will use the application without full concentration	development_process_design	challenges	We give the UX more attention than the technical issues, considering the design of the competitors and who and when will the application be used. The case is different in the enterprise applications because we focus on the functionality more than the design.
ART1	17	In Enterprise we focus more on functional meanwhile in single user we focus on UX and usability	development_process_design	enterprise_app	

ART1	19	Game is a different story because game applications have more challenges in term of Art, taste and physiology because in service application often there is a need for such application but in games often satisfy high needs so you should make him enjoy in the game and there is no matrix to measure the Enjoyment, so we need to identify who are the target audience carefully (in some case your target could be male or female from the age X to Y years).	development_process_design	game_appilcation	
ART1	8	Actually, The UX is more important than technical side to gain user satisfaction.	development_process_design	UX_importance	
ART1	7	We sketch the UI as a wireframe using UXPin tool.	development_process_design		
ART1	9	There is often a discussion between the UX specialist and product manager about the UI.	development_process_design		
ART1	33	in mobile is less than other to stay in competition, usually it two week release and one feature with small enhancement and bug fixing, But at the two-week release there is a high risk because it's difficult to avoid bugs due to rapid development.	agile_principles_time_boxing		In mobile shorter than other fields (2 weeks), we hope to get out of time boxing and reach release per feature to always compete
ART1	34	we hope to reach release per feature, mobile app feature almost need shorter time than other technology, so when you make release per featur that mean you always compete	agile_principles_time_boxing		
ART1	28	Sometimes there is a trade-off between IOS and Android, where do we need to implement feature first? You may decide to make a feature on a platform before the other because the number of users in this platform is more than the other.	agile_principles_requirements_prioritization		there is a trade off between platforms, so we decide to make a feature on a platform before the other because the number of users in this platform is more than the other.
ART1	29	User requirement (from feedback) some times	agile_principles_requirements_prioritization		
ART1	35	Refactoring in mobile is little compared to other applications because there is no time	agile_principles_refactoring		in mobile less than other fields because there is no time
ART1	27	No, because the plan is not clear	agile_principles_no_change_of_started_projects		
ART1	24	Yes, project divided into release based on list of feature	agile_principles_iterations_and_increments		In mobile are less than other applications (two weeks)
ART1	25	Some times in release that contain major changes (in DB or backend), you should to make sure that release that should work compatible with each other, so you should remain old release until new release become stable and all customer are satisfy with new release so you will drop new release	agile_principles_iterations_and_increments		
ART1	26	We do not apply this principle because the time between the releases is short and it is not allowed to apply this principle. It can be applied in case we agree with a group of users to use internal release before we deliver external release, but this can be costly	agile_principles_internal_and_external_releases	agile_principles_Iterations_and_increments	we not applied because need time
ART1	30	Yes, Sometimes we have to stop working on a particular feature to make campaign (new feature) not in plan as response for competitor new feature	agile_principles_no_change_of_started_projects		it is require to implement new feature as response for compettor

RefID	ID	Sentence	Code1	Code2	Theme
ET1	1	We work as outsourcing, so we take ready requirement with design and UX for each feature from client, Sometimes we intervene if the requirement or the UX is illogical or does not fit the current application to reach a logical solution with the client.	development_process_requirment	outsourcing_issue	requirements and UX is taken from client
ET2	12	We receive requirement with mockups	development_process_requirment	outsourcing_issue	
ET1	2	Each two weeks we make a release, we input the requirement in to Jira, each sprint we make breakdown for each story and make a planning and estimation	development_process_management		the Feature breakdown into tasks and each task take estimation
ET1	3	then we distribute it between developers after develop it we testing it and upload it on the store	development_process_management		
ET2	13	and break it down to task and put estimate on it and put it on Jira and	development_process_management		
ET1	4	Development in android for UI is faster than IOS because there are many library support that (UI is the half of feature in mobile)	development_process_devlopment	challenges	after development we do manual testing in the same iteration we could do automation testing if the client paid for that. However, android development is faster than IOS
ET2	14	the developer start develop on these requirement and then test it using manual test, if user need and paid we use automation for UI and feature	development_process_devlopment	testing_techniques	
ET2	18	Based on client plan	agile_principles_requirements_prioritization	outsourcing_issue	prioritization done based on client plan
ET1	10	Client request has high priority.	agile_principles_requirements_prioritization		
ET2	19	Often there are no factoring because there is time restriction so we follow dimple design from the beginning	agile_principles_refactoring		
ET2	17	Yes product owner always exist	agile_principles_on-site_customer		product owner particpate in planning
ET1	9	If client request an new feature during the release we stop working on the current feature and we start on new feature	agile_principles_no_change_of_started_projects	outsourcing_issue	This is based on client, he can change as he like
ET1	5	Each release take two weeks.	agile_principles_iterations_and_increment		each wo week we have release (after devlopemnt done, ther are two days for code freez and testing)
ET1	6	At end of each release there are two days for code freeze.	agile_principles_iterations_and_increment		
ET1	7	We make the same feature in IOS and Android if on feature doesn't ready on one platform we make release on another platform and doesn't make it in another platform	agile_principles_iterations_and_increment		
ET2	15	Yes 2 weeks for each sprint and release	agile_principles_iterations_and_increment		
ET1	8	Each one week we make beta release for internal testing and using	agile_principles_internal_and_external_releases		each week we give a special group an relese as internal release
ET2	16	Yes we have special group for internal release	agile_principles_internal_and_external_releases		
ET1	11	We use to testing technique , one for each feature if the feature is accepted the we integrate it with the project and make integration testing using manual testing	agile_principles_frequent_integration		we using manual testing for integration

