14th International Conference on Intelligent Software Methodologies, Tools and Techniques (SOMET2015), Naples, Italy, Sept 15-17, 2015, © Springer.

Mobile Application Testing in Industrial Contexts: An Exploratory Multiple Case-Study

Samer Zein[1], Norsaremah Salleh[1], John Grundy[2]

[1] Department of Computer Science, International Islamic University, Kuala Lumpur, Malaysia
samer.m.zain@gmail.com, norsaremah@iium.edu.my
[2]Centre for Computing and Engineering Software Systems, Swinburne University of Technology, Melbourne, Australia
jgrundy@swin.edu.au

Abstract— Recent empirical studies in the area of mobile application testing indicate the need for specific testing techniques and methods for mobile applications. This is due to mobile applications being significantly different than traditional web and desktop applications, particularly in terms of the physical constraints of mobile devices and the very different features of their operating systems. Nonetheless, very few of the existing studies are based on a thorough investigation and understanding of real-world software development teams in an industrial contexts. Therefore, we were motivated to conduct the study reported herein: a multiple case-study approach involving four software development companies in the area of mobile and smartphones application. We aimed to identify testing techniques currently being applied by developers and challenges that they are facing. Our principle results are that many industrial teams seem to lack sufficient knowledge on how to test mobile applications, particularly in the areas of mobile application life-cycle conformance, context-awareness, and integration testing. We also found that there is no formal testing approach or methodology that can facilitate the development team to systematically test a critical mobile application.

Keywords— Software testing, mobile applications, case study.

1 Introduction

Mobile applications are becoming very popular in that they are currently being integrated into almost all IT domains and their usage is widespread in everyday life [1]. Some estimates from the International Data Corporation (IDC) show that for the first time, the worldwide shipment of smartphones had reached one billion units in 2013 alone [2]. Mobile applications nowadays are not only developed to serve the entertainment or social media sector, but also target safety and time critical domains, such as payment systems, m-government, military, and mobile health initiatives, to mention just a few[1], [3]. Mobile users are increasingly relying on mobile applications for their computing needs in a way that is outpacing the use of personal computers and desktops. Users fully expect that such applications are easy, reliable and safe to use. The increasing usage in enterprise as well as leisure activities means users now expect mobile device-based solutions for most if not all of their computing tasks. Applications should thus be well-integrated, well-designed, accessible, robust and reliable. However, this makes mobile application solutions not only more complex and challenging to develop, but are also difficult to test and to validate [4].

It has been reported that the peculiarities of mobile application testing are due in part to the diversity of mobile platforms and features of mobile devices[1], [5]. For example, devices range in screen size (small pager or smartphone to large tablet); interaction mechanism (stylus, finger, keyboard, gesture); network bandwidth (e.g. Bluetooth, 3G, WiFi); storage capacity, CPU speed, device size, and device integration with external peripherals and other computing systems. This makes it very hard to ensure that applications can be used effectively and efficiently under any context and environment. An important finding from the literature shows that there is a lack of studies on how to elicit specific testing requirements related to mobile application life cycle properties from requirements specifications [6]. In particular, developers have to build (most of) the application in order to ensure that it will look and behave correctly. These issues motivate a need for better mobile testing approaches, techniques and tools[1], [6].

In a recent study we conducted a systematic mapping study targeting empirical studies of mobile application testing techniques and challenges [6]. Our mapping study identified and analysed 45 empirical studies and revealed significant gaps in the current body of research, suggesting several areas for further investigation. Among the issues we identified was most notably the need for research in real-world industrial contexts of mobile application testing. We also identified the need for specific testing techniques for mobile application development, life cycle conformance and integration testing.

Since little information on mobile application testing techniques and issues is currently available in the context of industrial development companies, we undertook this research using an exploratory multiple-case study methodology. Understanding the developers' and test engineers' perception of the peculiarities, testing techniques and issues of mobile application testing are a challenge [1]. Such an understanding can reveal important insights and produce new perspectives on how mobile applications are developed in industrial contexts. It can also influence the research into new testing techniques and tools and importantly, how these techniques and tools might be most effectively transitioned into industrial practice. Therefore we have conducted a comprehensive multiple-case study investigation on mobile application testing techniques and issues using four different industrial cases. Each industrial case represents a software development company in the field of mobile applications development located in Palestine, to which the lead author had good access to the company and staff. In this study, we analysed the results from these multiple-case studies and compared them with mobile application testing techniques and challenges identified from our mapping study. The novelty of our study is that it represents the first attempt to explore and investigate mobile application testing techniques and challenges in four different companies and to help reveal key developer needs in such context. Based on our motivation above, the following study objectives were formulated:

- to find out how industrial development teams currently tackle testing requirements of mobile applications in real industrial contexts;
- to investigate and identify specific testing challenges and issues related to testing mobile applications that industrial mobile application developers have;
- to compare the results of our case study with the state of the art from our mapping study to help increase the generalizability of others findings to date;
- to identify key future research areas in mobile application development testing that would be of benefit to industrial developers.

The remainder of this paper is organized as follows: Section 2 summarises the current state of the art in mobile application testing and an overview of related work for this study. Section 3 describes the methodology we used to conduct the case study research. Section 4 presents the results from data analysis from the four industrial case studies, followed by a detailed discussion of these results in Section 5. Finally, Section 6 presents threats to validity and section 7 concludes our work and outlines key areas for future research.

