

Chapter 48

A Comparative Study of Web Application Testing and Mobile Application Testing

Maryam Ahmed and Rosziati Ibrahim

Abstract Web application have gained increased acceptance over the years in companies and organization as the world move to a global village. Software developers have also grown interest in developing web applications compared to stand-alone application because of the immense benefits it offers such as ubiquity, platform dependence, low cost of support and maintenance, better speed and performance, piracy proof etc. As mobile application emerged in the last decade, attention has been focused on mobile applications by organizations and businesses in order to maximize their profits as much as possible. There has been a rapid increase of software release in the mobile applications store. As the growth of both web application and mobile application increase, the question of quality assurance remains a concern. A comparative study of software testing techniques can be performed to improve the standard of testing of both web and mobile application. This paper therefore reviews the similarity and difference in the testing mechanism.

Keywords Software testing · Mobile application testing · Web application testing · Mobile applications · Web applications

48.1 Introduction

The idea of quality assurance has been in existence even before the existence of software hence quality of manufactured products including software has been a subject of keen interest from past till present. Quality has been an issue as long as

M. Ahmed (✉) · R. Ibrahim
Computer Science and Informtion Technology, Universiti Tun Hussein Onn Malaysia
(UTHM), 86400 Parit Raja, Batu Pahat, Johor, Malaysia
e-mail: hi120055@siswa.uthm.my

R. Ibrahim
e-mail: rosziati@uthm.edu.my

human has been producing product. Software testing is an important and major area of Software Quality Assurance (SQA). Software testing is the process of executing a program with the purpose of finding faults. Testing include effort to find defects but does not include getting solution to fixing the defects. This is the difference between testing and quality assurance as quality assurance is not limited to developing the test plan but also include testing, preventing and fixing the faults found during the process of testing. However, testing can cover different areas such as Specification testing, design testing and implementation testing. Implementation testing which deals with the working system of the software is most time referred to as software testing [1]. Testing cannot be complete or perfect. Having a flawless testing is not realistic [1], as it could take more than a life time to complete even the simple software. Saleh explained the unrealistic nature of complete testing by using a small program of a user filling a text field of 20 characters. This program test will be complete by testing all possible input values. If an assumption of 80 characters is done, then number of possible combinations is 20^{80} . A computer that takes nanoseconds to process one combination, testing will take 10^{11} years to complete all possible combination which is very impossible and unrealistic. Since it is impossible to guarantee an 100 % error-free software, errors that are not detected at the time of development before deployment may be discovered and reported by the end user or in the process of testing for a subsequent release of the software. Despite the imperfection of software testing, testers need to put their best effort in improving the reliability and efficiency of the software as it can affect the well-being of human and their safety.

The current surge in both number and demand for mobile applications has called the attention of test engineers to the testing of mobile application as users are more concerned about the quality and functionality of the software. However, the mobile applications testing has been an issue given the different platforms the applications work on such as the iOS, Windows mobile, and the Google android; whose market has continued to grow rapidly over the years [2, 3]. Also, Hardware complexity and diversity is another challenge in mobile application testing. Unlike the PC environment that operates with one central processing unit ($\times 86$ -micro-processor) on which applications running on the system must be tested (whether voice or data services), mobile phones are built with different types of processors that run at different speed and with different memory usage. Testing web application involve looking into the application and attention should be paid to the browser being used (considering factors like backward and forward buttons) [4]. Connectivity is another issue with the smart mobile device. A mobile device could use the 3G technology, wifi, GPRS all with different level of signal strength [5]. Therefore, testing mobile application requires a testing technique that will be compatible with the software and hardware complexity of the mobile phone.

48.2 Web Application Testing

The advancement of web application is following an evolution similar to that of software systems. Web applications tend to evolve rapidly and pass through frequent modifications as a result of new technological and business opportunities as well as feedback from users [6]. Web application testing can however be more complex than testing standalone or traditional programs due to their synchronized, dispersed and platform-independent environment on which it is been run. Several researches have been going on since the last decade when web application started becoming common around the globe. In 2000, research was carried out to extend data flow testing technique to web application testing. An approach to supporting data flow testing for web application was presented [7]. The approach is such that the white box test artifacts of an application were captured in a Web Application Test Model (WATM) there by treating each component of the web application as an object, while the elements of an HTML or XML were considered as code variables within an object. However, there is need for review of this technique and other similar approach as this is based on the structural testing.

