Exploring Agile Mobile App Development in Industrial Contexts: A Qualitative Study

Ali Asfour[a], Samer Zain[b], Norsaremah Salleh[c], John Grundy[d]

[a] *Master in Software Engineering, Birzeit University, Palestine*

[b] *Department of Computer Science, Birzeit University, Palestine*

[c] *Department of Computer Science, International Islamic University Malaysia, Malaysia*

[d] *Faculty of Information Technology, Monash University, Australia*

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, feedback via app stores, 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 applications, there is a lack of studies of how mobile app development teams can adopt agile methods and the challenges they are facing. Our study explores this area to better understand how some representative industrial teams approach agile mobile app development, and the challenges they are facing. We conducted a qualitative study involving four different mobile app development companies. From our findings we argue that not all agile development principals are necessarily applicable within the mobile app development context. Furthermore, mobile app development teams face additional challenges when adopting agile methods such as development automation tools and on-line app store restrictions.

Keywords: Mobile Apps, Mobile App Development, Agile Methods, Case Study

1. Introduction

Mobile devices, also known as smart phones and tablets, are being used by more and more people for business and personal tasks. The increasing popularity and adoption of mobile devices in people's lives is due to their ease of use; portability; (Kaur and Kaur, 2015); evolution of their computational power; and rich mobile application (app) markets (Corral et al., 2013b). Furthermore, the use of mobile apps is no longer limited to the entertainment purposes, but they have now become a central player in many critical domains such as health, education, finance, and marketing (Scharff and Verma, 2010). Consequently, mobile app development has become a main target for researchers and software industry practitioners.

Email addresses: ali.f.asfour@gmail.com (Ali Asfour), szain@birzeit.edu (Samer Zain), norsaremah@iium.edu.my (Norsaremah Salleh), john.grundy@monash.edu (John Grundy)

Previously, most of mobile apps were small or medium in size and scope and targeted entertainment sectors. Thus, the development teams typically consisted of one or two members at most. Therefore, the development processes of mobile apps were not given much attention by researchers and followed a limited number of best software development practices (Kaleel and Harishankar, 2013). Nonetheless, the current proliferation of mobile apps requires specific mobile development methods to produce a high quality and reliable apps. This purpose can be achieved by adopting recent standards of software development methodologies in the app development life cycle process (Kaleel and Harishankar, 2013).

Mobile app development is in many ways much more complex and fault prone than traditional web and desktop applications (Liu et al., 2010). For example, mobile apps' UI (User Interface) and UX (User Experience) provide new features for user interaction, which have not been previously explored in depth in research compared to desktop and web applications (Scharff and Verma, 2010; Oulasvirta et al., 2011). In addition, the fragmentation of mobile platforms and device hardware adds a new dimension of complexity since each platform has its own different characteristics and constraints (Corral et al., 2013b; Wasserman, 2010). Moreover, mobile apps in many sectors need to be developed quickly, as the mobile app industry has short time to market requirements and strong competition (Flora et al., 2014b). Traditional software engineering approaches cannot cope up with these new challenges in the mobile app development industry (Dehlinger and Dixon, 2011). This is because mobile industry outpaced the traditional software engineering approaches (Dehlinger and Dixon, 2011). The literature proposes different software development methodologies dedicated for mobile app development (Corral et al., 2013a).

Agile software development processes and methods have caught the attention of researchers and software engineers around the world and many studies consider the agile method as a natural fit for the mobile app industry (Flora and Chande, 2013). However, researchers recommend that the development process should be tailored to better suit mobile apps' peculiarities (Flora and Chande, 2013). In many cases, researchers have addressed some of the shortcomings of agile methods by merging the practices of agile software development with practices from other software development frameworks (Corral et al., 2013b).

Although there are several agile software engineering methods intended specifically for mobile app development (Corral et al., 2013a), they still have several limitations (Flora et al., 2014b). Furthermore, to our best knowledge, there is no study yet available that has focused on identifying actual advantages and disadvantages or challenges when adopting agile development methods in mobile app development within real-world industrial contexts. Therefore, we have conducted a qualitative study based on multiple-case-study (four different industrial case) approach. Each case represents a software development company in Palestine in the field of mobile app development. The aim of the study is to explore and investigate how industrial teams adopt agile methods in the real-world contexts.

In consequence, we formed the following key objectives for this study:

- provide an understanding of issues and challenges faced by mobile app developers in the real word;
- understand the ways being used to implement agile and incremental development practices for mobile app development in industrial contexts;
- provide an in-depth understanding of the strengths, weaknesses and outstanding issues related to agile mobile app development from real-world usage; and
- understand how existing agile methods might be better tailored for mobile app development contexts.

Such an investigation can reveal implicit challenges and needs of real industrial teams. Moreover, we have compared what is currently being applied in the industrial practice with state-of-the-art to help developers in solving adoption issues that they are facing. Our results suggest that not all agile principles are suitable

for mobile app development contexts and that industrial teams face additional challenges when adopting agile methods. Furthermore, the automation development tools available for mobile app developers have significant drawbacks and the on-line app stores' restrictions add more burden on the development teams.

The rest of this paper is organized as follows. Section two provides a background of a number of mobile development challenges and classification of mobile apps based on the type of services that the app provides. Section three presents and reviews some major related works. Section four describes the methodology that we have applied for our data collection and analysis. Section five presents our study data analysis results, followed by a detailed discussion of these results in section six. Finally, section seven presents recommendations to improve the agile mobile app development process and section eight concludes the paper and outlines key areas for future research.

2. Mobile App Development Challenges

Unlike the development of traditional web and desktop applications, mobile app development poses a number of additional challenges. Some challenges are associated with the rapidly changing business requirements (Kaleel and Harishankar, 2013). Other challenges are associated with platform and hardware fragmentation, user experience (UX), expertise of software developers, security, and tight schedule and budget (Corral et al., 2013b; Flora et al., 2014a; Flora and Chande, 2013). The state-of-the-art for mobile app development challenges can be summarized as follows:

Platform Fragmentation: There are several mobile operating systems such as Android and iOS, and each with many versions in use, each requiring different specifications, hardware, tools, programming languages, and IDEs (Integrated Development Environments) in order to build apps for each platform (Flora et al., 2014b; Santos et al., 2016; Kumar et al., 2016). Consequently, if a native mobile app is required to run on both Android and iPhone devices; two teams of developers usually need to be formed with each team targeting one specific platform. (Flora et al., 2014b; Kumar et al., 2016).

