

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

Samer Zein^{1(✉)}, Norsaremah Salleh¹, and John Grundy²

¹ Department of Computer Science, International Islamic University,
Kuala Lumpur, Malaysia

samer.m.zain@gmail.com, norsaremah@iium.edu

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. In this paper, we presented a multiple case-study 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 a 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]. 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, 2]. Therefore, users fully expect that such applications are easy, reliable and safe to use. 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 [3].

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, 4]. Additionally, 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 [5]. These issues motivate the need for better mobile testing approaches, techniques and tools [1, 5].

In a recent study we conducted a systematic mapping study that analyzed 45 empirical studies of mobile application testing techniques and challenges [5] and revealed significant gaps in the current body of research, suggesting several areas for further investigation. Among the issues we identified were most notably the needs for research in real-world industrial contexts, specific testing techniques, life cycle conformance and integration testing of mobile applications.

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. Each industrial case in our study 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 analyzed 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 investigate specific testing challenges and issues faced by industrial teams as well as how such teams currently tackle testing requirements of mobile applications.
- 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.

2 Related Work

Amalfitano et al. [3] present approaches to test context-aware applications and conduct case studies on real mobile applications to evaluate their approaches. In this present case study, 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 [4]. 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 behavior

of application under all cases [4]. Such understanding will ensure that developers can build a reliable and robust mobile applications that operates correctly and is able to maintain data integrity [6–8]. 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 [4, 9]. 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]. Similarly, mobile applications often require communication with other external applications such as social networking sites (e.g. Facebook, Twitter and MySpace) [6].

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 [10]. With the help of test automation, test engineers are more likely to be able to keep pace with developers to maintain agility [11, 12]. Automated UI testing and regression testing [12] 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.

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 [13–15]. Additionally, several techniques have been introduced to test context-aware applications. For instance, Sama et al. [16] 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. [3], 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, 3, 16]. In our industrial case studies, we investigated the type of testing techniques applied and compared them with those described in the research literature.

3 Research Methodology

In this section we discuss the case study methodology we used for our study. According to [17], 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 [18]. 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 [19].

3.1 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 [17]. This case study investigates and draws conclusions from four different industrial cases. Figure 1 shows our approach to multiple-case study design, inspired by [17].

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 [18]. 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?

3.2 Research Questions

We have formulated the following research questions:

- **RQ1:** How do teams in industrial mobile application development contexts approach testing?

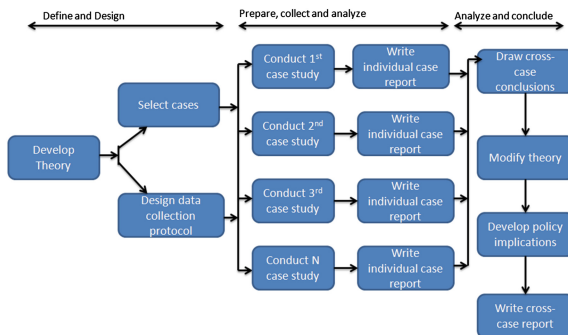


Fig. 1. Multiple case-study design [17]

- **RQ2:** What are the testing techniques applied and how effective are they compared to the state of the art?

3.3 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 [18]. 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. 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.

3.4 Data Collection and Analysis 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 [17, 20]. These were all applied in the case study design and its data collection and analysis.

Data collection through observations was done through taking field notes on the behavior and activities of developers and test engineers in mobile application development teams (i.e. observant without participation). 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. 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 [21]. The groups are focused because the individuals gathered have common experience or share common views [22]. 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 [23].

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 [17, 18]. 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.

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

We applied the thematic coding process reported in several qualitative studies [17, 18, 23, 24]. 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 analyzed 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 [17].

4 Results

In this section, we presented the results from our multiple case-studies. 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 built for 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 fourth 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. 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.

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. Except for fourth case C4 which used MS Team Foundation, the rest of cases used MS Excel to record bugs. During the

Table 1. Demographics of four cases

Case 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 (4 members each)
C3	Social apps, business apps	SCRUM	Medium	3 teams (4 members each)
C4	e-Banking, e-Billing, HR	Waterfall	Medium	3

Table 2. Tools and techniques applied

Case#	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

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.

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. 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 [12].

We also noticed that since the mobile development community available online is considered relatively young and small, 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.

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 [25]. *“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 with no speciality between team members. *“we do not have speciality 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 [4, 6, 7].

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 or applied specific testing techniques to test this important peculiarity.

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 [7], and the applications that they have developed. This result is also confirmed by the study at [4]. *“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.

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 researchers also noticed some use of record-and-replay tools such as Robotium. Furthermore, a test engineer stated that Robotium is not easy to use unless one has a thorough understanding of Android development: *“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 with no application of specific testing techniques suggested by [26, 27].

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.

On the other hand, several interviewees suggested other areas for automation such as the design of user interfaces in Android platforms: *“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 and Threats to Validity

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, 3, 4, 16]. 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.

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, 10]. Manual functional testing is not sufficient to produce a robust and reliable mobile application, in particular if that application is considered to be critical [10]. A dedicated complementary testing technique should be applied to cover specific areas for mobile applications:

- For life-cycle conformance testing, the studies at [4, 9] provide a more robust life-cycle models than that defined by the Android Developer website.
- Regarding testing of context-aware apps, the study at [3] 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 [28],

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 [16] 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 [12, 29]. 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 [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 applications and not only the functional parts.