2 Related Work

The novelty of our case study is that it reports the first attempt to explore the real-world of mobile application testing contexts. To our best knowledge, there are no field or case studies in the literature exploring testing methods and testing challenges in the relatively young and rapidly evolving area of mobile and smartphone applications in real contexts. However, there are some case studies that are reported to evaluate a certain testing frameworks or models. Such studies, apply field study approach for the sole purpose of evaluating their suggested solution.

For instance, Canfora et al. [7] conducted a case study to investigate and compare their proposed solution platform which addresses the problem of automating execution of user experience (UX) test for applications running on Android OS. Another study by Ravindranath et al. [8] conducted a field experiment to evaluate their testing tool on 30 market applications using 30 users for the duration of 4 months. On the other hand, Losada et al. [9] evaluated the application of an agile usability engineering methodology named InterMod in the development process of real mobile application project.

Further, Amalfitano et al. [4] present approaches to test context-aware applications and conduct case studies on real mobile applications to evaluate their approaches.

In this present case study, and after conducting a thorough literature survey, our main focus was on determining the testing challenges associated with life-cycle conformance testing, integration testing and automation testing in the context of mobile application development. In terms of the techniques reported in the existing literature, we focused on any specific method or procedure that is used in testing mobile applications. The state-of-art of mobile application testing challenges and techniques can be summarized into four important aspects as below:

Life-cycle conformance: For mobile applications, the *life-cycle* is regarded as the different states that an application can go through during its run time and the transitioning between these states [5]. When developing mobile applications running on modern mobile operating systems such as Android, iOS and J2ME, the developer has to be fully aware of the states of the life-cycle in order to ensure correct behaviour of application under all cases [5]. Additionally, both the developer and test engineer should have adequate understanding of mobile application life-cycle models, properties and events to make sure that no data is lost when the user swaps between applications or when the operating system shuts down the application itself. Such understanding will ensure that developers can build a reliable and robust mobile applications that operates correctly and is able to maintain data integrity [10, 11, 12]. Findings from our mapping study showed that there are only two studies that have proposed and evaluated such approaches to test conformance of mobile applications to life cycle models [5], [13]. However, the suggested approaches are very basic and that most of the critical steps proposed are manual and depend on the developer or tester's perception on the problem in hand.

Integration testing: Integration testing for mobile applications refers to the testing of mobile applications taking into account inter-application communication via intents or content providers in Android platform [1]. Usually, and in most cases, mobile applications do not exist on their own and there is a need to exchange data with other components or applications that belong to the operating system. Similarly, mobile applications also often require communication with other external applications such as social networking sites (e.g. Facebook, Twitter and MySpace) [10]. This has motivated us to investigate how such integration testing is carried within industrial contexts, and what are the particular challenges that mobile application integration testing engineers face.

Automation testing: Most mobile application development is considered to be rapid development and teams should deliver applications to market in a short period of time to keep up with the market demand [14]. With the help of test automation, test engineers are more likely to be able to keep pace with developers to maintain agility [15, 16]. Furthermore, test automation can save test engineers from time consuming, routine and error prone manual testing activities. In our mapping study, we identified a number of studies offering tools and methods to automate test process particularly when performing manual functional testing of application through the user interface (UI) (e.g. [14], [17]). Such manual UI tests were repeated whenever there is a new build released by developers. This activity thus consumes a lot of time and effort. Automated UI testing and regression testing [16] are important requirements for agile teams. In our industrial case studies, we wanted to explore how teams apply test automation and which testing activities are involved. The investigation also included investigating which other parts in the mobile application development process that could be automated.

Testing techniques: Our literature review also showed that there are several methods and procedures that can be used to test mobile applications under the categories of unit testing, functional UI testing, performance testing, and usability testing [17, 18, 19]. Additionally, several techniques have been introduced to test context-aware applications. A mobile application is considered to be context-aware when it is aware of the computing environment in which it runs and consequently adapts to changes in that context such as user, time, task, and physical location. For instance, Sama et al. [20] defines a new model based on finite-state machine for the detection of faults of incorrect adaptation logic and asynchronous updating of context information. In another study by Amalfitano et al. [4], an approach based on the definition of reusable design patterns is presented for automatic generation of test cases. In general, context-aware mobile applications are known to be very challenging applications to test [1], [4], [20]. In our industrial case studies, we investigated the type of testing techniques applied and compared them with those described in the research literature. We also included in our analysis and discussion of findings to what extent these techniques are appropriate within the domain of the sorts of mobile applications under test that we observed.

3 Research Methodology

We begin this section with a general discussion about the case study methodology and then we provide detailed information about the methodology we used in our study.

3.1 Case Studies

According to [21], case studies represent an empirical enquiry, investigating a contemporary phenomenon within its real life context, especially when the boundaries between phenomenon and context cannot be clearly specified. Case studies provide key value and deeper understanding of phenomena under study within real-world settings [22]. Additionally the qualitative nature of case study research focuses more on the natural setting of a phenomenon under investigation and produces detailed qualitative information, as richer answers can be obtained from study subjects [23, 24].

Although the case study research method was originally used for exploratory purposes, they can be used for explanatory and descriptive purposes as well [20]. Case study methodology is based on systematic way for looking at events, collecting and analysing data and reporting results [25, 26]. It is highly recommended that case studies rely on several sources of data and evidence [21], [25], [27]. In case of exploratory case study research, it is important to include industrially-based cases because the context can play an important role in defining an emerging theme or theory [25], [28].