Qian [8] developed a tool for testing web application and to support analysis of the application. They came up with ReWeb and TestWeb. The ReWeb views could be useful in understanding the system organization in terms of navigation path and variable usage while the TestWeb generator and executor of test cases of an application can be used to explore the system to a satisfactory level. Improvement is needed in reducing the manual activities in the process. Automation is required in the area of state unrolling and merging. Bin Zhu, Huaikou Miao and Lizhi Cai (2009) proposed an approach to generating test cases for web application testing using a navigation tree considering the web browser history mechanisms and the user interface facilities such as the back, forward and refresh tool [9]. Another work on web application testing worth mentioning is the development of a cross-browser web application testing tool (Shauvik Roy, 2010). This tool is to find the difference between corresponding elements of a web page opened in different browsers [10]. The tool was evaluated on nine real world applications and result shows that the tool is effective in finding cross-browser issues while keeping the false positive low. Qian Zhongsheng also came with an approach to reducing and optimizing the test case generation from user request traces [8]. A large volume of meaningful user sessions were obtained after purging their irrelevant information by analyzing user logs on the web server. This approach evaluates test cases considering the test coverage ratio only. Many factors need to be considered aside from the test coverage ratio such as the running cost of each test case (CPU time), loading time, time to save the test case and the influence of different test criteria on a test case. The Table 48.1 summarizes the state of the art of web application testing.

Table 48.1 Web application testing techniques

Author	Topic	Methodology/ Contribution	Comment
Chien-Hung Liu David C. Kung Pei Hsia Chih- Tung Hsu [7]	Structural testing of web applications	Web application test model (WATM)	This approach was based on Structural or white box testing
Fillipo Ricca Pallo Tonella [6]	Analysis and testing of web applications	ReWeb and TestWeb	Involves manual processes, therefore need to review to automate processes
Bin Zhu Huaikou Miao Lizhi Cai [4]	Testing a web application involving web browser interaction	Navigation tree and put into consideration web browser history and user interface	Other factors affect the testing of web aside from these two factors
Shauvik Roy Choudhary Husayn Versee Alessandro Orso [10]	A cross-browser web application testing tool	Tool to find the difference between two different browsers	From the result, it was effective in finding cross browser difference
Qian Zhongsheng [8]	Test case generation and optimization for user session-based web application testing	To reduce and optimize test cases generation from user request traces	More factors need to be considered

48.3 Mobile Application Testing

Mobile application also known as Mobile Applications can be used to define the applications that are developed for handheld smart devices such as the mobile phones, etc. These applications can be pre-installed applications that come with the phone straight from the manufacturer or it can be downloaded from the web or app store to enhance better functionality of the smart device. These devices work with powerful application for user consumption and for this reason increases its complexity. The complexity of the mobile applications comes with a cost [9]. Mobile application testing can be very challenging. Many questions need to be addressed when testing a mobile app. Different framework have been proposed for mobile application testing. A cloud based approach was proposed by Baride and Dutta. This approach was proposed to solve the problem of homogeneous testing of mobile application of different platforms [11]. The success of this work was explained in terms of it automation, adaptability to different mobile environment and actual devices and the complexity of the testing which include performance testing, security testing and synchronization testing. Another mobile testing framework was proposed by Ping et al. This framework was based on the V-model and cloud test mechanism. The criteria to be considered in designing a testing model for mobile application was mentioned to include testing scope in terms of the network connection been used, emulators used in testing and devices used [12].

The developed model is however yet to be implemented. The work of Muhammad Karami (2013) presented an automatic analysis approach for security inspection of android applications. This research was inspired by the scrutiny that some malware examples are only triggered on user behavior or action [13]. Hu Cuixiong proposed a novel technique of bringing android specific bug to light and shows how to construct an effective test automation approach for addressing the bugs detected, hence ensuring the reliability of application running on android platform [5].