RefID	ID	Sentence	Code1	Code2	Theme
AT2	23	There are stress because the release is short and when upload the release it could be rejected from sore so you should change some feature to make the application acceptable by Store	mobile_app_development_stress		The stress due to short release and store constraint
AT1	9	The hybrid application usually used for a proof of concept. Because of its low investment cost and it is cross platform so no need to build the same application twice, so when we need to take feedback from clients we use hybrid application, but it has bad performance	hypride_application		Hybrid application is used as a proof of concept to take quick feedback from users about specific feature
AT1	1	We start with scrum 0 (the first release take from 3 weeks to 1 month), we analysis the requirement and identification for the core feature in the project and ask client if there any unclear issue.	development_process_requirment		we analyze requirement from client and identify core feature and divide it to stories
AT1	2	After that, we take a look into the requirement and almost it clear and divide it into user story and explain it for the client, then we estimate how each sprint needs time, then we try to put time-frame for the project.	development_process_requirment		
AT2	22	then we divide these feature between iteration, then we give high priority for high risky features and we finish it and get feedback from the client on each feature after each iteration and user acceptance test, we used Bugzilla	development_process_prioritization	outsourcing_issue	The features divded between itteration, because the time and cost is excpected due o ther is asigned contract with enterprse
AT2	10	The project was fixed cost with fixed time so often no change (we know the backlog and the features) so we divide the iteration to 12 iterations each one 2 weeks and we divided the feature between these feature.	development_process_management	outsourcing_issue	
AT1	5	these feedback could be done in the next sprint or treat it as ad hook and fix it within current sprint if it was simple bug.	development_process_maintenance		the bug found by client fixed in next itteration
AT2	19	Each bug we found we fixed it in the next iteration	development_process_maintenance		
AT1	4	and the user acceptance test at the end of each sprint and we take the feedback from user	development_process_feedabck	testing_techniques	There are user acceptance test and sprint retrospective with client
AT1	6	At each sprint there are retrospective with client.	development_process_feedabck		
AT2	18	In each iteration we finished testing and development (manual testing).	development_process_devlopment	testing_techniques	each iteration take 2 weeks development and testing
AT1	3	Then we start development in releases and manage the backlog using Jira,	development_process_devlopment		
AT1	7	the first development release could take 1 month then the next releases should be 2 weeks	development_process_devlopment		
AT2	21	We build full wireframe for the project it’s important to be exist like this wireframe using UXPin and take the client approval on it and to make accurate time and cost estimation.	development_process_design	outsourcing_issue	UX is done for all app and take client approval for it
AT1	8	We used wireframe which sketched by UXPin for UI we add them in sprint planning	development_process_design		
AT2	20	Automation give you accuracy and fast and it useful to run automation testing in mobile because we always add features so we can run automation testing, but to build automation framework and resource you need to invest in time and resources but you can use it for long time but in our case the scope was limited so it isn’t useful to waste QA resources in term of cost and time	automation_testing		automation is imprtant because it fast an accurate. But it need resources and to invest in time
AT2	27	Two weeks	agile_principles_time_boxing		two weeks per itteration
AT2	29	Based on client needs	agile_principles_requirements_prioritization	outsourcing_issue	requirment prioritzation done based clients view
At1	15	Based on client	agile_principles_requirements_prioritization		
At1	17	Yes, but less than another platform	agile_principles_refactoring		it done but less than other platform because code size is small
AT2	30	Code size is small so it less than other	agile_principles_refactoring		
At1	14	Yes, in enterprise the product owner often exist in planning, but in single end user sometimes we use focus group with influencers and take feedback	agile_principles_on-site_customer		The product owner particpate in planing
AT2	28	we was participate with product owner	agile_principles_on-site_customer		
At1	13	This depend on client (because we outsourcing)	agile_principles_no_change_of_started_projects	outsourcing_issue	according to client
At1	11	2 weeks release because we need more regular feedback and the mobile app feature is simpler than other platforms	agile_principles_iterations_and_increment		the project divided into itteration each one two weeks, at the end of this itteration we relaese new version
AT2	24	Each sprint is release about (2 weeks) because it small and simple feature	agile_principles_iterations_and_increment		

