Journal of Engineering Science and Technology
Special Issue on ISSC'2016, April (2017) 86 - 93

THE PERCEIVED USABILITY OF AUTOMATED TESTING TOOLS FOR MOBILE APPLICATIONS

AZHAM HUSSAIN*, HAMIDAH ABDUL RAZAK,
EMMANUEL O. C. MKPOJIOGU

Human-Centered Computing Research Lab, School of Computing,
Universiti Utara Malaysia, 06010 Sintok, Malaysia,
*Corresponding Author: azham.h@uum.edu.my

Abstract

Mobile application development is a fast-emerging area in software development. The testing of mobile applications is very significant and there are a lot of tools available for testing such applications particularly for Android and iOS. This paper presents the most frequently used automated testing tools for mobile applications. In this study, we found that Android app developers used automated testing tools such as JUnit, MonkeyTalk, Robotium, Appium, and Robolectric. However, they often prefer to test their apps manually, whereas Windows apps developers prefer to use in-house tools such as Visual Studio and Microsoft Test Manager. Both Android and Windows apps developers face many challenges such as time constraints, compatibility issues, lack of exposure, and cumbersome tools, etc. Software testing tools are the key assets of a project that can help improve productivity and software quality. A survey method was used to assess the perceived usability of automated testing tools using forty (40) respondents as participants. The result indicates that JUnit has the highest perceived usability. The study's result will profit practitioners and researchers in the research and development of usable automated testing tools for mobile applications.

Keywords: Usability, Mobile, Testing, Automated tools.

1. Introduction

As mobile applications become more and more important for businesses and consumers expecting higher quality apps for mobile devices, testing teams need to adapt and get ready to verify and evaluate mobile apps as part of their projects. Evaluating the quality of mobile devices is especially resource-intensive and time-consuming, especially as automating tests for mobile devices can be quite complex.

There are various tools and online resources to help build tests for mobile devices, record and run automated UI and unit tests for mobile apps and code libraries, as well as help the developers test responsive and fluid web interfaces.

Automated testing simplifies the testing effort with a minimum set of scripts. The Automation Tester is a technical specialist (a tester, quality assurance specialist or software developer), who enables the creation of software, debugging and support of operational state test scripts, test suite and tools for automated testing. Test Script is a set of instruction, which automatically check for certain piece of software. Test Suite is a combination of test scripts, to test a particular piece of software and Test Run is a combination of test scripts and test suites which depends on the objectives pursued and a possible tool for automated testing [1,2-3]. The test includes testing objective, methods of testing new functions, total time and resources required for the project, and testing environment. The test strategy gives description about the product risks at test level and suggests which types of test to be performed and, which entry and exit criteria apply [4-6].

There are some automated mobile web testing tools such as MonkeyTalk, Robotium, and Appium etc. which has some pros and cons. MonkeyTalk provides support for iOS as well as Android devices, where it has 3 component, MonkeyTalk IDE, MonkeyTalk agent, MonkeyTalk scripts. MonkeyTalk IDE is built on top of eclipse which has functionality keys like Record/Replay and it communicate with MonkeyTalk browser and creates test case with the help of MonkeyTalk agent. MonkeyTalk agent triggers event when users perform any action on real device, and the IDE adds that command into the script file. MonkeyTalk supported in both Android and iOS focuses on mobile apps. Robotium is a framework for running automated Android tests. It is popular because of its ease of use, fast execution, and readability of tests, and also can develop powerful test cases with minimal knowledge of the application under test [1,7]. It is an open-source test framework for writing automatic gray box test cases for Android applications. Robotium is used in only android apps and has record/ reply functionality with limited features. It is limited to Android. Appium is an open source test automation framework for use with native, hybrid and mobile web apps. It drives iOS and Android apps using the WebDriver protocol. Robolectric also has a framework that can write unit tests and run them on a desktop JVM while still using Android API [1-8]. In this study, the perceived usability of automated testing tools for mobile apps was evaluated. There is scarcity of studies related to the usability of automated testing tools for mobile apps. This warranted the current research so as to ascertain the extent of the perceived usability of the commonly used mobile apps' automated testing tools.

Nowadays automated tests are used during almost every testing process. This is not surprising, as properly organized automated testing greatly reduces time needed for a testing process, excludes errors and omissions in tests execution caused by a human factor. There is a wide choice of tools for automation. Some of them are free, some are rather expensive. Some automation tools were created years ago; some have just appeared on the market. Each tool is unique and possesses certain characteristics. Wide choice of available automation tools makes it difficult to select the most suitable ones for a project. The problem is that hardly any of the existing tools fully corresponds to project requirements. Many different kinds of computers have appeared recently, their appearance started rapid development of software products. The most amazing is evolution of mobile devices; they significantly differ from

common personal computers by characteristics, ways and conditions of interaction with them. Consequently, smart phones and tablets require special mobile applications that differ from desktop ones.