As our case study was expected to fill the informational gaps identified from the literature, we chose to take a qualitative approach to investigating the mobile application testing techniques and challenges in real companies. Such an approach can increase our understanding about how exactly mobile applications are tested at industrial contexts and whether special care is being taken to address specific peculiarities for mobile applications such as life-cycle conformance and context awareness.

3.2 Case Study Design

The design of this case study is based on multiple-case holistic design. Results and evidence from multiple-case designs is more convincing and compelling and thus, the overall study is considered to be more robust as compared to single-case design [21]. However, selecting multiple-case design requires considerably more extensive resources and time. This case study investigates and draws conclusions from four different industrial cases. Each single case represents a software development company in the area of mobile and smartphone applications in Palestine. Figure 1 shows our approach to multiple-case study design, which is inspired by [21].

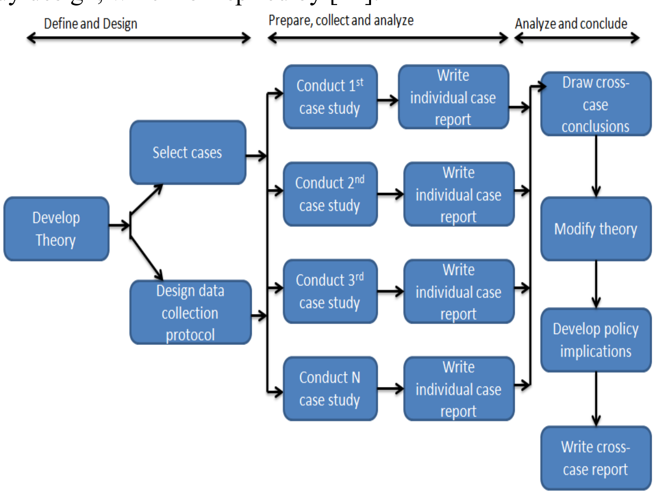

Fig.1Multiple case-study design [21]

It is highly recommended to define a theoretical frame of reference that makes the context of the study clear to those who conduct the research and to those who review the results. Nevertheless, since theories in general are underdeveloped in software engineering, the frame of reference can be expressed in terms of related work that the study builds upon [22].

We have constructed the following three (3) key factors affecting mobile application testing that can be used as a basis for our data collection. These factors emerged from our literature review as well research direction reported by Muccini et al. [1]:

- *Life cycle conformance* – how and to what extent does mobile application testing take into account the life-cycle states of mobile applications?
- *Integration testing* – how and to what extent does testing examine the particularly critical issue of mobile application integration, including app-to-app data exchange, app-to-server data exchange, moving from one app to another, seamless and consistent user interfaces, context-awareness and so on?
- *Testing techniques and test automation* – what are the automated testing techniques employed by industrial mobile application testers and what further opportunities appear to automate additional laborious tasks?

3.3 Research Questions

We have formulated the following key research questions:

- ***RQ1:*** How do teams in industrial mobile application development contexts approach testing?
- ***RQ2:*** What are the testing techniques applied and how effective are they compared to the state of the art?

The research questions later on shaped data collection procedures by focusing on certain aspects and constructing specific interview questions in case study protocol.

3.4 Case Selection and Context

In software engineering, the cases and units of analysis can be anything that is contemporary software engineering phenomenon in its real-life context [22]. For instance, the case study can be an individual, team, process, software company, technology, software product, and so on. In our case study, the cases are the four software development companies in the field of mobile application development and unit of analysis are the development teams that consist of developers and testers working in the companies. Our focus here was on the development team itself, the issues, challenges and techniques they face and apply respectively. The selection of the case study companies was based on availability of team members, willingness of company management, personal relationship with the main author and past working experience. In line with the University's requirements, we have obtained a written consent from each of the companies to participate in this case study and also approval of the IIUM Research Ethics Committee (IREC ID 240) for this study prior to performing the data collection.

3.5 Data Collection Procedures

Data was collected through interviews, observations and focus groups. The three principles of data collection which we used are: (1) use multiple sources of evidence; (2) use of a case study database; and (3) maintaining a chain of evidence [21], [27], [29], [30], [31]. These were all applied in the case study design and its data collection and analysis. The first case (referred as C1) was the only case in which we were granted the permission to collect data through observations.

In the first case study, the data collection strategy consisted of two parts: the first part was a three-month period of observation by the first author. The second part was one-to-one interviews with all development team members. In the second case study, the main researcher conducted one-to-one interviews with all team members. In the third case study, the researcher carried out focus group interviews in which the whole development team was interviewed in one meeting session. In the fourth case, several one-to-one interviews were used. Such diversity in data collection techniques helped us in applying data triangulation across the case studies.

Data collection through observations was done through taking field notes on the behaviour and activities of developers and test engineers in mobile application development teams (i.e. observant without participation). All activities involved in the software testing for mobile applications were investigated which include the following:

- Type of testing activities, techniques and methods used;
- Use of Integrated Development Environments (IDE), testing tools or framework, bug records and bug management processes;
- Strategy on how bugs are identified and recorded;
- Challenges and limitations of testing methods and techniques applied.