AT2	25	And every two weeks there is a new release that will positively affect the users who find that there is an update every two weeks on the store and feel that the developers of this application are interested in encouraging them to use it	agile_principles_iterations_and_increment		
At1	12	Based on project, this may cost process	agile_principles_internal_and_external_releases	outsourcing_issue	make internal and external needs time and costlt
AT2	26	No, it costly and needs time	agile_principles_internal_and_external_releases		
At1	16	Yes, we use and manual testing and stress test	agile_principles_frequent_integration		we use manul tesing for integration and stress test

RefID	ID	Sentence	Code1	Code2	Theme
Y1	12	There is stress especially in IOS, because when you finish new release you don’t know when his release will deliver to the end user.	mobile_app_development_stress		There is a constraint from application stores (specially App store) on adding new release and adaption rate for new release from user is low this make stress to developer and QA, because there are concerns that a user will lose due to a bug in the application
Y1	13	Moreover, when new release is available on store, the adaption rate is low from user.	mobile_app_development_stress		
Y1	14	And if there are critical bug is uploaded with app to store, so due to constraint on sore and adaption rate this make stress to developer and QA	mobile_app_development_stress		
Y2	35	There are stress in mobile application development because if there are bug in custom version and user doesn’t update new version which is contain the fixed for this bug that mean you loss this user so we always try to add configuration out of mobile application to fix any bug quickly	mobile_app_development_stress		
Y2	22	In mobile there are challenges don’t exist in other technologies (these was the reason why we start in hybrid at the beginning)	mobile_app_development_challenges		We build the same feature twice for two different platforms in addition to the tools that doesn’t help to accelerate the development process such as compilation tools that need a long time, this makes mobile application development process is not smooth as others fields
Y2	23	You build the same feature for two platforms (so you replica the resource)	mobile_app_development_challenges		
Y2	24	The development isn’t smooth as other platform , because the tools don’t help you to make the development process fast as other platform as php for example.	mobile_app_development_challenges		
Y2	25	moreover compilation time is long so long waiting time this factor is effect specially when the project become complex (this waste a lot time for developers), so we compile the code on device specially in android to accelerate build and compilation process	mobile_app_development_challenges		
Y2	36	We use hybrid in the beginning, mainly we used ionic as a proof of concept to test how the mobile app is important for our business	hypride_application		Hybrid was used in POC to test how the mobile app is important in our field
Y1	7	More over the constraint (specially in term of time) in end user maybe less than enterprise, since in enterprise there is signed contract that force you to commitment on specific date	enterprise_app		There are time constraints, especially in enterprise applications because you are a site with a contract enterprise that should commit you to time or you will lose the contract. Moreover, you should be interested in UX because you have competitors such as social networking applications and other famous applications. If you do not take this into consideration, there will be no acceptance of the application from the users and thus you will lose your contract with the company.
Y1	8	Sometimes you should focus in UX in enterprise, because even in enterprise project there are competitors maybe these competitors are indirect such as Facebook, Instagram, but these competitors Imposes competition on you, because your application UX and performance should be similar to these APPs	enterprise_app		
Y1	6	Requirement gathering is different between Enterprise and end user, so prioritization is different, so you will listen to end user and they are responsible to find bugs and feature	development_process_requirment	challenges	There is a road map from which to derive the application requirements. Any employee can add a new requirement, and the user can provide us with new requirements either via e-mail or through a special form. It is important to find statistics about platforms, devices and age of users
Y2	26	At the beginning we make analysis to find out which devices and platform are most used in our target, so we make sure we have these devices to test the app on it, and we make sure once a new device will release we get it	development_process_requirment	challenges	
Y1	1	requirement could be Simple (change text, image position) from company request by slack using feature request channel, the follow up question is asked for this feature to explain it, then it transformed into Jira	development_process_requirment	requirement_source	
Y1	2	requirement could be by email from product team based on checkout funnel (because this is an e-commerce project) so this type of requirement is reaction for data from client.	development_process_requirment	requirement_source	
Y1	3	main requirement from Strategy of company based on road map or company strategy.	development_process_requirment	requirement_source	
Y2	27	Anyone in the company can input a new requirement ad there reasons for this requirement these requirement entered in pipeline for product management,	development_process_requirment	requirement_source	
Y1	11	Our application follow feature parity so we always try to uniform management process and testing process (manual and automation)on two platform	development_process_management	challenges	we follow feature parity so we uniform management process and testing process on two platform
Y1	5	put feature On Jira -> needs grooming, prioritize requirement -> grooming-> design ->development -> testing	development_process_management		
Y2	28	then the project manager analysis and validate these requirement with the requester if the project manager approve on these requirement the measure the importance of these requirement for the product and how many time it needs and put descriptive description for it	development_process_management		
Y2	30	then put it in the backlog, when sprint started they pick it (sprint take two weeks)	development_process_management		
Y2	33	then we keep enhancement on the picked feature sometimes we make more variations for the picked variation (using ab test) so the same feature is done in different type and always there are suggestion to enhance it based on the user conversion.	development_process_maintenance		When optimizing a feature we can develop more variations and based on the results of AB testing we adopt the form that was the most popular