Different tools have also been developed to aid mobile application testing. MobileTest (2007) is an automatic testing tool for black box mobile devices. This tool can develop refined, maintainable and reusable test case library for testing system level and application level software on different smart devices [14]. The tool was validated by comparing with the result of the TestQuest. Hermes (2009) is another tool for testing smart mobile applications. The motivation of development of this tool comes from the heterogeneity of the mobile device [15]. Hermes can be used for test writing and a distributed run time for automating test execution and reporting. There is need to look into the cost/benefit ratio to improve the evaluation of this tool. Adaptive Random Testing (ART) 2010: ART was also developed from a black box view of testing mobile application [16]. The motivation was from the fact that mobile application has to deal with user input and constant changing in the device or user environment. MZoltar (2013): Mzoltar offers a vigorous analysis of mobile applications which shows a diagnostic report that makes it easy to comprehend [17]. This approach helps in localizing the bugs in android mobile applications by relying on the Spectrum-based fault localization (SFL). Table 48.2 shows a summary of the state of the art of mobile application testing.

48.4 Difference of Web and Mobile Application Testing

With the high increase of mobile users and the rate at which the internet is being accessed, there is need to bridge the gap in the quality of mobile and web applications. There are different factors to be considered in understanding the adaptability of the two testing techniques. These factors affect the way at which these applications are developed and tested. Such factors include and not limited to: Life conformance or event trigger mode, GUI (GUI of the Input and Output system), Data Synchronization or Network Management, Power Management and Memory management.

48.4.1 Event Context/Life Conformance

In web application running on a desktop, application life cycle is dependent on the operating system. The operating system handles the states of the event the application is passing through and it safeguards the precise comportment of the

Table 48.2 Mobile application testing techniques

Author	Topic	Methodology/ Contribution	Comment
Srikanth Baride Kamlesh Dutta [11]	A cloud based software testing paradigm for mobile applications	Cloud based approach to solve the problem of heterogeneity of mobile devices	This approach did not cover the issue of testing connectivity
Tan Ping Ping Hamizan Sharbini I Wee Bui Lin [12]	Designing a mobile application testing model	V-model and cloud base mechanism	An ongoing research on mobile testing framework
Mohammad Karami Mohamed Elsabagh Parnian Najafborazjani Angelos Stavrou [13]	Behavioral analysis of android applications using automated instrumentation	An automated dynamic analysis approach for Security inspection of android applications	There is need for support for more complex interfaces such as OpenGL and gesture views
Cuixiong Hu Iulian Neamtiu [5]	Automating GUI testing for android applications	An approach to test android application with focus on the GUI	There were exceptions in the result due to bugs that do not fall under activity/event/type categories
Jiang Bo Long Xiang Gao Xiaopeng [14]	MobileTest: A tool supporting automatic black box test for software on smart mobile devices	MobileTest took to event based approach to simplify generation of test cases	This helps in reducing the complexity of smart mobile testing while still being effective
Sakura She Sasindran Sivapalan Ian Warren [15]	Hermes: a tool for testing mobile device applications	Hermes an automated test execution and reporting tool	More research to validate and refine the tool especially in the area of reporting
Zhifang Liu Xiaopeng Gao Xiang Long [16]	Adaptive random testing of mobile application	Adaptive random test (ART)	The experimental result of this tool shows that it reduces time taken to find first defect and reduces number of test cases generated.
Pedro Machado José Campos Rui Abreu [17]	MZoltar: automatic debugging of android applications	MZoltar localizes the bugs in android mobile applications by relying on the Spectrum-based fault localization (SFL)	This approach is unique in identifying potential defects quickly through diagnostic report

application at all statuses. This is not the case in smart mobile operating system such as iOS, Android. Due to limited resources and complexity of the system, the operating systems cannot retain the comprehensive state of an application at any time when there is a change in the state of the life cycle. The application therefore takes care of itself to avoid data loss in case the application is paused or even killed. This makes it a requirement for a mobile application to be life conformance in its design. Testing the life conformance of an application life implies it is responding reasonably to change of state in the operating system such that events like low memory or low battery do not lead to loss of data in the mobile device [18–20].

48.4.2 GUI

Testing GUIs on a general note is a tough and demanding task for so many reasons: Firstly, the input space has a countless and potentially unlimited number of combinations of inputs and states of system outputs, generating test cases then becomes tougher. Secondly, even simple GUIs possess an enormous number of states which are also due to interaction with the inputs. Also, many complex dependencies may hold between different states of the GUI system, and/or between its states and inputs [21]. Web application GUI testing still seems easier. Users of desktop GUI applications might be expected to refer to documentation or lessons to completely understand the usage of the applications. Unlike mobile device, mobile applications are expected to have a simple and intuitive user interface where most, if not all, usage scenarios of an application should be evident to the average user from the GUI [22].