For the interviews, the first author conducted a face-to-face as well as email and online interviews with test engineers and other software development team members. The interview protocol was designed prior to data collection (see Appendix). Interviews were semi-structured since they are well suited for this kind of research as discussed by Hancock & Algozzine [32]. Using semi-structured interviews, the researcher 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 the interviewees were invited to express themselves more openly and freely to define the situation from their own perspective. Since the development teams were considerably small (2-6 members in most cases), we interviewed every member in each team involved in our case study. Each interview consisted of three parts and lasted for half an hour. In the first part the interviewees were given an introduction of study purpose and why they were selected. In second part, the interviewees were asked to give brief introduction about their work and experience. In the final part, the interviewees answered the interview questions defined in the protocol.

We also conducted a focus group discussion in order to get more insights, different data patterns and to apply data triangulation [33]. The groups are focused because the individuals gathered have common experience or share common views [34]. Since the group interviewed represented the whole development team and were small in size (4 persons), we treated the groups as adjuncts of interviewing individuals [35]. Our role during these session was as a moderator: to induce all group members to express their opinions but with little, if any, direction [35]. During the session, we put attention to one person at a time while still being appropriately respectful to the others.

Data collected during observation, interviews and focus group interviews needs to be maintained in a manner that can be easily retrieved and traced by other researchers and to maintain chain of evidence [21, 22]. All notes taken were stored in documents and spread sheets recorded using an identification code[1]. Later during thematic coding, sentences were given special codes and were also linked to their original documents. Data collection lasted for a period of four months.

3.6 Analysis procedures

We applied the thematic coding process reported in several qualitative studies [21], [22], [25], [35], [36]. During this process, case study material such as interview answers, observation notes and researcher's reflections were studied. Then, a set of codes were formulated. Next, the whole material was divided in sentences and passages, given appropriate codes, and stored in spreadsheets. Then the spreadsheets, providing our case study database, were analysed together to apply triangulation of data. Triangulation was used to combine the different data sources to corroborate our findings and conclusions and to minimize bias [21].

4 Results

In this section, we presented the results from our multiple case-studies. We included four different cases in our study. To preserve confidentiality, we refer to our case studies as C1, C2, C3 and C4. The first case study was C1, a startup company with a small development team consisting of two (2) developers applying agile and prototype-based processes. The team is working on a new daily task management system that deals with web and smartphone clients. The mobile application was built to support both the Android and iPhone platforms. The application complexity is high due to the fact that the application has many functional requirements and has to work on both online and offline modes with complex data synchronization logic and rules.

The second case, C2 is a well-established and large software development company developing software applications of various types such as cloud-based, web and mobile solutions. This company applies the SCRUM agile method and provides business and software solutions and a wide range of IT services. C2 has three mobile application development teams, each consisting of four (4) members, and offers custom mobile applications targeting personal assistant, navigation, and sales to mention a few.

The third case study, C3, is a software development company applying a strict SCRUM method and has an established and proven history of developing mobile applications as solutions to organizations with different sizes. They have specialized and skilled mobile specialists building mobile applications and applying best practices. This company, in contrast with C2, specializes in developing mobile applications for Android, iPhone and Blackberry platforms. Each platform has special development team of 4 members.

The forth case, C4, is a relatively old and established software development company building web and mobile applications for the enterprise level. What makes this company interesting is that their mobile application team is concerned with developing critical applications in the areas of e-billing and banking services. Such applications require a great attention during development and testing to produce reliable and robust applications. The development process applied in this company is waterfall and the mobile development team consists of three developers. The demographic profile of the companies is shown in Table 1. The companies are small and medium in size and the organization size varied from 3 to 125 people.

TABLE 1. DEMOGRAPHICS OF FOUR CASES

[1] Case study protocol and summary data is available at https://sites.google.com/site/casestudymobileapp/

Case Study ID	Mobile Apps Domain	Dev. Method	Org. Size	Team Size
C1	Task and project management	Prototype based, Waterfall	Small	2
C2	Navigation, search, personal assistant, traffic, sales	SCRUM	Large	3 teams, each has 4 members
C3	Social apps, business apps	SCRUM	Medium	3 teams, each has 4 members
C4	e-Banking, e-Billing, HR	Waterfall	Medium	3

A summary of tools used by mobile application development teams, testing techniques applied and the average knowledge of the properties and conditions of life-cycle (LC) model for all cases are shown in Table 2. On the other hand, Table 3 shows how each case recorded the software bugs. During the four months of data collection, we have conducted observations and interviews with a total of nine (9) developers and a focus-group discussion involving four (4) developers.

TABLE 2. TOOLS AND TECHNIQUES APPLIED

Case ID	Tools used	Platform	Testing Techniques	Average LC Knowledge %
C1	Eclipse	Android, iPhone	Functional testing, performance testing	70%
C2	Eclipse, Android Robotium, Traceview, Node JS, JUnit	Android, iPhone,	Functional testing, performance testing, automated GUI testing, unit testing	52%
C3	Android studio, XCode, Eclipse, JUnit	Android, iPhone, Blackberry	Functional testing, performance testing, unit testing	80%
C4	Eclipse, Team Foundation System	Android, iPhone	Functional testing, performance testing	65%

Note: LC = Life cycle

TABLE 3. BUG RECORDING TOOLS

Case ID	Bug recording tool
C1	MS Excel
C2	MS Excel
C3	MS Excel
C4	MS Team Foundation System

In the following we provide a broad discussion of the development process and environment used, and then we move to more specific areas of life-cycle conformance, integration testing, and testing techniques.