Y2	32	Then test it then we put the these feature online and divide it between users randomly, each feature have goals and based on conversion rate for the user we adopt one of feature variation	development_process_feedabck	challenges	AB testing mainly adopted to getfeedback from users
Y2	34	If the feature isn't platform related we divide different feature between iOS and android user but if it platform related we divide variation between the same platform user	development_process_feedabck	challenges	
Y2	31	sometimes we develop feature in two form if the UX supply more than one UI design	development_process_development	challenges	The feature could bed develop in different variations
Y1	10	These challenges Look more clearly in QA there are many devices should be test for this app.	development_process_development	Testing_challenges	
Y1	9	form factor (size screen) and different variation and flavor of the same platform is the main challenges in mobile application, so as a developer when you will be confident your feature is stable in different platform to give it to QA team.	development_process_development		
Y1	4	There are design system (design pattern) used to cover the diversity of mobile devices, it was defined for mobile application	development_process_design		we may make different UX variation, we have design pattern for UX issues
Y2	29	make the design with the UX experts if it needs (sometimes there are many variation, then UX design and project manager try to reach to the best variation most often there are two form for the feature)for it	development_process_design		
Y2	38	I think automation is a ghost in mobile application development because the development process itself is not smooth.	automation_testing	mobile_app_development_challenges	Automation testing needs innovate because many thing not ready which is expensice due tolarge variation in mobile devices and platform
Y2	39	So you should innovate in using the automation mobile application development because many things are not yet ready.	automation_testing	mobile_app_development_challenges	
Y2	40	This cause High cost under the large variation in mobile devices and platforms.	automation_testing	mobile_app_development_challenge	
Y1	16	yes we apply time boxing, but we hope adopting release per feature	agile_principles_time_boxing		
Y1	20	Based on business roadmap and objective we do it	agile_principles_requirements_prioritization		priortization is based on business strategy
Y2	45	Based on business strategy	agile_principles_requirements_prioritization		
Y1	17	It rarely adopted because there are millions of users and millions of session that used by user, we follow AB testing by offering the same feature in different way for the users and dividing these version between the users, then choosing the feature is more accepted by the users, and enhance this version in future by make another two version from the original version	agile_principles_on-site_customer	mobile_app_development_challenges	we adopted Ab testing mainly, user may participate and give feedback by reporting section
Y1	18	There are analytic system used to collect data about user behavior. So we emphasize on macro behavior which is related to checkout funnel	agile_principles_on-site_customer		
Y1	19	There are reporting issue section in the app enable the user to provide reports about bug, issues, and suggestion	agile_principles_on-site_customer		
Y2	44	It is difficult to apply as in traditional technology, but we try to collect feedback from users using the questionnaires, but users are less interested in these types of questionnaires, so we use the AB testing	agile_principles_on-site_customer		
Y2	43	Yes, because always we have clear road map	agile_principles_no_change_of_started_projects		they commit to roadmap so each itteration ther are feature should be finished to achieve goal
Y2	37	Yes each two weeks we have release. Current each sprint is release but we hope do release per feature which means higher frequency for the releases, but we can't do that before we reach 100% automation of testing so you should covering your platforms in automation testing which means you don't need human interruption to make sure your app have bug or not	agile_principles_iterations_and_increment	mobile_app_development_challenges	each itteration tow week, we hope do release per feature which means higher frequency for the releases, but we can't do that before we reach 100% automation of testing
Y2	41	There are continuous communication between feature requester, project manager, developer and QA to make sure the feature is developed and tested as expected.	agile_principles_iterations_and_increment		
Y2	42	After two week there are three or four days to make integration testing with bug fixing , some time we postpone the release to make sure we released correct version Because there are concerns if the version has bugs especially in iOS because there is a long review time and update rate less than android	agile_principles_iterations_and_increment		
Y1	15	Yes, Beta virgin is given for QA and any employee in the company using beta distribution platform for mobile apps	agile_principles_internal_and_external_releases		we give beta version for our employee
Y1	21	Automation for API and mainly manual testing for mobile still there is no automation	agile_principles_frequent_integration	mobile_app_development_challenge	using manula atomation for integration tes