The remaining part of this paper is organized as follows: In section 2, the methodology of the study is presented. Section 3 discusses the result and findings from the study. Lastly, the conclusion is presented in section 4.

2. Method

In this study, a survey approach was used to ascertain the perceived usability of automated testing tools. Forty (40) respondents (who were users of automated testing tools as developers and testers) were conscripted into the study to get their opinion on using automated tool for testing the mobile application. Table 1 provides a summary of the selected articles.

Table 1. Selected articles.

S/No	Title	Author and Year
1	Understanding the Test Automation Culture of App Developers	Kochhar et al. [9]
2	Survey on Mobile Automation Testing Tools	Momin [7]
3	Automating GUI testing for Android Application	Hu & Neamtiu [10]
4	MobiGUITAR – A Tool for Automated Model-Based Testing of Mobile Apps	Amalfitano et al. [11]
5	Software Testing of Mobile Applications: Challenges and Future Research Direction	Muccini et al. [12]
6	Adaptive Random Testing of Mobile	Liu et al. [13]
7	A Tool for Testing Mobile Device Applications	She et al. [14]
8	Automated GUI testing on the Android Platform	Kropp & Morales [15]
9	Software Testing Automation Using Appium	Shah et al. [16]
10	Software Testing of mobile applications using scripting Technique: A study on Appium	Singh et al. [17]
11	MonkyTalk Open Source Automation Testing Tool	Gorillalogic [18]
12	MonkeyTalk Test iOS/Android Apps in The Mobile Jungle	Bridgwater [19]
13	Automated concolic testing of smartphone apps	Anand et al. [20]
14	Automated testing with targeted event sequence generation	Jensen et al. [21]
15	Automated generation of oracles for testing user-interaction features of mobile apps	Zaeem et al. [22]
16	A Novel Approach of Automation Testing on Mobile Devices	Nagowah & Sowamber [23]
17	Mobile Software Testing – Automated Test Case Design Strategies	Selvam [24])
18	Automated Software Testing with Traditional Tools & Beyond	Quilter [25]

Eighteen appropriate articles among other articles were searched, selected and used in the study to obtain the information related to the automated tool used during testing.

3. Results and Discussion

The following is a summary of our findings: Android apps developers prefer using standard framework such as JUnit, but they also use Android specific testing tools such as MonkeyTalk, Robotium and Robolectric. However, many Android developers prefer to test their applications manually without the help of any testing framework or tools. Most Windows app developers make use of Visual Studio, Coded UI, Selenium, and Microsoft Test Manager to test their apps. Android and Windows app developers face numerous challenges in testing their apps and in using automated testing tools. These challenges include time constraints, compatibility issues, lack of exposure, cumbersome tools, emphasis on development, lack of organization support, unclear benefits, poor knowledge of tools that are suitable for various applications, lack of experience, and steep learning curve. However, the automated testing tools make the testing process to be faster and also enable more efficient documentation [6,16,18,21-22,29-36].

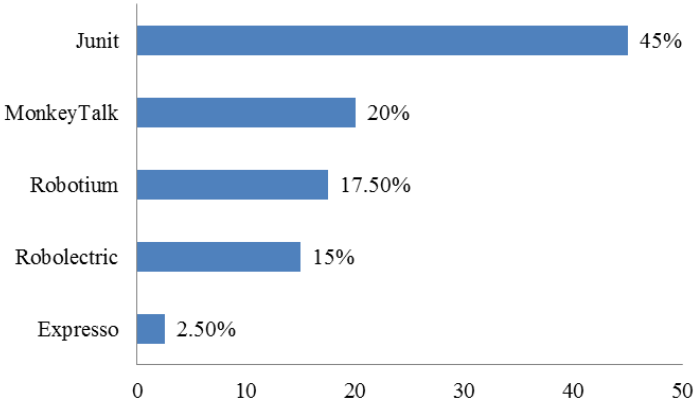

Fig. 1. Automated testing tools' perceived usability.