Environment and development process: In general, it was observed that mobile application development is rapid in nature in most of companies studied (except for C4) and teams have to deliver increments of about one week sprints (iterations). On the other hand the development process applied by investigated cases ranged from waterfall, prototype-based to strict SCRUM. The dominant platforms between all companies for mobile applications are Android and iPhone. Regarding the development platform, one of the team leaders in C4 mentioned that:

"building for Android platform is very flexible compared to iPhone platform as we have to invoke many of the OS services. For instance, we are developing an application that keeps track and manages other running applications. This cannot be done on iPhone" – Team leader, C4.

We noticed that a considerably a small amount of testing was done during the initial sprints and that comprehensive testing is done only in the final sprints before major releases. In this case, they did not use the test-first development approach. This issue has been noticed in all four case study companies. This implies that agile best practices such as test driven development are not strictly followed, as claimed by C2 and C3. This also implies that certain types of bugs are sometimes overlooked and missed [16]. In addition, early identification of requirements problems is less likely to occur.

We also noticed that since the mobile development community available online is considered relatively young and small, compared to other technologies such as Java, .NET and PHP, developers in the C1 company very often find it much more time consuming to search for a solution or certain examples of API usage. This is highlighted by one of the developers in C1:

"it is not easy to find solutions online when developing mobile apps, the community and available information is not that large" – developer, C1.

Additionally, it was noticed that testing of mobile apps for their compatibility with different screen sizes consumes much effort and considerable time. Moreover, and since many mobile apps communicate with servers, it was observed for all cases that the testing process of such communication is also time consuming for the developers during the run-time. That is, the developers have to manually check if that data has been transferred to the server correctly and if the mobile app was able to get important data from server in correct form and acceptable time period.

Even though the dominant integrated development environment (IDE) used for Android development is Eclipse, all teams in all cases reported that they will move to *Android Studio* in the near future. This is because Google has announced that it will be the future supported platform for Android development [37].

"We have one team that is using Android studio, soon the rest of the teams will join" – Team leader, C3.

Additionally, in all cases the team size for all mobile application development was considered to be quite small, with team members ranging from two to four people including the team leader. All teams stated that they do not have a specific team role; i.e., having a dedicated team member to perform testing activities, system design, etc. Instead, all team members are responsible for all development activities.

"we do not have specialty in our team, all team members perform all various tasks"— developer, C1.

Life-cycle conformance testing: Testing of the life-cycle conformance of a mobile application is important to maintain a robust and reliable mobile application [5], [10], [11]. It is very normal that a mobile application user swaps through applications causing those applications to go through different states in the life-cycle model. The developer has to make sure that no data is lost and resources are released and/or acquired correctly to maintain integrity and correct behaviour of the application.

However, the findings from all cases revealed that none of the developers and test engineers showed deep and comprehensive understanding of the life-cycle properties and models of their mobile applications. All of the interviewees agreed that they do not fully understand the properties; states and conditions for life-cycle models. Moreover, we did not find any specific technique being used to test life-cycle conformance in applications in all studied cases. The developers of all cases made it clear that they do not apply any specific method or any systematic procedure to test if their application conforms to life cycle properties. In fact, they assume that life-cycle conformance is partially done during manual functional testing of the mobile applications.

An interesting result was noticed when the team members of the third case (C3) stated that they found some inconsistencies between the life-cycle model provided by Android Developers on their official website at [11], and the applications that they have developed. More specifically, the team stated that sometimes the application moves into different state paths than those provided by Android Developers official web site which also made the testing process harder. This result is also confirmed by the study at [5].

"We noticed during the testing of one Android application using LogCat that the application followed a different path than that stated at Android Developer site" – developer, C3.

Integration testing: Development teams in all cases in this study did not apply any specific method or procedure to test inter-application communication. The researcher did not notice any specific care taken when performing this kind of test. Instead, integration testing is partially done manually through the debugger with very basic test scenarios.

Testing techniques and test automation: Manual functional testing is the dominant testing techniques in all of the case studies. In fact, manual functional testing of mobile applications compromises about 80-90% of total testing time and effort across each company. The rest is spent on using unit testing and application performance testing. The researchers also noticed some use of record-and-replay tools such as Robotium, but their use is very minor and limited. Furthermore, a test engineer stated that Robotium is not easy to use unless one has a thorough understanding of Android development in order to construct effective test scripts.

"Android Robotium tool is not easy to be used by non-Android developers, for instance one has to know exactly how to access UI controls' Ids in order to build test scripts" – developer, C2.

In another case (C3), one team reported that sometimes they use a special tool to monitor the application in the field after it is deployed to user. Specific testing techniques for context-awareness were totally absent. Mobile applications that compromise context-aware behaviours are not tested with specific testing techniques such as those identified from the literature [8, 38].

Test automation on the other hand was applied on very limited basis and mostly at automating the execution of unit tests as well as the automation of GUI tests (regression testing) through special testing tool such as Android Robotium. Only one case (C2) reported that they used GUI record-and-replay tool in one of their projects. The rest of the cases implied that they did not use record-and-replay tools as they do not believe they are very much useful for mobile application.