.

References

[1] Tore Dybå and Torgeir Dingsøyr. “Empirical studies of agile software development: A systematic review”. In: *Information and software technology* 50.9 (2008), pp. 833–859.

[2] Juyun Cho. “A hybrid software development method for large-scale projects: rational unified process with scrum”. In: *Journal of Issues in Information Systems* 5.2 (2009), pp. 340–348.

[3] Adel Hamdan Mohammad, Tariq Alwada’n, et al. “Agile Software Methodologies: Strength and Weakness”. In: *International Journal of Engineering Science and Technology* 5.3 (2013), p. 455.

[4] Shakira Banu Kaleel and Ssowjanya Harishankar. “Applying agile methodology in mobile software engineering: Android application development and its challenges”. In: *Computer Science Technical Reports* (2013).

[5] Josh Dehlinger and Jeremy Dixon. “Mobile application software engineering: Challenges and research directions”. In: *Workshop on Mobile Software Engineering*. Vol. 2. 2011, pp. 29–32.

[6] Christelle Scharff and Ravi Verma. “Scrum to support mobile application development projects in a just-in-time learning context”. In: *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering.* ACM. 2010, pp. 25–31.

[7] Antti Oulasvirta, Mikael Wahlström, and K Anders Ericsson. “What does it mean to be good at using a mobile device? An investigation of three levels of experience and skill”. In: *International journal of human-computer studies* 69.3 (2011), pp. 155–169.

[8] Luis Corral, Alberto Sillitti, and Giancarlo Succi. “Software development processes for mobile systems: Is agile really taking over the business?” In: *Engineering of Mobile-Enabled Systems (MOBS), 2013 1st International Workshop on the.* IEEE. 2013, pp. 19–24.

[9] Anthony I Wasserman. “Software engineering issues for mobile application development”. In: *Proceedings of the FSE/SDP workshop on Future of software engineering research.* ACM. 2010, pp. 397–400.

[10] Harleen K Flora, Xiaofeng Wang, and Swati V Chande. “An investigation into mobile application development processes: Challenges and best practices”. In: *International Journal of Modern Education and Computer Science* 6.6 (2014), p. 1.

[11] Zhifang Liu, Xiaopeng Gao, and Xiang Long. “Adaptive random testing of mobile application”. In: *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on.* Vol. 2. IEEE. 2010, pp. V2–297.

[12] Luis Corral, Alberto Sillitti, and Giancarlo Succi. “Agile software development processes for mobile systems: Accomplishment, evidence and evolution”. In: *International Conference on Mobile Web and Information Systems*. Springer. 2013, pp. 90–106.

[13] Harleen K Flora and Swati V Chande. “A REVIEW AND ANAYSIS ON MOBILE APPLICATION DEVELOPMENT PROCESSES USING AGILE METHODLOGIES”. In: *International Journal of Research in Computer Science* 3.4 (2013), p. 9.

[14] Pekka Abrahamsson et al. “New directions on agile methods: a comparative analysis”. In: *Software Engineering, 2003. Proceedings. 25th International Conference on.* Ieee. 2003, pp. 244–254.

[15] Anitha Ashishdeep, Jitendra Bhatia, and Krunal Varma. “Software Process Models for Mobile Application Development: A Review”. In: *Computer Science and Electronic Journal* 7.1 (2016), pp. 150–153.

[16] Robert Holler. “Mobile application development: a natural fit with agile methodologies”. In: *VerisonOne LLC, Alpharetta* (2006).

[17] Kai Petersen and Claes Wohlin. “A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case”. In: *Journal of systems and software* 82.9 (2009), pp. 1479–1490.