With regard to the perceived usability of some of the automated testing tools examined, JUnit takes the lead. As can be seen in Fig. 1, the perceived usability of JUnit was the highest (45%). The perceived usability of MonkeyTalk was next (20%), followed by Robotium (17.50%), and then Robolectric (15%). Then automated testing tool with the least perceived usability was Expresso (2.50%). Beside the tool mentioned above, other tools available for mobile application include MonkeyRunner, Ranorex, Appium and UI Automater. However, they are not included in this study due to the recommendation by respondent. Respondents nominated JUnit maybe because it promotes the idea of "first testing then coding", which emphasizes on setting up the test data for a piece of code that can be tested first and then implemented. This approach is like "test a little, code a little, test a little, code a little." It increases the productivity of the programmer and the stability of program code, which in turn reduces the stress on the programmer and the time spent on debugging.

MonkeyTalk also popular` maybe it is an open source mobile app automation testing tool for Android and iOS. MonkeyTalk is a simple-to-use tool which automates real, functional interactive tests for iOS, Android, Web/HTML5, Hybrid and Flex apps. This open source tool can be used for simple 'smoke tests'

or for 'data-driven test' suites on native, mobile, and hybrid apps, real devices or simulators. Robotium also is an open-source test framework for writing automatic gray-box testing cases for Android applications. With the support of Robotium test case, developers can write function, system and acceptance test scenarios, spanning multiple Android activities. Robotium can be used both for testing applications where the source code is available and applications where only the APK file is available.

Based on the standard ISO 9241, HCI handbooks, and existing usability studies related to mobile applications, the usability of automated testing tools could be affected by the following five generic factors [7-9,25,27-28,37-44]: **Efficiency**: Resources expended in relation to the accuracy and completeness with which users achieve goals. **Satisfaction**: Freedom from discomfort, and positive attitudes towards the use of the product. **Learnability**: The system should be easy to learn so that the user can rapidly start getting work done with the system. **Memorability**: The system should be easy to remember so that the casual user is able to return to the system after some period of not having used it without having to learn everything all over again. **Errors**: The system should have a low error rate, so that users make few errors during the use of the system and that if they do make errors they can easily recover from them. Further, catastrophic errors must not occur [25,37-44].

4. Conclusions

A test automation tool automates the usual steps that are involved in a test. There are different types of tools that are suitable for various applications. These tools make the testing process faster and more efficient. This paper reports the survey investigating the perceived usability of automated testing tools for Android and iOS and found that JUnit has the highest perceived usability. Maximizing automation is an effective way of expediting the testing process. It reduces the long-term testing process. The factors such as support for mobile application platforms, script reusability and total cost ownership should be taken into account when selecting automated testing tools. More so, the perceived usability of the automated testing tools should also be a priority when considering the automated tools to use. In this study, the perceived usability of automated testing tools for mobile applications was evaluated. The results revealed that the perceived usability of JUnit was the highest (45%) as afore mentioned, while the perceived usability of MonkeyTalk was next (20%), followed by Robotium (17.50%), and then Robolectric (15%). The automated testing tool with the least perceived usability was Expresso (2.50%). This research report will benefit practitioners (like developers, designers, testers and usability experts) as well as academicians in the research and development of usable automated testing tools for mobile applications. Tests automation is a complex task. It requires thorough preparation and researches. One should also keep up with all the novelties in information technology, applications and test automation tools. All this knowledge is necessary for creating the most effective tests.

5. Acknowledgement

The authors would like to thank Universiti Utara Malaysia for providing financial support for the paper publication under University Grant (SO Code: 12637).