Hardware Fragmentation: Mobile hardware frequently changes in terms of memory, speed, graphics processing, processing capability, sensors, inadequate energy supply, OS version updates, and computational power (Flora et al., 2014b; Kumar et al., 2016). This can be very challenging because some hardware features required by the app are not supported equally by the targeted mobile devices nor indeed different verions of the same OS. In this case the developers have to exercise more effort to make their app dynamic in terms of acquiring hardware resources (Kumar et al., 2016).

User Experience (UX) and User Interface (UI): Unlike desktop applications, mobile apps impose different interaction methods such as gesture, voice, orientation, swiping, use of stylus, and change of screen orientation to mention a few. The device can be used in a very wide range of situations e.g. bright sunlight, dark, remote locations, train etc. These new methods and usage contexts severely affect the overall UI design of the app (Flora et al., 2014b; Santos et al., 2016). Additionally, they require additional programming and testing effort (Kaleel and Harishankar, 2013). On the other hand, screen size fragmentation adds new dimension of complexity (Dehlinger and Dixon, 2011). Small screens means that little data can be displayed and at the same time, users expect to find the information effortlessly. Consequently, developers have to spend additional time and effort to design apps to display the most relevant information for the user (Flora et al., 2014b). Moreover, in some cases, the design of the UI is considered to be one of the most important activities in the development of mobile apps (Flora et al., 2014b; Kumar et al., 2016).

Novice developers: A large portion of the mobile app developer community are considered to be novices. These novice developers lack the required knowledge and experience to develop an apps for variety of

platforms. This is partly due to the fact that they come from traditional web and desktop development backgrounds (Guo et al., 2014).

Insufficient and incomplete requirements: Sometimes mobile app projects are quickly initiated even though some business requirements are still not clearly defined. In consequence, this will increase the workload of development, integration; and app testing (Flora et al., 2014b; Kumar et al., 2016).

Budget and Schedule: Mobile app development is considered to be rapid in order to keep pace with fierce marketing schedules (Kaleel and Harishankar, 2013). Some claim that the development of mobile projects are small and therefore need a small budget. This is a common mistake because mobile app development is a complex process involving different stages (Flora et al., 2014b; Kumar et al., 2016).

Quality Assurance Issues: Compared to desktop and web applications, mobile apps are inherently different. For instance, mobile apps have to deal with inputs from the user as well as inputs from the constantly changing environment e.g. location. Additionally, mobile devices and smart phones are still limited with resources compared with powerful laptop and desktop computers. Further, mobile apps are normally deployed on mobile devices with different OS versions and different screen sizes (Zein et al., 2016).

Security and Privacy: Security is a major challenge in the mobile app industry due to the existence of different devices, operating systems, open platform and malicious apps that can be installed without detection (Kaleel and Harishankar, 2013).

User review feedback cycle: as apps are typically published and downloaded from app stores, real world users can rapidly feedback ratings and detailed information about the apps, often its faults and limitations. Similarly, competing apps are typically hosted in the same store. This adds extra pressures to development teams to both fix and extend their app features to address user feedback as well as competitors, themselves rapidly extending and releasing newer versions of their apps.

Classes of app: The software engineering literature classifies mobile apps into different categories based on a range of different criteria. Mobile applications can be categorized into consumer (End-user) and enterprise apps based on the type of services that app provides (Bhosale and Bhosale, 2014).

- **End-user apps:** Theses 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. (Bhosale and Bhosale, 2014).
- **Enterprise apps:** These apps are designed to meet business needs and aim to increase the efficiency, productivity and satisfaction of company employees (Bhosale and Bhosale, 2014).

There are differences between the characteristics of these two classes of apps: Firstly, the average lifetime of end-user apps is short compared with enterprise applications that have long time goals. Therefore, the enterprise apps require continued maintenance according to new standards and techniques (Nitze, 2014). Secondly, enterprise apps had wider purposes and scope compared with the end-user apps. Therefore, it requires continuous update according to objectives changing, which means it would incur more development cost (Nitze, 2014). Thirdly, end-user apps are often stand-alone, while enterprise apps often use APIs to use the business functions, which means that these apps are constrained by the technical and legal constraints of the enterprise (Nitze, 2014). Finally, enterprise apps require additional requirements to take appropriate action if mobile devices are lost or stolen (Unhelkar and Murugesan, 2010).

3. Related Work

The environment of mobile app development is highly competitive, dynamic, and uncertain (Flora and Chande, 2013). This is clear from the unique challenges facing the mobile app industry. Therefore, as in other domains the adoption of "agile methods" has been proposed as a natural fit for software development of mobile applications to provide a solution for these challenges (Flora et al., 2014a). Key attractions include use of generalists vs specialists, customer on site, test-first development, evolutionary and emergent requirements, and continuous delivery.

However, studies carried out on application of such agile practices to the development of mobile applications has indicated there is a need for tailoring generic, agile software development processes to suit the requirements of mobile apps in particular (Flora et al., 2014a). Therefore, some authors fulfilled some of the shortcomings of the agile method by incorporating practices from other framework (Corral et al., 2013b).

In 2003, the suitability of the agile method for mobile application development was discussed for the first time (Abrahamsson et al., 2003). Then the study by (Abrahamsson, 2005) mapped between agile themes and development characters were observed in mobile application development. The mapping demonstrated why agile is considered the most suitable approach for mobile application development processes. This is due to short development cycles, highly volatile environment, application level software, small teams, object-oriented environment and identifiable customer.

Zein et al, conducted an exploratory multiple case in 2015 involving four software development companies (Zein et al., 2015). 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 lifecycle conformance, context-awareness, and integration testing. In addition, the researchers argued that the industrial mobile app team did not apply formal testing approaches.