48.4.3 Network Management

The emphasis of present research is tending towards planning and building of network and infrastructures and applications for mobile system. Testing software in relation to network connection now remains a concern. This is becoming a hindrance in the evolution of mobile computing since the development of smart mobile application is challenging owing to the scarcity in computational resources. The change in location and network can also lead to data loss and poor functionality. This implies that a change in network or location can mean a small shift away from the servers that is being used toward a new one [23].

Table 48.3 Adaptability of mobile and web application testing

Features	Mobile	Web	References
GUI	Input: touch screen and key board	Input: key board	Hu [5], She et al [15], Chang et al [17], Belli [20], Yang et al. [26]
	Output: small screen	Output: larger screen	
Life cycle	Life cycle is dependent on application since system resources are limited	Operating system takes care of the life cycle of an application	Amalfitano et al. [8], Bo et al. [14], Franke et al. [21], She et al. [15]
Performance (Memory, storage and network management)	Mobility allow easy change in location and therefore could result in network strip off for short distance network	Network connectivity not frequently cut-off	Zhu et al. [4]
Power management	Most mobile applications tend to consume much power, hence need for power management mobile software	Power management under control	Zhang et al. [24], Ismail et al. [27]

48.4.4 Power Management

As users tend to use vital and complex applications, low power consumption becomes a requirement in building mobile applications. Developers of smart mobile device (hardware and software) have considered power-saving features, such that applications can vigorously regulate their power ingestions depending on the required functionality and performance. Software developers therefore need to understand the implication of building a high power-hungry application and should put into consideration the in-built feature of power management in mobile device [24] (Table 48.3).

48.5 Conclusion

While the web has taken over businesses and individual life, the mobile is however anticipated to surpass as the world platform for local and internet applications in the nearest future. As the functionality of the mobile and web applications increase so is the complexity and hence the complexity of the testing technique. There is need to fill the gap in the testing of these two important applications area. The difference in performance and testing could be viewed from the event context, GUI interface, network management, power and memory management. There are present works that have been looking into these factors as highlighted in this paper;

however, there is a strong need for more research in building more effective testing technique that can handle both the web and mobile application testing.

Acknowledgments This research is supported under the Graduate Research Incentive Grants (GIPS), vote 1256, Universiti Tun Hussein Onn Malaysia.