[18] B. Algozzine and D. Hancock. *Doing Case Study Research: A Practical Guide for Beginning Researchers, Third Edition.* Teachers College Press, Teachers College, Columbia University, 2016. ISBN: 9780807758137.

[19] Jill Jesson and Fiona Lacey. "How to do (or not to do) a critical literature review". In: *Pharmacy education* 6 (2006).

[20] Kent Beck et al. "Manifesto for agile software development". In: (2001).

[21] Steven L Goldman. *Agile competitors and virtual organizations: strategies for enriching the customer*. Van Nostrand Reinhold Company, 1995.

[22] Zulkefli Mansor et al. "Success Determinants in Agile Software Development Methodology". In: *Procedia Engineering* (2011).

[23] Jiangping Wan and Ruoting Wang. "Empirical research on critical success factors of agile software process improvement". In: *Journal of Software Engineering and Applications* 3.12 (2010), p. 1131.

[24] Claudia Melo et al. "Agile team perceptions of productivity factors". In: *Agile Conference (AGILE), 2011*. IEEE. 2011, pp. 57–66.

[25] Ani Liza Asnawi, Andrew M Gravell, and Gary B Wills. "Empirical investigation on agile methods usage: issues identified from early adopters in Malaysia". In: *Agile Processes in Software Engineering and Extreme Programming* (2011), pp. 192–207.

[26] Akhil Kumar and Bindu Goel. "Factors Influencing Agile Practices: A Survey". In: *International Journal of Engineering Research and Applications* 2.4 (2012), pp. 1347–1352.

[27] Samia Abdalhamid and Alok Mishra. "Factors in Agile Methods Adoption". In: (2017).

[28] LEOR Vijayasarathy and Dan Turk. “Agile software development: A survey of early adopters”. In: *Journal of Information Technology Management* 19.2 (2008), pp. 1–8.

[29] Tsun Chow and Dac-Buu Cao. “A survey study of critical success factors in agile software projects”. In: *Journal of systems and software* 81.6 (2008), pp. 961–971.

[30] Harleen K Flora, Swati V Chande, and Xiaofeng Wang. “Adopting an agile approach for the development of mobile applications”. In: *International Journal of Computer Applications* 94.17 (2014).

[31] Pekka Abrahamsson. “Keynote: Mobile software development–the business opportunity of today”. In: *proceedings of the International Conference on Software Development.* 2005, pp. 20–23.

[32] Alan R Santos et al. “Investigating the Adoption of Agile Practices in Mobile Application Development.” In: *ICEIS (1).* 2016, pp. 490–497.

[33] Samer Zein, Norsaremah Salleh, and John Grundy. “Mobile application testing in industrial contexts: an exploratory multiple case-study”. In: *International Conference on Intelligent Software Methodologies, Tools, and Techniques.* Springer. 2015, pp. 30–41.

[34] Pekka Abrahamsson et al. “Mobile-D: an agile approach for mobile application development”. In: *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications.* ACM. 2004, pp. 174–175.

[35] Yang-Jae Jeong, Ji-Hyeon Lee, and Gyu-Sang Shin. “Development process of mobile application SW based on agile methodology”. In:

Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on. Vol. 1. IEEE. 2008, pp. 362–366.

[36] Vahid Rahimian and Raman Ramsin. "Designing an agile methodology for mobile software development: A hybrid method engineering approach". In: *Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on.* IEEE. 2008, pp. 337–342.

[37] Thiago Ferraz V da Cunha, Valeria LL Dantas, and Rossana MC Andrade. "SLeSS: A Scrum and Lean Six Sigma integration approach for the development of sofware customization for mobile phones". In: *Software Engineering (SBES), 2011 25th Brazilian Symposium on.* IEEE. 2011, pp. 283–292.

[38] N Ajit Kumar, KT Hari Krishna, and R Manjula. "Challenges and Best Practices in Mobile Application Development". In: *Imperial Journal of Interdisciplinary Research* 2.12 (2016).

[39] Chenkai Guo et al. "An automated testing approach for inter-application security in Android". In: *Proceedings of the 9th International Workshop on Automation of Software Test.* ACM. 2014, pp. 8–14.

[40] Anureet Kaur and Kulwant Kaur. "Suitability of Existing Software Development Life Cycle (SDLC) in Context of Mobile Application Development Life Cycle (MADLC)". In: *International Journal of Computer Applications* 116.19 (2015).

[41] Suhas Bhosale and Monali Bhosale. “Architectural considerations while developing Enterprise Mobile Applications”. In: *International Journal of Scientific and Research Publications* (2014).