On the other hand, several interviewees suggested other areas for automation such as the design of user interfaces in Android platforms. According to those interviewees, the design of user interfaces is very time consuming and repetitive and that a more advanced user interface design tool is needed to save time:

"Considerable time is wasted when building user interfaces and the GUI designer for Android at Eclipse IDE is not very smart" – team leader, C4.

5 Discussion

This section discusses our key case study results to answer the research questions and compares them with state of the art. Observations made based on the results are described as well as implications for practice and research. Prior work has documented the peculiarities of mobile applications and identified several testing approaches and techniques to address these peculiarities in the areas of life-cycle conformance, context-awareness and integration testing [1], [4], [5], [20], [39]. However, there are no studies that investigate how much development teams in real industrial contexts are aware of such peculiarities, how testing is approached and what are the challenges faced by these teams. In this study, we applied an exploratory multiple-case study approach to increase our understanding of testing techniques applied by industrial teams and the challenges that they are facing.

Several issues can be concluded from the results. First and relating to research questions RQ1 and RQ2*:* It is apparent that testing of mobile applications is approached with a similar mind-set to traditional software testing techniques, with very little, if any, attention given to mobile application specific peculiarities and testing techniques. Furthermore, development teams are more concerned with developing highly responsive mobile applications with fancy user interfaces quickly and in short development cycles.

However, mobile applications are quite different than traditional web and desktop applications and therefore should be approached using different testing techniques[1], [14]. Manual functional testing is not sufficient to produce a robust and reliable mobile application, in particular if that application is considered to be critical [14]. A dedicated complementary testing technique should be applied to cover specific areas for mobile applications:

- For life-cycle conformance testing, the studies at [5], [13] provide a more robust life-cycle models than that defined by Android Developer website. The studies also suggest a more systematic approach for testing as well.
- Regarding testing of context-aware apps, the study at [4] presents an approach based on the definition of reusable event patterns for the manual and automatic generation of test cases for mobile application testing. In another study by [40], the authors target the problem of identifying and exposing faults of buggy context providers and propose a fault tolerant application design. Additionally, the study at [20] defines a new model for detecting faults of incorrect adaptation logic, asynchronous updating of context information and defines algorithms to automatically detect such faults.
- Test driven development (TDD) is one of the core practices in agile methods [16], [41]. This technique however, was not applied by teams using agile development. We strongly recommend careful consideration to applying TDD in mobile application testing, to improve early phase testing, to identify requirements problems early, and to reduce testing costs later in development [11].

Second, development teams should develop a specific testing strategy that is appropriate to the application under test. Such a strategy should contain appropriate testing techniques to address all aspects of mobile application and not only the functional part. For instance, test engineers should start writing special test scripts early during the analysis phase of each sprint. These testing scripts can address life-cycle conformance, test integration as well as context-awareness, depending on the domain and testing requirements of the application. In addressing life-cycle conformance, development teams should be aware that online documentation and models are not always accurate and therefore they should look for more precise models available in the literature [5], [13].

Third, it was noticed that the online documentation, forums, and community available on the Internet to assist mobile application developer are still relatively small and much less mature compared to other mature and large communities such as web application development. However, many developers now assume that good advice and solutions will be available to them for mobile application development as they are for traditional web and desktop development tools and APIs.

As a consequence when developers go online to search for an API (Application Programming Interface), solutions, or other community assistance, it takes a considerable amount of time to find a reliable solution for their problem. Such possible delay should be taken into account during sprint planning. These online resources will improve over time, but may then also suffer the same problem of more traditional technologies of difficulty in finding solutions or identifying good vs bad suggested solutions. Research into support for enhanced searching and locating high quality solutions from developer community forums, integrated into IDEs, would greatly benefit mobile application development teams.

6 Threats to Validity

Ensuring validity and reliability in a case study provides robustness and confidence conclusions and results [22]. Validity and reliability strategies were applied from the beginning in our study. The three criteria used to reach high level of rigor in our case studies are construct validity, external validity, and reliability [19]:

- Construct validity: having correct operational measures for the concepts being studied;
- External validity: identifying the domain for which the study findings can be generalized; and
- Reliability: Demonstrating that data collection procedures can be repeated with same results.

As suggested by Yin [21], strategies outlined in Table 4 were applied to ensure case study validity. The internal validity was not considered due to the type of this study that is exploratory in nature.

TABLE 4. STRATEGIES TO MAINTAIN CASE STUDY VALIDITY

Test	Tactic	Strategy
Construct validity	- Use multiple sources of evidence - Establish chain of evidence	- Review evidence from observations, interviews and focus groups - During thematic coding, sentences were numbered and also linked to their source documents. The same is done for themes
External validity	- Use of theory	- Theory was developed based on literature research
Reliability	- Use case study protocol - Develop case study database	- A case study protocol was created and used - case transcripts, observations, reflections, and spread sheets were created

7 Conclusions and Future Research

This paper presents an exploratory multiple case-studies that aim to investigate how mobile application testing is approached in real world industrial contexts. Our industrial case studies can be characterized as the first attempt to explore and investigate how appropriate are such mobile application testing techniques to the challenges faced by industrial teams. We studied four mobile application development companies to see how they approach testing and the particular issues developers face when testing mobile applications.