References

1. Saleh, K.A.: Software engineering. J. Ross Publishing Inc. p. 244
2. Franke, D., Weise, C.: Providing a software quality framework for testing of mobile applications. 2011 fourth IEEE international conference on software testing, verification and validation, 431–434 (2011). doi:[10.1109/ICST.2011.18](https://doi.org/10.1109/ICST.2011.18)
3. Wasserman, A. I.: Software engineering issues for mobile application development. In: Proceedings of the FSE/SDP workshop on future of software engineering research – FoSER-10, p. 397. (2010). doi:[10.1145/1882362.1882443](https://doi.org/10.1145/1882362.1882443)
4. Zhu, B., Miao, H., and Cai, L.: Testing a web application involving web browser interaction. 2009 10th ACIS international conference on software engineering, artificial intelligences, networking and parallel/distributed Computing, 589–594 (2009). doi:[10.1109/SNPD.2009.59](https://doi.org/10.1109/SNPD.2009.59)
5. Hu, C.: Automating GUI testing for android applications, (Section 4), pp. 77–83
6. Ricca, F., Tonella, P.: Analysis and testing of web applications. In: Proceedings of the 23rd international conference on software engineering. ICSE 2001, 25–34 (2001). doi:[10.1109/ICSE.2001.919078](https://doi.org/10.1109/ICSE.2001.919078)
7. Kung, C. L. D. C., Box, P. O.: Structural testing of web applications Chih-Tung Hsu, 84–96
8. Qian, Z.: Test Case Generation and Optimization for User Session-based Web Application Testing. J. Comput **5**(11), 1655–1662 (2010). doi:[10.4304/jcp.5.11.1655-1662](https://doi.org/10.4304/jcp.5.11.1655-1662)
9. Zhifang, L., Bin, L.: Test automation on mobile device (2007), 1–7 (2010)
10. Roy Choudhary, S., Versee, H., Orso, A.: A cross-browser web application testing tool. 2010 IEEE international conference on software maintenance, 1–6 (2010). doi:[10.1109/ICSM.2010.5609728](https://doi.org/10.1109/ICSM.2010.5609728)
11. Baride, S., Dutta, K.: A cloud based software testing paradigm for mobile applications. ACM SIGSOFT Softw. Eng. Notes **36**(3), 1 (2011). doi:[10.1145/1968587.1968601](https://doi.org/10.1145/1968587.1968601)
12. Ping, T. P., Sharbini, H., Lin, W. B., Tan, V., Mun, W., Julaihi, A. A.: Designing a mobile application testing model, 255–260
13. Karami, M., Elsabagh, M., Najafiborazjani, P., Stavrou, A.: behavioral analysis of android applications using automated instrumentation
14. Bo, J., Xiang, L., Xiaopeng, G.: MobileTest: a tool supporting automatic black box test for software on smart mobile devices. Second international workshop on automation of software test (AST '07), 8–8. (2007). doi:[10.1109/AST.2007.9](https://doi.org/10.1109/AST.2007.9)
15. She, S., Sivapalan, S., Warren, I.: Hermes: a Tool for Testing Mobile Device Applications. Australian Softw. Eng. Conf. **2009**, 121–130 (2009). doi:[10.1109/ASWEC.2009.17](https://doi.org/10.1109/ASWEC.2009.17)
16. Liu, Z., Liu, B., Gao, X.: SOA based mobile application software test framework. 2009 8th international conference on reliability, maintainability and safety, 765–769 (2009). doi:[10.1109/ICRMS.2009.5270087](https://doi.org/10.1109/ICRMS.2009.5270087)
17. Machado, P., Campos, J., & Abreu, R.: MZoltar: automatic debugging of Android applications. In: Proceedings of the 2013 international workshop on software development lifecycle for mobile - demobile 2013, 9–16 (2013). doi:[10.1145/2501553.2501556](https://doi.org/10.1145/2501553.2501556)
18. Amalfitano, D., Fasolino, A. R., Tramontana, P.: A GUI crawling-based technique for android mobile application Testing. 2011 IEEE fourth international conference on software testing, verification and validation workshops, 252–261. (2011). doi:[10.1109/ICSTW.2011.77](https://doi.org/10.1109/ICSTW.2011.77)

19. Amalfitano, D., Fasolino, A. R., Tramontana, P., Amatucci, N.: Considering Context Events in Event-Based Testing of Mobile Applications. 2013 IEEE sixth international conference on software testing, verification and validation workshops, 126–133. (2013). doi:[10.1109/ICSTW.2013.22](https://doi.org/10.1109/ICSTW.2013.22)
20. Belli, F.: Finite state testing and analysis of graphical user interfaces. In: Proceedings 12th international symposium on software reliability engineering, 34–43 (2001). doi:[10.1109/ISSRE.2001.989456](https://doi.org/10.1109/ISSRE.2001.989456)
21. Franke, D., Kowalewski, S.: Testing conformance of life cycle dependent properties of mobile applications, (2012). doi:[10.1109/ICST.2012.36](https://doi.org/10.1109/ICST.2012.36)
22. Yang, W., Prasad, M. R., Xie, T.: A grey-box approach for automated gui-model generation of mobile applications
23. Satoh, I., Society, I. C.: A testing framework for mobile computing software, 29(12), 1112–1121 (2003)
24. Zhang, L., Dick, R. P., Mao, Z. M., Wang, Z., Arbor, A.: Accurate online power estimation and automatic battery behavior based power model generation for smartphones, 105–114
25. Chang, T., Yeh, T., Miller, R. C.: GUI testing using computer vision, (Figure 1), 1535–1544 (2010)
26. Yang, Y. J., Kim, S. Y., Choi, G. J., Cho, E. S., Kim, C. J., Kim, S. D.: A UML-based object-oriented framework development methodology. In: Proceedings 1998 asia pacific software engineering conference (Cat. No.98EX240), 211–218. doi:[10.1109/APSEC.1998.733722](https://doi.org/10.1109/APSEC.1998.733722)
27. Ismail, M.N., Ibrahim, R., MdFudzee, M.F.: A survey on content adaptation systems towards energy consumption awareness. *Adv. Multimedia* **2013**, 3 (2013)