To evaluate adopting agile methods in mobile application development and how these approach improve the development process, (Flora et al., 2014a) conducted an online survey involving 130 participants from mobile community including development team members, agile experts, researchers, and other stakeholders. The result of their study indicated that mobile app has dynamic and incomplete requirement, which lead to building mobile application with limited set of features in the first release and update it later by frequent interaction with clients. These make agile methods suitable for mobile application development for the reasons: short time, require flexibility, and reduces time to market.

In the same way, the study by (Santos et al., 2016) conducted a survey to understand how agile practices support mobile application development. With 20 students responding to this survey, most of them did not have experience with a range of development methodologies. The result of their study showed that agile methods are suitable for mobile application development especially in project management and control and development time. The result also showed there are some particular challenges with developing mobile apps. Another survey was conducted from the mobile research and development community in (Flora et al., 2014b) with 130 responses from development team members, consultant and top-level manager. The participants indicated that agile practices should be adapted in mobile application development to mitigate such challenges. However, the participants indicated that they thought that generic agile methods must also be tailored to each app development team.

The effectiveness of agile practices has also been evaluated by the empirical study of (Scharff and Verma, 2010) . In their study, they applied the Scrum method in a classroom setting to assess the effectiveness of application of Scrum in mobile application development. The authors claimed that Scrum helps mobile app development teams to accomplish their tasks in time and that the Scrum method is the key factor for

the success of the project when the time is a constraint. In the same context, (Kaleel and Harishankar, 2013) provided a detailed analysis on scrum practices that suit mobile application development. The author proposed including new practices, which is important for mobile application 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 application development.

Another set of studies proposed mobile software development frameworks based on adapting agile principles. Abrahamsson et al. in (Abrahamsson et al., 2004) proposed the Mobile-D as a first attempt to adopt agile practices in mobile app development to meet the specific demands of this volatile environment. In the same way, Jeong et al. proposed MASAM in 2008 as a process for mobile app development (Jeong et al., 2008). Moreover, Rahimian et al. proposed HME (Rahimian and Ramsin, 2008) as a new agile method that a hybrid Agile and risk-based methodology based on methodology engineering techniques. Another agile method was proposed by Cunha et al (da Cunha et al., 2011), they proposed approach called SLeSS, integrates between scrum and Lean Six Sigma.

On the other hand, the utilization of these approaches in a real world setting was criticized by Corral et al. (Corral et al., 2013b). These proposed agile methods for mobile app development were claimed to be still largely theoretical and that they have not been rigorously evaluated in real industrial contexts (Flora and Chande, 2013).

To summarize, with the growing adoption of mobile apps and the presence of an unlimited number of mobile apps at online stores, additional challenges are facing the development of mobile apps. Such challenges are not present in traditional web and desktop applications. Therefore, the use of agile software development methods in this context has attracted attention from researchers and practitioners. Although some agile frameworks were proposed for mobile app development, these frameworks largely still remain theoretical and there is no evidence of their particular benefits in practice. In addition, there is still a lack of research to understand the key issues and challenges faced by industrial teams when adopting agile practices specifically for mobile app 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 applications (Petersen and Wohlin, 2009). Therefore, there is a need to explore and investigate when adopting agile methods in mobile app development. Thus, conducting a qualitative study to explore adopting agile method in mobile app development can reveal implicit challenges and needs of real industrial teams. Subsequently, this study aims to conduct a multiple-case study to explore issues and challenges of adopting agile practices in mobile apps development to draw more general conclusions.

4. Research Methodology

In this section we present the qualitative research approach that we have applied in this study.

4.1. Case Studies

A case study research method is an empirical study that investigates a contemporary phenomenon within its real-life context, especially when we cannot identify clearly the boundary between the phenomenon and context (Yin, 2013). The qualitative nature of this methodology is concerned with the natural setting of the phenomenon under investigation to obtain detailed qualitative information (Creswell and Poth, 2017). This enables the researcher to obtain better answers from participants and get in-depth understanding of

the context (Runeson and Höst, 2009). This type of research follows a systematic process for data collection, data analysis and generation of results (Verner et al., 2009).

Case studies are used for exploratory purposes, in addition they can be used for explanatory and descriptive purposes (Yin, 2013). 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 (Runeson and Höst, 2009; Verner et al., 2009).In addition, It is strongly recommended that case studies be based on several sources of data and evidence (Yin, 2013).

Use of such case study research is a perfect fit to our study objectives. This will enable us to investigate the adoption of agile and incremental development method in mobile app development industrial context with a qualatative approach wth a small number of participants for in-depth understanding of their contexts and experiences.

4.2. Research Questions

We have formulated the following key research questions for our study:

RQ1) How do industrial teams apply agile methods when developing their mobile apps?

RQ2) What are the challenges and merits of these agile methods used by these industrial teams compared with the state-of-the-art?

RQ3) How can generic agile software development methods be adjusted to better serve mobile app development contexts?

Data collection procedures were shaped based on the above research questions by focusing on certain aspects and constructing specific interview questions.

4.3. 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 (Yin, 2013). We investigated and drew conclusions from several industrial cases. Each single case represents a mobile app software development company in Palestine in the field of mobile apps. See Fig. 1 that show multiple case study design. We chose companies and teams with a view to understanding a range of organisational, team, project and product contexts for agile mobile app development.

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. We used a theoretical frame of reference for software engineering studies based on prior relevant works (Runeson and Höst, 2009).

4.4. Case Selection and Context

In our 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 and ensuring a variety of companies, products and teams.