We found that in virtually all cases we studied, developers and test engineers lack sufficient knowledge and skills with testing techniques and tools on how to develop or test a mobile application that conforms to life-cycle properties and models. Additionally, we found no awareness on how to test specifically mobile application inter-application communications issues, known as application integration testing. We also discovered that there is no formal testing approach or methodology that can facilitate the development team to systematically test a critical mobile application. Instead, and in almost all cases, testing mostly relies on manual functional testing through the user interface. Automation of issues of lifecycle conformance, application integration, and other testing are seldom performed using automated testing tools. We found that the teams we studied were mostly focused on producing mobile applications that are highly responsive, having fancy or elegant user interfaces, and being delivered in short development cycles.

The findings from our study showed the absence of specialized testing techniques being used in four industry case studies to test specific and important issues of mobile applications such as life-cycle conformance and integration testing. Additionally, our case study showed that traditional and inappropriate testing methods are often being applied to test mobile applications. In future work, we intend to design and implement a comprehensive testing framework for mobile applications.

Acknowledgment

This research was funded by the Ministry of Higher Education Malaysia under FRGS research grant (FRGS14-125-0366). We would like to thank the team leader and developers in the four companies who have participated in our case study.

A. Appendix (Interview questions protocol)

Date:	Venue:	Interviewer:	Interviewee:

1. **(RQ1): What are the factors and conditions that affect the application overall quality and robustness? How can these factors be measured?**
   a. What is the current software process followed in the development of mobile applications?
   b. Who is responsible of each phase?
   c. What are the specific activities of each phase?
   d. What are the IDEs used in development?
   e. What kind of testing techniques applied?
   f. How is testing done exactly? And how effective are they?
   g. Do you think that mobile applications are different than traditional desktop or web applications, and thus need specific testing techniques?
2. **(RQ2): What are properties and specification of testing framework that can increase product quality as well as the efficiency and productivity of test engineers?**
   a. Is there any testing framework or tool applied?
   b. If yes, how do you evaluate testing framework efficiency?
   c. What are the limitations of current test tools applied?
   d. What are the wish lists for testing framework?
   e. Where does testing automation can be best fit?
   f. How long does it take to perform testing activities?
3. **(RQ3): How to automatically verify if a mobile application confirms to life cycle model?**

a. Are you aware of application life cycle?
b. What are the procedures taken to assure life cycle conformance?
c. How application conformance to life cycle model is verified and tested (if any)?

4. **(RQ4): What are the activities that can be automated in order to increase the efficiency of test engineers?**
   a. What are the laborious tasks that should be automated?
   b. Can automation be applied in other parts of development process, such as requirement elicitation?
5. **(RQ5): Can traditional integration testing techniques be applied on mobile application testing?**
   a. Is inter-application communication tested? Or application is treated only as isolated?
   b. What integration testing techniques are applied?
   c. Is there specific software faults related to inter-application communication?

References

[1] Muccini, H., A. Di Francesco, and P. Esposito. *Software testing of mobile applications: Challenges and future research directions*. in *7th International Workshop on Automation of Software Test (AST),*2012.

[2] *Smartphone OS Market Share, Q2 2014*. 2014 [cited 2014 22/10/2014]; Available from: http://www.idc.com/prodserv/smartphone-os-market-share.jsp.

[3] Payet, É. and F. Spoto, *Static analysis of Android programs.* Information and Software Technology, 2012. **54**(11): p. 1192-1201.

[4] Amalfitano, D., A.R. Fasolino, P. Tramontana, and N. Amatucci. *Considering Context Events in Event-Based Testing of Mobile Applications*. in *Software Testing, Verification and Validation Workshops (ICSTW), 2013 IEEE Sixth International Conference on*. 2013.

[5] Franke, D., S. Kowalewski, C. Weise, and N. Prakobkosol. *Testing Conformance of Life Cycle Dependent Properties of Mobile Applications*. in *IEEE Fifth Int'l Conf. on.Software Testing, Verification and Validation (ICST),* 2012.

[6] Zein, S., N. Salleh, and J. Grundy, *A Systematic Mapping Study of Mobile Application Testing Techniques.* Journal of Systems and Software: Under review 2015.

[7] Canfora, G., F. Mercaldo, C.A. Visaggio, M. D'Angelo, A. Furno, and C. Manganelli. *A case study of automating user experience-oriented performance testing on smartphones*. in *Software Testing, Verification and Validation (ICST), 2013 IEEE Sixth International Conference*. 2013.

[8] Ravindranath, L., J. Padhye, S. Agarwal, R. Mahajan, I. Obermiller, and S. Shayandeh, *AppInsight: mobile app performance monitoring in the wild*, in *Proceedings of the 10th USENIX conference on Operating Systems Design and Implementation*. 2012, USENIX Association: Hollywood, CA, USA. p. 107-120.

[9] Losada, B., M. Urretavizcaya, J. Lopez, and I. Castro, *Combining InterMod agile methodology with usability engineering in a mobile application development*, in *Proceedings of the 13th International Conference on Interacción Persona-Ordenador*. 2012, ACM: Elche, Spain. p. 1-8.

[10] Lee, W.-M., *Beginning Android 4 application development*. 2012: Wiley. com.

[11] *Processes and Application Life Cycle*. 2014 [cited 2014 March]; Available from: http://developer.android.com/guide/topics/processes/process-lifecycle.html#.

[12] Haseman, C., *Creating Android Applications: Develop and Design*. 2011: Peachpit Press.