[42] Robert K Yin. *Case study research: Design and methods.* Sage publications, 2013.

[43] Sarah Lewis. “Qualitative inquiry and research design: Choosing among five approaches”. In: *Health promotion practice* 16.4 (2017), pp. 473–475.

[44] Per Runeson and Martin Höst. “Guidelines for conducting and reporting case study research in software engineering”. In: *Empirical software engineering* 14.2 (2009), p. 131.

[45] June M Verner et al. “Guidelines for industrially-based multiple case studies in software engineering”. In: *Research Challenges in Information Science, 2009. RCIS 2009. Third International Conference on.* IEEE. 2009, pp. 313–324.

[46] Victoria Clarke and Virginia Braun. “Teaching thematic analysis: Overcoming challenges and developing strategies for effective learning”. In: *The psychologist* 26.2 (2013), pp. 120–123.

[47] Robert K Merton. “Thematic analysis in science: notes on Holton’s concept”. In: *Science* 188.4186 (1975), pp. 335–338.

[48] Colin Robson and Kieran McCartan. *Real world research : a resource for users of social research methods in applied settings.* Fourth edition. Chichester, West Sussex : Wiley, 2016. ISBN: 9781118745236.

[49] Daniela S Cruzes and Tore Dyba. “Recommended steps for thematic synthesis in software engineering”. In: *Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on*. IEEE. 2011, pp. 275–284.

[50] Thomas A Kappel. “Perspectives on roadmaps: how organizations talk about the future”. In: *Journal of Product Innovation Management* 18.1 (2001), pp. 39–50.

[51] Ronald N Kostoff and Robert R Schaller. “Science and technology roadmaps”. In: *IEEE Transactions on engineering management* 48.2 (2001), pp. 132–143.

[52] Tanja Suomalainen, Raija Kuusela, and Maarit Tihinen. “Continuous planning: an important aspect of agile and lean development”. In: *International Journal of Agile Systems and Management* 8.2 (2015), pp. 132–162.

[53] Maximilian Speicher, Andreas Both, and Martin Gaedke. “Ensuring web interface quality through usability-based split testing”. In: *International Conference on Web Engineering*. Springer. 2014, pp. 93–110.

[54] Javier Cámara and Alfred Kobsa. “Facilitating Controlled Tests of Website Design Changes Using Aspect-Oriented Software Development and Software Product Lines”. In: *Transactions on Large-Scale Data-and Knowledge-Centered Systems I*. Springer, 2009, pp. 116–135.

[55] Thomas Crook et al. “Seven pitfalls to avoid when running controlled experiments on the web”. In: *Proceedings of the 15th ACM SIGKDD*

international conference on Knowledge discovery and data mining. ACM. 2009, pp. 1105–1114.

[56] Sheetal Sharma, Darothi Sarkar, and Divya Gupta. “Agile processes and methodologies: A conceptual study”. In: *International journal on computer science and Engineering* 4.5 (2012), p. 892.

[57] Noura Abbas, Andrew M Gravell, and Gary B Wills. “Historical roots of agile methods: Where did “Agile thinking” come from?” In: *International Conference on Agile Processes and Extreme Programming in Software Engineering.* Springer. 2008, pp. 94–103.

[58] Kai Petersen and Claes Wohlin. “Issues and advantages of using agile and incremental practices”. In: *Software Engineering Research and Practice.* 2008.

[59] Kai Petersen and Claes Wohlin. “The effect of moving from a plan-driven to an incremental software development approach with agile practices”. In: *Empirical Software Engineering* 15.6 (2010), pp. 654–693.

[60] Dean Leffingwell and Dave Muirhead. “Tactical Management of Agile Development: Achieving Competitive Advantage”. In: *Rally Software Development Corporation* (2004), pp. 1–23.

[61] Harleen K Flora and Swati V Chande. “A systematic study on agile software development methodologies and practices”. In: *International Journal of Computer Science and Information Technologies* 5.3 (2014), pp. 3626–3637.

[62] Zhang Xihui. “Software development methodologies trends and implications”. In: *Adopting the IS 2009 model curriculum: A panel session to Address the Challenges for Program Implementation: 173, 2009. Engineering & Remote Sensing* 71.4 (2005), pp. 415–424.

[63] Lan Cao et al. “A framework for adapting agile development methodologies”. In: *European Journal of Information Systems* 18.4 (2009), pp. 332–343.

[64] Kent Beck. *Extreme programming explained: embrace change.* addison-wesley professional, 2000.