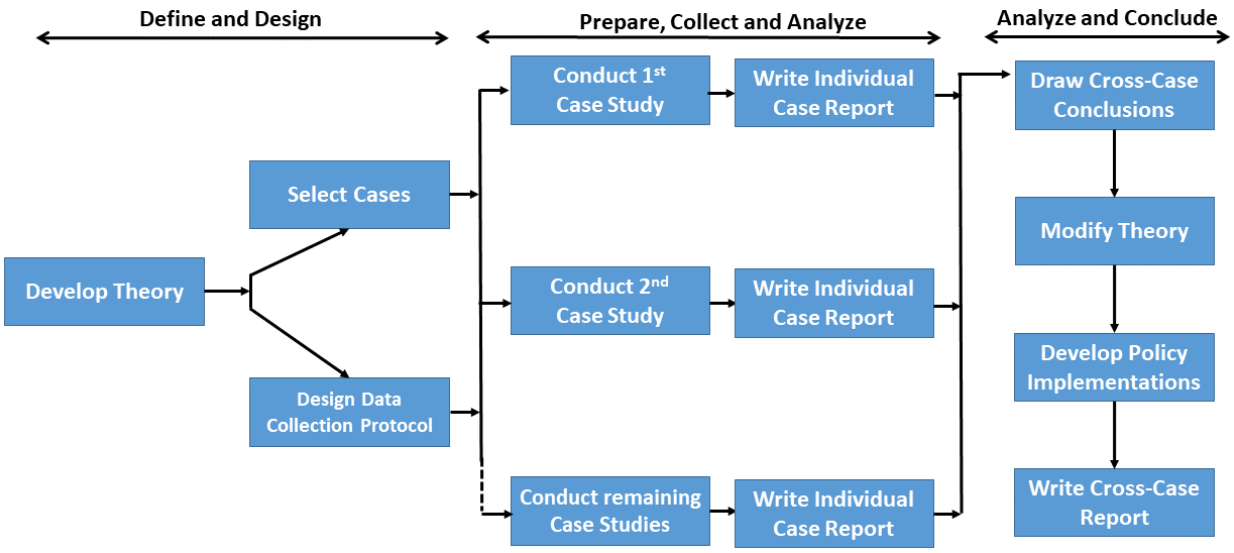

Figure 1: Multiple Case Study Design (Yin, 2013)

4.5. Data Collection Methods

Our data collection methods are based on observations, interviews, and focus groups. Using these three data collection methods we applied data triangulation in order to validate and crosscheck the findings of our study and increase its data reliability (Yin, 2013).

4.5.1. Observations

Unlike interviews, observations often provide more objective and descriptive information related to the research topic (Hancock and Algozzine, 2016). Data collection through observations was carried out through taking field notes on the behaviors and activities of developers in mobile app development teams without participation (Yin, 2013). This can help us to reveal more information about the challenges and issues of adopting agile methods, activities, techniques applied in software development process.

During the observations, the first author recorded the activities in unstructured or semi-structured way at the research site. All activities involved in the software development process for mobile apps are observed and investigated. Therefore the observed activities include all development activities (requirements elicitation, requirements analysis, design, development and testing, and which agile method is followed), the usage of the IDE (integrated development environments), bug records, bug management processes and challenges and limitations of testing methods and techniques applied.

4.5.2. Interviews

In this method, face-to-face interviews were conducted with the mobile app software development team members. Interviews were semi-structured and open-ended since they are well suited for this kind of research (Yin, 2013). The first author asked predetermined and flexibly-worded questions to collect tentative answers. In addition, the researcher 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.

4.5.3. Focus groups

These interviews were semi-structured and generally open-ended questions to gather views and opinions from group of participants. By interviewing individuals, we gained efficiency but with some loss in depth. These group interviews however allowed participants to express themselves when they are part of a group than when they are the target of an interview (Hancock and Algozzine, 2016).

4.6. Thematic Data Analysis

For data analysis, we applied a thematic analysis method, which is reported in many qualitative Studies (Yin, 2013; Runeson and Höst, 2009; Verner et al., 2009). Thematic analysis is an iterative method for identifying common themes in qualitative data to understand specific phenomena in particular contexts (Clarke and Braun, 2013). This method classifies concepts that may affect studied phenomenon such as behaviors, events or activities (Merton, 1975). 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 (Robson and McCartan, 2016). Other researchers recommend to maintain a chain of evidence from the findings to the original data (Runeson and Höst, 2009). To achieve this, we carefully assigned special identification numbers for each session of the interviews, focus groups, and observations. Later, each sentence of these sessions was given a special number that is derived from the originating session number. Using this approach, we could link studied sentences to their original session.

The thematic data analysis process is based on guidelines provided by (Clarke and Braun, 2013; Cruzes and Dyba, 2011). First, researchers should be intimately familiar with their data by reading and re-reading it. Secondly, 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, we conducted a detailed analysis of each theme, 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.

4.7. Data Collection Procedures

The availability of team members and willingness of company management were the main factors in the form of data collection strategy used in this study. Data collection lasted for a period of three months. In the first case study, the first author 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 was 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, we carried out several one-to-one interviews. In the fourth case, we carried out focus group interviews with members of mobile development in one meeting session and one-to-one interview.

A diversity of data collection methods has helped us to apply data triangulation in this research study. Moreover, to maintain the chain of evidence (Yin, 2013; Runeson and Höst, 2009), 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, and all data records were numbered using special number.

Moreover, during thematic analysis coding, each sentence was given a special code and linked to its original document.

5. Results

In this section we present the key results of the data collected using observations, interviews and focus groups. We first present a background of the selected case studies included in this study in terms of the nature of their work; the tools, languages and techniques applied in their teams. The four cases are given codes as follows C1, C2, C3 and C4, to refer to each case to keep confidentiality.

5.1. Demographic Information

The first case study company, C1, specializes in the mobile app development field and has two development teams. Each team has two developers and a QA engineer applying agile development processes. During this study they were working on a taxi software app that deals with web and smart phone clients. Their app was built to support both the Android and IOS platforms.

The second case company, C2, is a well-established and large software development company developing various types of app. This company provides business and software solutions and a wide range of IT services. They mainly focus on developing mobile apps for the enterprise level and apply a Kanban (with feature per release) agile process for their 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 company, 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 based on the Scrum agile method. This company has five mobile app development teams. Each team consists of approximately five members. Most of their apps are social and business mobile apps and they support native app (Android and iOS app) and hybrid app development approaches.