[13] Franke, D., C. Elsemann, and S. Kowalewski. *Reverse Engineering and Testing Service Life Cycles of Mobile Platforms*. in *Database and Expert Systems Applications (DEXA), 2012 23rd Int'l Workshop on*. 2012.

[14] Amalfitano, D., A.R. Fasolino, and P. Tramontana. *A GUI Crawling-Based Technique for Android Mobile Application Testing*. in *2011 IEEE Fourth Int'l Conf. on Software Testing, Verification and Validation Workshops (ICSTW)*. 2011.

[15] Larman, C., *Agile and iterative development: a manager's guide*. 2004: Addison-Wesley Professional.

[16] Crispin, L. and J. Gregory, *Agile Testing: A Practical Guide for Testers and Agile Teams*. 2008: Pearson Education.

[17] Amalfitano, D., A.R. Fasolino, P. Tramontana, S. De Carmine, and A.M. Memon. *Using GUI ripping for automated testing of Android applications*. in *2012 Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE),* 2012.

[18] Heejin, K., C. Byoungju, and W.E. Wong. *Performance Testing of Mobile Applications at the Unit Test Level*. in *Third IEEE Int'l Conf. on*. 2009*Secure Software Integration and Reliability Improvement. SSIRI 2009.*.

[19] Harrison, R., D. Flood, and D. Duce, *Usability of mobile applications: literature review and rationale for a new usability model.* Journal of Interaction Science, 2013. **1**(1): p. 1-16.

[20] Sama, M., S. Elbaum, F. Raimondi, D.S. Rosenblum, and W. Zhimin, *Context-Aware Adaptive Applications: Fault Patterns and Their Automated Identification*, IEEE Transactions on Software Engineering, 2010. **36**(5): p. 644-661.

[21] Yin, R.K., *Case Study Research: Design and Methods*. 2009: SAGE Publications.

[22] Runeson, P. and M. Höst, *Guidelines for conducting and reporting case study research in software engineering.* Empirical software engineering, 2009. **14**(2): p. 131-164.

[23] Cresswell, J.W., *Qualitative inquiry and research design: Choosing among five traditions*. 1998, Sage Publications.

[24] Lankshear, C. and M. Knobel, *A handbook for teacher research*. 2004: McGraw-Hill International.

[25] Verner, J.M., J. Sampson, V. Tosic, N.A.A. Bakar, and B.A. Kitchenham. *Guidelines for industrially-based multiple case studies in software engineering*. in *Research Challenges in Information Science, 2009. RCIS 2009. Third International Conference on*. 2009. IEEE.

[26] Eisenhardt, K.M., *Better stories and better constructs: The case for rigor and comparative logic.* Academy of Management review, 1991. **16**(3): p. 620-627.

[27] Yin, R.K., *Applications of case study research (applied social research Methods).* Series, 4th edn. Thousand Oaks: Sage Publications, 2003.

[28] Woodside, A.G. and E.J. Wilson, *Case study research methods for theory building.* Journal of Business & Industrial Marketing, 2003. **18**(6/7): p. 493-508.

[29] Klein, H.K. and M.D. Myers, *A set of principles for conducting and evaluating interpretive field studies in information systems.* MIS quarterly, 1999: p. 67-93.
[30] Lee, A.S., *Case studies as natural experiments.* Human Relations, 1989. **42**(2): p. 117-137.
[31] Lethbridge, T.C., S.E. Sim, and J. Singer, *Studying software engineers: Data collection techniques for software field studies.* Empirical software engineering, 2005. **10**(3): p. 311-341.
[32] Hancock, D.R. and B. Algozzine, *Doing case study research: A practical guide for beginning researchers*. 2006: Teachers College Press.
[33] Creswell, J., *Research design: Qualitative, quantitative, and mixed methods approaches*. 2009: SAGE Publications, Incorporated.
[34] Tashakkori, A. and C. Teddlie, *Sage handbook of mixed methods in social & behavioral research*. 2010: Sage.
[35] Yin, R.K., *Qualitative research from start to finish*. 2010: Guilford Press.
[36] Miles, M.B. and A.M. Huberman, *Qualitative data analysis: A sourcebook of new methods*, in *Qualitative data analysis:* 1984, Sage publications.
[37] *Android Studio*. Available from: http://developer.android.com/tools/studio/index.html.
[38] Lettner, F. and C. Holzmann, *Automated and unsupervised user interaction logging as basis for usability evaluation of mobile applications*, in *Proceedings of the 10th International Conference on Advances in Mobile Computing and Multimedia*. 2012, ACM: Bali, Indonesia. p. 118-127.
[39] Franke, D. and C. Weise. *Providing a software quality framework for testing of mobile applications*. in *Software Testing, Verification and Validation (ICST), 2011 IEEE Fourth Int'l Conference on*. 2011. IEEE.
[40] Bo, J., L. Xiang, G. Xiaopeng, L. Zhifang, and W.K. Chan. *FLOMA: Statistical fault localization for mobile embedded system*. in*, 2011 3rd International Conference on Advanced Computer Control (ICACC),* 2011.
[41] Vu, J.H., N. Frojd, C. Shenkel-Therolf, and D.S. Janzen. *Evaluating test-driven development in an industry-sponsored capstone project*. in *Sixth Int'l Conf. on Information Technology: New Generations, ITNG'09*. 2009. IEEE.