[65] Anant Mishra, Kingshuk K Sinha, and Sriram Thirumalai. “Execution Capabilities of Distributed Technology Projects: A Comparative Evaluation of Efficiency Using Stochastic Frontier Analysis”. In: (2009).

[66] Richard A Guzzo and Marcus W Dickson. “Teams in organizations: Recent research on performance and effectiveness”. In: *Annual review of psychology* 47.1 (1996), pp. 307–338.

[67] Jim Highsmith. *Agile project management: creating innovative products.* Pearson Education, 2009.

[68] Mark Fenton-O’Creevy. “Employee involvement and the middle manager: evidence from a survey of organizations”. In: *Journal of Organizational Behavior* (1998), pp. 67–84.

[69] Victor Szalvay. “An introduction to agile software development”. In: *Danube technologies* 3 (2004).

[70] MA Awad. “A comparison between agile and traditional software development methodologies”. In: *University of Western Australia* (2005).

[71] Barry Boehm and Richard Turner. “Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods”. In: *Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on.* IEEE. 2004, pp. 718–719.

[72] Jens Norin and Daniel Karlström. “Lean configuration management”. In: *Softhouse Consulting, Stormgatan* 14 (2006).

[73] Ahmed Nagy, Mercy Njima, and Lusine Mkrtchyan. “A bayesian based method for agile software development release planning and project health monitoring”. In: *Intelligent Networking and Collaborative Systems (INCOS), 2010 2nd International Conference on.* IEEE. 2010, pp. 192–199.

[74] Anna Perini, Filippo Ricca, and Angelo Susi. “Tool-supported requirements prioritization: Comparing the AHP and CBRank methods”. In: *Information and Software Technology* 51.6 (2009), pp. 1021–1032.

[75] Lan Cao and Balasubramaniam Ramesh. “Agile requirements engineering practices: An empirical study”. In: *IEEE software* 25.1 (2008).

[76] Zornitza Racheva, Maya Daneva, and Luigi Buglione. “Supporting the dynamic reprioritization of requirements in agile development of software products”. In: *Software Product Management, 2008. IWSPM’08. Second International Workshop on.* IEEE. 2008, pp. 49–58.

[77] Andrea De Lucia and Abdallah Qusef. “Requirements engineering in agile software development”. In: *Journal of Emerging Technologies in Web Intelligence* 2.3 (2010), pp. 212–220.

[78] Feiyi Su and Linkai Zhu. “An empirical study of the Decision-making process in Agile software development based on industries from China”. In: (2015).

[79] Ion Ivan and Mihai Liviu Despa. “Statistical indicators for measuring innovation in IT project management”. In: *Procedia Economics and Finance* 10 (2014), pp. 167–177.

[80] Helena Holmström Olsson, Hiva Alahyari, and Jan Bosch. “Climbing the” Stairway to Heaven”–A Mulitiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software”. In: *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on*. IEEE. 2012, pp. 392–399.

[81] Brian Fitzgerald, Gerard Hartnett, and Kieran Conboy. “Customising agile methods to software practices at Intel Shannon”. In: *European Journal of Information Systems* 15.2 (2006), pp. 200–213.

[82] Frank Maurer and Sebastien Martel. “Extreme programming. Rapid development for Web-based applications”. In: *IEEE Internet computing* 6.1 (2002), pp. 86–90.

[83] Martin Fowler and Kent Beck. *Refactoring: improving the design of existing code*. Addison-Wesley Professional, 1999.

[84] Raimund Moser et al. “A case study on the impact of refactoring on quality and productivity in an agile team”. In: *Balancing Agility and Formalism in Software Engineering*. Springer, 2008, pp. 252–266.

[85] Mikael Lindvall et al. “Empirical findings in agile methods”. In: *Conference on extreme programming and agile methods*. Springer. 2002, pp. 197–207.

[86] Todd Sedano, Paul Ralph, and Cécile Péraire. “Practice and perception of team code ownership”. In: *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*. ACM. 2016, p. 36.

[87] Florian Lettner, Clemens Holzmann, and Patrick Hutflesz. “Enabling a/b testing of native mobile applications by remote user interface exchange”. In: *International Conference on Computer Aided Systems Theory*. Springer. 2013, pp. 458–466.

[88] Mubbashir Nazir, Umar Mir, and Anuj Yadav. “A Mechanism for Mining Top Impacted Segments in Web Analytics based on the Variants used for AB Testing”. In: (2014).

[89] Giordano Tamburrelli and Alessandro Margara. “Towards automated A/B testing”. In: *International Symposium on Search Based Software Engineering*. Springer. 2014, pp. 184–198.