Finally, the forth case study company, C4, is a startup mobile app software development company. They apply a Scrum agile method. Moreover, they are applying best practices in building mobile apps by specialized and skilled mobile specialists. They are also 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 a hybrid mobile app framework that they use as a proof of concept prototyping platform. Each platform has its own special development team of five members. The demographic data for each case is shown in Table 1.

A set of tools and languages currently used by each team in developing mobile apps, as well as the platforms used to produce mobile apps and tools used by each case to recording the software bugs, are summarised in Table 2. During the three months of data collection through observations, interviews and focus groups, a total of 14 developers were involved in our study.

5.2. Development Process and Environment

In this section we discuss the development processes used by our case study companies and their activities as well as the environment used in mobile app development. The quotations in this section and the following section represent anonymousised actual responses we obtained from participants.

Table 1: Demographics of four cases

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 5 members	4 teams, each has 4 members	5 teams, each has 5 members	2 teams, each has 4 members

Table 2: Tools, languages and techniques applied

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

Generally, all four cases applied some form of agile method as a development process for their mobile app development. All of them applied the 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 releases about every two weeks. However, C2 provided a release per feature. One of the project managers in C4 said: *We hope do a release per feature which means higher frequency for the releases, but we cant do that before we reach 100% automation of testing* project manager, C4.

We observed that the customers and a technology road map 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 (Kappel, 2001; Kostoff and Schaller, 2001). Moreover, this road map should answer a set of questions that are related to markets, products, features and technologies (Suomalainen et al., 2015). 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 major 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 are 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. Then they gave high priority for high risk features. In addition, they got feedback from the customer after each iteration and acceptance tests.

We noticed that all teams give great care to the UX (User Experience) more than any other aspects in a 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 impose 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 the mobile game apps domain, according to a team member from C1, who 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 teams as the main development platforms. We also observed that the teams try to apply feature parity between these platforms, which means they develop the same feature in iOS and Android platforms. They 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 the 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.

Furthermore, it was clearly observed that there are many constraints on mobile app development. For instance, the diversity of users and the changes they need. In addition, 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 a mobile app always needs to be changed and updated due to a diversity in users and user feedback there are always changes in need. Moreover, there are always new devices need to be compatible with your app.* developer, C1.

Although the development process of mobile apps is known to be rapid, it was noticed that existing tools do not support such constraints. For example, compilation and automation testing tools for mobile apps need more time compared to other web and desktop tools, *The development isnt smooth as other platform, because the tools dont 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, we observed that there is a lack of statistics about the devices and platforms used by target end 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" - project manager, C1.

Regarding testing, we observed that, in almost all cases, teams made heavy use of 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 isnt 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 our previous study in (Zein et al., 2015).

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 customers especially at the beginning of the project. They also send questionnaire to the customers 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) (Speicher et al., 2014; Cámara and Kobsa, 2009; Crook et al., 2009).

More specifically, C4 implemented the same feature in two different versions and distributed them between users, then using analytic system which 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's 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 a lot of stress due to inherently very 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 on-line store so you should change some features to make the app acceptable by the store"* - project manager, C3.

5.3. Agile Principles

In this section we discuss agile principles that were mentioned in (Petersen and Wohlin, 2009). During the data collection we investigated how teams might by applying, or not, each of these principles. All participants in interviews and focus groups were asked about these principles.

We focus on the principles that are applied differently or not applied in our case study examples. 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 3.

Table 3: Key Agile Principles

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 (Petersen and Wohlin, 2009; Sharma et al., 2012; Abbas et al., 2008; Petersen and Wohlin, 2008, 2010).
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 (Petersen and Wohlin, 2009, 2008, 2010).
Time Boxing	Time Boxing means that each iteration of project has time limit (i.e. fix duration and deadline) (Petersen and Wohlin, 2009; Sharma et al., 2012; Petersen and Wohlin, 2008, 2010; Leffingwell and Muirhead, 2004).
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 (Petersen and Wohlin, 2009; Sharma et al., 2012; Petersen and Wohlin, 2010).
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 (Flora and Chande, 2014; Xihui, 2005)
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 (Cao et al., 2009; Beck, 2000).
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 (Petersen and Wohlin, 2009, 2008). This practice helps to resolve potential misunderstandings and determine the basis for a smoother implementation of the day-to-day activities of the project (Mishra et al., 2009)
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. (Guzzo and Dickson, 1996; Highsmith, 2009). Self-organizing team leads to motivating members to commit themselves to their responsibility, greater creativity and higher productivity and quality (Fenton-O'Creevy, 1998).
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 (Szalvay, 2004; Awad, 2005)
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 (Boehm and Turner, 2004).
Adaptive Planning	The planning in agile is less formal (Norin and Karlström, 2006), and it is adapted for a release with short period and focus on factors that effect on increments delivery (Nagy et al., 2010). Moreover, Agile always welcomes change in technology, requirements or method itself (Abbas et al., 2008).

Requirements Prioritization	Agile methods specify the most important requirements to implement earlier than others based on their business value (Perini et al., 2009; Cao and Ramesh, 2008). Since whole system functionality cannot be implemented in the same iteration (Racheva et al., 2008). Therefore, the requirement with less priority is implemented in the upcoming iteration (De Lucia and Qusef, 2010).
Fast Decision Making	Agile development empowers members who have the power to make decisions that are not limited to a particular role (Su and Zhu, 2015). 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 (Ivan and Despa, 2014).
Frequent Integration	Team members' code should be integrated and tested frequently (Olsson et al., 2012; Petersen and Wohlin, 2008, 2010, 2009). This practice with small releases enable the team always to deliver working software because it guarantees a permanent availability of an executable system (Olsson et al., 2012).
Simplicity of Design	Agile software processes maintains the simplicity of system design as much as possible at any moment in time (Fitzgerald et al., 2006), maintaining simplicity of design helps the team to work productively with minimal documentation outside the source code (Maurer and Martel, 2002).
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(Fowler and Beck, 1999; Moser et al., 2008; Lindvall et al., 2002)
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 (Sedano et al., 2016)

We observed that all cases followed the use of short iterations and incremental development principles. Further, it was confirmed by the participants that the iterations and release times in mobile app development is shorter than other web and desktop domains (one week for each iteration and two weeks for each release), as a 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 its difficult to avoid bugs due to rapid development.* Developer, C1.