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.

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]. As suggested by Yin [17], we applied the use of multiple evidence and chain of evidence strategies to address construct validity; use of theory strategy to address external validity case study protocol for reliability. The internal validity was not considered due to the type of this study that is exploratory in nature.

6 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. Instead, and in almost all cases, testing mostly relies on manual functional testing through the user interface. 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. 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.

References

1. Muccini, H., Di Francesco, A., Esposito, P.: Software testing of mobile applications: challenges and future research directions. In: 2012 7th International Workshop on Automation of Software Test (AST). IEEE (2012)
2. Payet, É., Spoto, F.: Static analysis of android programs. *Inf. Softw. Technol.* **54**(11), 1192–1201 (2012)
3. Amalfitano, D., Fasolino, A.R., Tramontana, P., Amatucci, N.: Considering context events in event-based testing of mobile applications. In: 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops (ICSTW) (2013)
4. Franke, D., Kowalewski, S., Weise, C., Prakobkosol, N.: Testing conformance of life cycle dependent properties of mobile applications. In: 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation (ICST) (2012)
5. Zein, S., Salleh, N., Grundy, J.: A systematic mapping study of mobile application testing techniques. *J. Syst. Softw.*: Under review (2015)
6. Lee, W.-M.: *Beginning Android 4 Application Development*. Wiley, Hoboken (2012)
7. *Processes and Application Life Cycle*, March 2014 (2014). <http://developer.android.com/guide/topics/processes/process-lifecycle.html#>
8. Haseman, C.: *Creating Android Applications: Develop and Design*. Peachpit Press, Berkeley (2011)
9. Franke, D., Elsemann, C., Kowalewski, S.: Reverse engineering and testing service life cycles of mobile platforms. In: 2012 23rd International Workshop on Database and Expert Systems Applications (DEXA) (2012)
10. Amalfitano, D., Fasolino, A.R., Tramontana, P.: A GUI crawling-based technique for android mobile application testing. In: 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops (ICSTW) (2011)
11. Larman, C.: *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley, Boston (2004)
12. Crispin, L., Gregory, J.: *Agile Testing: A Practical Guide for Testers and Agile Teams*. Pearson Education, Boston (2008)
13. Heejin, K., Byoungju, C., Wong, W.E.: Performance testing of mobile applications at the unit test level. In: Third IEEE International Conference on Secure Software Integration and Reliability Improvement, SSIRI 2009 (2009)
14. Amalfitano, D., Fasolino, A.R., Tramontana, P., De Carmine, S., Memon, A.M.: 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)
15. Harrison, R., Flood, D., Duce, D.: Usability of mobile applications: literature review and rationale for a new usability model. *J. Interact. Sci.* **1**(1), 1–16 (2013)

16. Sama, M., Elbaum, S., Raimondi, F., Rosenblum, D.S., Zhimin, W.: Context-aware adaptive applications: fault patterns and their automated identification. *IEEE Trans. Softw. Eng.* **36**(5), 644–661 (2010)
17. Yin, R.K.: *Case Study Research: Design and Methods*. Sage Publications, Thousand Oaks (2009)
18. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. *Empirical Softw. Eng.* **14**(2), 131–164 (2009)
19. Woodside, A.G., Wilson, E.J.: Case study research methods for theory building. *J. Bus. Ind. Mark.* **18**(6/7), 493–508 (2003)
20. Lethbridge, T.C., Sim, S.E., Singer, J.: Studying software engineers: data collection techniques for software field studies. *Empirical Softw. Eng.* **10**(3), 311–341 (2005)
21. Creswell, J.: *Research design: qualitative, quantitative, and mixed methods approaches*. Sage Publications, Thousand Oaks (2009)
22. Tashakkori, A., Teddlie, C.: *Sage Handbook of Mixed Methods In Social and Behavioral Research*. Sage Publications, Thousand Oaks (2010)
23. Yin, R.K.: *Qualitative Research From Start To Finish*. Guilford Press, New York (2010)
24. Miles, M.B., Huberman, A.M.: *Qualitative Data Analysis: A Sourcebook of New Methods*. Sage publications, Thousand Oaks (1984)
25. Android Studio. <http://developer.android.com/tools/studio/index.html>
26. Ravindranath, L., Padhye, J., Agarwal, S., Mahajan, R., Obermiller, I., Shayandeh, S., AppInsight: mobile app performance monitoring in the wild. In: *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*, pp. 107–120. USENIX Association: Hollywood (2012)
27. Lettner, F., Holzmann, C.: 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*, pp. 118–127. ACM, Bali (2012)
28. Bo, J., Xiang, L., Xiaopeng, G., Zhifang, L., Chan, W.K.: FLOMA: Statistical fault localization for mobile embedded system. In: *2011 3rd International Conference on Advanced Computer Control (ICACC)* (2011)
29. Vu, J.H., Frojd, N., Shenkel-Therolf, C., Janzen, D.S.: Evaluating test-driven development in an industry-sponsored capstone project. In: *Sixth International Conference on Information Technology: New Generations, ITNG 2009*. IEEE (2009)