We also observed that time-boxing was applied in all cases except for C2 (because it follows release per branch approach). However, the participants from C1 and C4 aimed 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, we 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.

Furthermore, it was observed that in enterprise apps the teams 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 an 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, we 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.

6. Discussion

In this section we discuss the results of our cases, use these to answer our key research questions, and compare them with state-of-the-art research. Our related work review confirmed that agile approach was seen as a natural fit for mobile app development industry (Wasserman, 2010; Abrahamsson et al., 2003; Corral et al., 2013a; Ashishdeep et al., 2016; Holler, 2006). Furthermore, using traditional software engineering approaches cannot always be applied directly to the mobile industry because of their tight timeframes, highly dynamic nature, and large scale competition in most sectors (Dehlinger and Dixon, 2011). We recommend that the development process chosen should be tailored to best suit mobile apps peculiarities (Flora and Chande, 2013). There are several agile software engineering methods proposed for mobile app development (Abrahamsson et al., 2004; Jeong et al., 2008; Rahimian and Ramsin, 2008; da Cunha et al., 2011).

Many insights can be obtained from our results. Firstly, it is clear that an agile software development approach has been applied in all study cases. The studied teams apply iterative and incremental development, which they claim helps 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.

Secondly, the iterations and incremental development in the agile methods are suitable for the 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:** It was obvious 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 is likely to make users more satisfied since they find that there are 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:** this principal does not seem to fit well 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:** some of the development teams cannot commit to this principle in the case of strong competition or high internal customer demand. it was noticed that sometimes they needed to stop working on a feature because the competitors launched a new feature or enterprise clients demanded a critical fix of feature urgently. Thus, in some situations teams need a process allowing them to suspend current focus work and develop a new feature out of sequence.
- **The on-site customer principal:** Since there are large numbers of users for mobile apps, the on-site customer principal cannot always be adopted in all mobile app development scenarios. However, development teams can get feedback from users using questionnaires, focus groups with sample of users, or using indirect ways such as the use of A/B testing and app store reviews.

The development teams we studied were adopting new techniques to address some of these challenges of mobile app development. For instance, it was noticed the teams had to offer rapid releases (one week for each iteration and two weeks for each release). This insight has also came up in other studies (Flora and Chande, 2013; Zein et al., 2015; Dehlinger and Dixon, 2011; Kaur and Kaur, 2015; Kumar and Goel, 2012). In addition, the challenge of the diversity of mobile devices and platforms was addressed by developing the same app for more than on platform (Android and iOS). Regarding the challenge of getting continuous feedback from users, teams applied questionaires, direct contact with users, and using indirect contact through A/B testing and analytics systems. On the other hand, in order to keep pace with the competitors and satisfy low tolerance app users, the studied teams 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., 2014a; Dehlinger and Dixon, 2011).

However, there are still some challenges in agile-based mobile app development that remain unsolved. Firstly, many of the development tools and IDEs do not currently support the rapid nature of mobile app development and associated agile practices. For instance, the compilation tools and automated testing tools take considerably longer time to finish. Secondly, there is a lack of adoption of automated testing tools because these tools need more investment in time and resources. This also came out in the study by (Zein et al., 2015). Thirdly, the diversity of mobile devices and platforms puts a lot more pressure on developers and QA engineers to develop and test the same mobile app for different platforms on various devices. Finally, there is a 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.

Adopting agile development methods when developing enterprise mobile apps is somehow different as compared to single end-user apps. In this type of app, 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 did not apply on-site customer principle by involving product owner in the planning sessions as in traditional app development.

Finally, in addition to the known challenges in mobile app development in the literature, we found that most mobile app developers are suffering from considerable work stress for several reasons. First, the rapid development of mobile apps and the short release time. Second, the long duration and the constraints imposed by some online stores to approve the uploaded apps. Third, developers are highly concerned about discovering app bugs before users do, since they feel that users of their mobile apps are less tolerant of issues than the users of other kinds of applications.

7. Recommendations

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

Firstly, there is a need for better statistics about devices and platforms used by the target end users to quickly reach development teams. Such statistics can help development teams when developing a new app to know which devices and platforms commonly used by the target users. 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 be used 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 showed that the capabilities of mobile app development tools are still limited when compared to traditional web and desktop applications' tools, and they need to be improved. More specifically, the compilation and debugging tools for mobile apps are considerably slow and relatively taking more need time to execute. Thus, optimizing 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 time and effort needs to be invested in set up of mobile app automation testing at the beginning of the project. However, using this automation testing is seen to be very helpful for mobile development team. This is especially in advanced release scenarios 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 testing 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.

Fourthly, although agile methods are claimed to be a natural fit for the mobile app development industry, it seems that some traditional agile principles cannot be applied directly to the mobile industry. Based on our results, we recommend to tailor the following agile principles to best suit mobile app development peculiarities:

- **Feature per release:** Feature per release should be applied instead of time boxing, because feature per release is more suit for the nature of mobile app development fast and fierce competition for the reasons we mentioned in the discussion section.
- **Eliminating No change of started project:** This is due to strong competition and high internal demand for customers in mobile app development. Therefore, in some cases it is better to stop working on a feature that was already included in the sprint, and start working on another unplanned feature to satisfy customers. During later sprints, the team can resume working on the feature that was stopped.
- **Use alternatives to on-site customer:** Since there are large numbers of users of mobile apps and there seems to be a low tolerance of app users to errors or usability problems, the on-site customer principal cannot always be adopted in all mobile app development scenarios. Therefore, teams need to use indirect ways to get feedback from users such as the use of A/B testing and app store reviews.

Finally, the restrictions imposed by online stores such as Google Play and App Store add additional overhead on developers. We understand that such restrictions are important to maintain the quality of uploaded apps. However, these restrictions consume valuable time and effort from development teams.

8. Conclusion

Mobile app development is different and more complex than traditional web and desktop contexts. We found that agile methods have to be tailored in order to be adopted for mobile app development. This study presents more 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. We argue 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 work stress, inappropriate automation tools, and very tight time-to-market. In future work, we intend to develop a new agile method for mobile app development based on the results of this study and then validate the effectiveness of this method in the real world.

References

Abbas, N., Gravell, A. M., and Wills, G. B. (2008). Historical roots of agile methods: Where did agile thinking come from? In *International Conference on Agile Processes and Extreme Programming in Software Engineering*, pages 94–103. Springer.

Abrahamsson, P. (2005). Keynote: Mobile software development–the business opportunity of today. In *proceedings of the International Conference on Software Development*, pages 20–23.

Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme, T., Jäälinoja, J., Korkala, M., Koskela, J., Kyllönen, P., and Salo, O. (2004). 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*, pages 174–175. ACM.

Abrahamsson, P., Warsta, J., Siponen, M. T., and Ronkainen, J. (2003). New directions on agile methods: a comparative analysis. In *Software Engineering, 2003. Proceedings. 25th International Conference on*, pages 244–254. Ieee.

Ashishdeep, A., Bhatia, J., and Varma, K. (2016). Software process models for mobile application development: A review. *Computer Science and Electronic Journal*, 7(1):150–153.

Awad, M. (2005). A comparison between agile and traditional software development methodologies. *University of Western Australia*.

Beck, K. (2000). *Extreme programming explained: embrace change*. addison-wesley professional.

Bhosale, S. and Bhosale, M. (2014). Architectural considerations while developing enterprise mobile applications. *International Journal of Scientific and Research Publications*.

Boehm, B. and Turner, R. (2004). Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods. In *Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on*, pages 718–719. IEEE.

Cámara, J. and Kobsa, A. (2009). 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*, pages 116–135. Springer.

Cao, L., Mohan, K., Xu, P., and Ramesh, B. (2009). A framework for adapting agile development methodologies. *European Journal of Information Systems*, 18(4):332–343.

Cao, L. and Ramesh, B. (2008). Agile requirements engineering practices: An empirical study. *IEEE software*, 25(1).

Clarke, V. and Braun, V. (2013). Teaching thematic analysis: Overcoming challenges and developing strategies for effective learning. *The psychologist*, 26(2):120–123.

Corral, L., Sillitti, A., and Succi, G. (2013a). Agile software development processes for mobile systems: Accomplishment, evidence and evolution. In *International Conference on Mobile Web and Information Systems*, pages 90–106. Springer.

Corral, L., Sillitti, A., and Succi, G. (2013b). 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*, pages 19–24. IEEE.

Creswell, J. W. and Poth, C. N. (2017). *Qualitative inquiry and research design: Choosing among five approaches*. Sage publications.

Crook, T., Frasca, B., Kohavi, R., and Longbotham, R. (2009). 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*, pages 1105–1114. ACM.

Cruzes, D. S. and Dyba, T. (2011). Recommended steps for thematic synthesis in software engineering. In *Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on*, pages 275–284. IEEE.

da Cunha, T. F. V., Dantas, V. L., and Andrade, R. M. (2011). 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*, pages 283–292. IEEE.

De Lucia, A. and Qusef, A. (2010). Requirements engineering in agile software development. *Journal of Emerging Technologies in Web Intelligence*, 2(3):212–220.

Dehlinger, J. and Dixon, J. (2011). Mobile application software engineering: Challenges and research directions. In *Workshop on Mobile Software Engineering*, volume 2, pages 29–32.

Fenton-O'Creevy, M. (1998). Employee involvement and the middle manager: evidence from a survey of organizations. *Journal of Organizational Behavior*, pages 67–84.

Fitzgerald, B., Hartnett, G., and Conboy, K. (2006). Customising agile methods to software practices at intel shannon. *European Journal of Information Systems*, 15(2):200–213.

Flora, H. K. and Chande, S. V. (2013). A review and anaysis on mobile application development processes using agile methodlogies. *International Journal of Research in Computer Science*, 3(4):9.

Flora, H. K. and Chande, S. V. (2014). A systematic study on agile software development methodologies and practices. *International Journal of Computer Science and Information Technologies*, 5(3):3626–3637.

Flora, H. K., Chande, S. V., and Wang, X. (2014a). Adopting an agile approach for the development of mobile applications. *International Journal of Computer Applications*, 94(17).

Flora, H. K., Wang, X., and Chande, S. V. (2014b). An investigation into mobile application development processes: Challenges and best practices. *International Journal of Modern Education and Computer Science*, 6(6):1.

Fowler, M. and Beck, K. (1999). *Refactoring: improving the design of existing code*. Addison-Wesley Professional.

Guo, C., Xu, J., Yang, H., Zeng, Y., and Xing, S. (2014). An automated testing approach for inter-application security in android. In *Proceedings of the 9th International Workshop on Automation of Software Test*, pages 8–14. ACM.

Guzzo, R. A. and Dickson, M. W. (1996). Teams in organizations: Recent research on performance and effectiveness. *Annual review of psychology*, 47(1):307–338.

Hancock, D. R. and Algozzine, B. (2016). *Doing case study research: A practical guide for beginning researchers*. Teachers College Press.

Highsmith, J. (2009). *Agile project management: creating innovative products*. Pearson Education.

Holler, R. (2006). Mobile application development: a natural fit with agile methodologies. *VerisonOne LLC, Alpharetta*.

Ivan, I. and Despa, M. L. (2014). Statistical indicators for measuring innovation in it project management. *Procedia Economics and Finance*, 10:167–177.

Jeong, Y.-J., Lee, J.-H., and Shin, G.-S. (2008). Development process of mobile application sw based on agile methodology. In *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, volume 1, pages 362–366. IEEE.

Kaleel, S. B. and Harishankar, S. (2013). Applying agile methodology in mobile software engineering: Android application development and its challenges. *Computer Science Technical Reports*.

Kappel, T. A. (2001). Perspectives on roadmaps: how organizations talk about the future. *Journal of Product Innovation Management*, 18(1):39–50.

Kaur, A. and Kaur, K. (2015). Suitability of existing software development life cycle (sdlc) in context of mobile application development life cycle (madlc). *International Journal of Computer Applications*, 116(19).

Kostoff, R. N. and Schaller, R. R. (2001). Science and technology roadmaps. *IEEE Transactions on engineering management*, 48(2):132–143.

Kumar, A. and Goel, B. (2012). Factors influencing agile practices: A survey. *International Journal of Engineering Research and Applications*, 2(4):1347–1352.

Kumar, N. A., Krishna, K. H., and Manjula, R. (2016). Challenges and best practices in mobile application development. *Imperial Journal of Interdisciplinary Research*, 2(12).

Leffingwell, D. and Muirhead, D. (2004). Tactical management of agile development: Achieving competitive advantage. *Rally Software Development Corporation*, pages 1–23.

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Williams, L., and Zelkowitz, M. (2002). Empirical findings in agile methods. In *Conference on extreme programming and agile methods*, pages 197–207. Springer.

Liu, Z., Gao, X., and Long, X. (2010). Adaptive random testing of mobile application. In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, volume 2, pages V2–297. IEEE.

Maurer, F. and Martel, S. (2002). Extreme programming. rapid development for web-based applications. *IEEE Internet computing*, 6(1):86–90.

Merton, R. K. (1975). Thematic analysis in science: notes on holton's concept. *Science*, 188(4186):335–338.

Mishra, A., Sinha, K. K., and Thirumalai, S. (2009). Execution capabilities of distributed technology projects: A comparative evaluation of efficiency using stochastic frontier analysis.

Moser, R., Abrahamsson, P., Pedrycz, W., Sillitti, A., and Succi, G. (2008). A case study on the impact of refactoring on quality and productivity in an agile team. In *Balancing Agility and Formalism in Software Engineering*, pages 252–266. Springer.

Nagy, A., Njima, M., and Mkrtchyan, L. (2010). 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*, pages 192–199. IEEE.

Nitze, A. (2014). The case for web technologies in mobile business apps. *Web Technologies for Business Apps*, pages 87–96.

Norin, J. and Karlström, D. (2006). Lean configuration management. *Softhouse Consulting, Stormgatan*, 14.

Olsson, H. H., Alahyari, H., and Bosch, J. (2012). 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*, pages 392–399. IEEE.

Oulasvirta, A., Wahlström, M., and Ericsson, K. A. (2011). What does it mean to be good at using a mobile device? an investigation of three levels of experience and skill. *International journal of human-computer studies*, 69(3):155–169.

Perini, A., Ricca, F., and Susi, A. (2009). Tool-supported requirements prioritization: Comparing the ahp and cbrank methods.

Information and Software Technology, 51(6):1021–1032.

Petersen, K. and Wohlin, C. (2008). Issues and advantages of using agile and incremental practices. In *Software Engineering Research and Practice*.

Petersen, K. and Wohlin, C. (2009). A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of systems and software*, 82(9):1479–1490.

Petersen, K. and Wohlin, C. (2010). The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empirical Software Engineering*, 15(6):654–693.

Racheva, Z., Daneva, M., and Buglione, L. (2008). Supporting the dynamic reprioritization of requirements in agile development of software products. In *Software Product Management, 2008. IWSPM'08. Second International Workshop on*, pages 49–58. IEEE.

Rahimian, V. and Ramsin, R. (2008). 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*, pages 337–342. IEEE.

Robson, C. and McCartan, K. (2016). *Real world research*. John Wiley & Sons.

Runeson, P. and Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2):131.

Santos, A. R., Kroll, J., Sales, A., Fernandes, P., and Wildt, D. (2016). Investigating the adoption of agile practices in mobile application development. In *ICEIS (1)*, pages 490–497.

Scharff, C. and Verma, R. (2010). 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*, pages 25–31. ACM.

Sedano, T., Ralph, P., and Péraire, C. (2016). Practice and perception of team code ownership. In *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, page 36. ACM.

Sharma, S., Sarkar, D., and Gupta, D. (2012). Agile processes and methodologies: A conceptual study. *International journal on computer science and Engineering*, 4(5):892.

Speicher, M., Both, A., and Gaedke, M. (2014). Ensuring web interface quality through usability-based split testing. In *International Conference on Web Engineering*, pages 93–110. Springer.

Su, F. and Zhu, L. (2015). An empirical study of the decision-making process in agile software development based on industries from china.

Suomalainen, T., Kuusela, R., and Tihinen, M. (2015). Continuous planning: an important aspect of agile and lean development. *International Journal of Agile Systems and Management*, 8(2):132–162.

Szalvay, V. (2004). An introduction to agile software development. *Danube technologies*, 3.

Unhelkar, B. and Murugesan, S. (2010). The enterprise mobile applications development framework. *IT professional*, 12(3):33–39.

Verner, J. M., Sampson, J., Tosic, V., Bakar, N. A., and Kitchenham, B. A. (2009). Guidelines for industrially-based multiple case studies in software engineering. In *Research Challenges in Information Science, 2009. RCIS 2009. Third International Conference on*, pages 313–324. IEEE.

Wasserman, A. I. (2010). Software engineering issues for mobile application development. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, pages 397–400. ACM.

Xihui, Z. (2005). Software development methodologies trends and implications. *Adopting the IS 2009 model curriculum: A panel session to Address the Challenges for Program Implementation: 173, 2009. Engineering & Remote Sensing*, 71(4):415–424.

Yin, R. K. (2013). *Case study research: Design and methods*. Sage publications.

Zein, S., Salleh, N., and Grundy, J. (2015). Mobile application testing in industrial contexts: an exploratory multiple case-study. In *International Conference on Intelligent Software Methodologies, Tools, and Techniques*, pages 30–41. Springer.

Zein, S., Salleh, N., and Grundy, J. (2016). A systematic mapping study of mobile application testing techniques. *Journal of Systems and Software*, 117:334–356.