References

1. Keynote.com (2014). Mobile testing. Retrieved June 5, 2014, from http://www.keynote.com/solutions/testing/mobile-testing.
2. Sofokleous, A.; and Andreou, A. (2008). Automatic evolutionary test data generation for dynamic software testing. *Journal of Systems and Software*, 81(11), 1883-1898.
3. Stottlemyer, D. (2001). Automated web testing toolkit: Expert methods for testing and managing web applications. Canada: John Wiley & Sons.
4. Mirzaei, N.; Malek, S.; Pasareanu, C.S.; Esfahani, N.; and Mahmood, R. (2012). Testing android apps through symbolic execution. *ACM SIGSOFT Software Engineering Notes*, 37(6), 1–5.
5. Liu, Y.; Xu, C.; and Cheung, S. (2012). Verifying android applications using java pathfinder. The Hong Kong University of Science and Technology, Technical Report.
6. Momin, S. (2015). Survey on mobile automation testing tools. International *Journal of Application or Innovation in Engineering & Management*, 4(1), 191-193.
7. Milano, D. (2001). Android Application Testing Guide, USA: Packt Publishing Ltd.
8. Gorillalogic.com. (2014). MonkeyTalk open source automation testing tool. Retrieved from at https://www.gorillalogic.com/monkeytalk.
9. Kochhar, P.S.; Thung, F.; Nagappan, N.; Zimmermann, T.; and Lo, D. (2015). Understanding the test automation culture of app developers. *IEEE International Conference on Software Testing, Verification and Validation*, 978(1), 4799-7125.
10. Hu, C.; and Neamtiu, I. (2011). Automating GUI testing for android application. *Automation of Software Test*. HI, USA, 77-83.
11. Amalfitano, D.; Fasolina, A.R.; Tramontana, P.; Ta, B.; and Memon, A. (in press). MobiGUITAR – A tool for automated model based testing of mobile apps. *IEEE Software*.
12. Muccini, H.; and Esposito, D.F. (2012). Software testing of mobile applications: Challenges and Future Research Direction. *7th International Workshop on Automation of Software Test (AST)*. Zurich, Switzerlan, 29-35.
13. Liu, Z.; Gao, X; and Long, X. (2010). Adaptive random testing of mobile. *2nd International Confernce. on Computer Engineering and Technology (ICCET)*, 2, 16-18.
14. She, S.; Sivapalan, S.; and Warren, I. (2009). A tool for testing mobile device applications. *Software Engineering Conference*. Australia, 121.
15. Kropp, M.; and Morales, P. (2010). Automated GUI testing on the android platform. *Testing Software and System*, 67.
16. Shah, G.; Shah, P.; and Muchhala, R. (2014). Software testing automation using appium, International Journal of Current Engineering and Technology, 4(5), 3528-3531.
17. Singh, S.; Gadgil, R. and Ayushi, C (2014). Automated testing of mobile applications using scripting technique: a study on appium. *International Journal of Current Engineering and Technology*, 4(5), 3627-3630.

18. Hussain, A.; Mkpojiogu, E.O.C.; and Hussain, Z. (2015). Usability evaluation of a web-based health awareness portal on smartphone devices using ISO 9241-11 model. *Jurnal Teknologi (Sciences & Engineering)*, 77 (4), 1-5.
19. Bridgwater, D. (2012). Monkey Talk test iOS/android apps in the mobile jungle. Retrieved from http://www.drdobbs.com/open-source/monkeytalk-tests-iosandroid-apps-in-the/232602236
20. Anand, S.; Naik, M.; Harrold, M.J. and Yang, H. (2012). Automated concolic testing of smartphone apps. *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*. New York, USA.
21. Jensen, C.S.; Prasad, M.R.; and Møller, A. (2013). Automated testing with targeted event sequence generation. International Symposium on Software Testing and Analysis (ISSTA). Denmark, 67–77.
22. Zaeem, R.N.; Prasad, M.R.; and Khurshid, S. (2014). Automated generation of oracles for testing user-interaction features of mobile apps. *Proceedings of the 2014 IEEE International Conference on Software Testing, Verification, and* Validation. Washington, DC, USA, 183–192.
23. Nagowah, L.; and Sowamber, G. (2012). A novel approach of automation testing on mobile devices. *International Conference on Computer & Information Science (ICCIS)*.
24. Selvam, R. (2011). Mobile software testing – Automated test case design strategies. *International Journal on Computer Science and Engineering* (IJCSE), 3(4), 1450-1461.
25. Quilter, P. (2011). Automated software testing with traditional tools & beyond. 3(5), 20.
26. Hussain, A.; and Mkpojiogu, E.O.C. (2015b). The effect of responsive web design on the user experience with laptop and smartphone devices. *Jurnal Teknologi (Sciences & Engineering)*, 77 (4), 41-47.
27. Mkpojiogu, E.O.C.; Hashim, N.L.; and Adamu, R. (2016). Observed demographic differentials in user perceived satisfaction on the usability of mobile banking applications. *8th Knowledge Management International Conference (KMICe'16)*. Chiang Mai, Thailand.
28. Hussain, A.; and Mkpojiogu, E.O.C. (2016a). Usability evaluation techniques in mobile commerce applications: A systematic review. *Proceedings of the International Conference on Applied Science and Technology (ICAST'16)*, Kedah, Malaysia.
29. Hussain, A.; and Mkpojiogu, E.O.C. (2016b). A systematic review of usability test metrics for mobile video streaming apps. *Proceedings of the International Conference on Applied Science and Technology (ICAST'16)*. Kedah, Malaysia.
30. Salam, M.A.; Keshk, A.; Ismail, N.; and Nassar, H. (2007). Automated testing of Java menu-based GUIs using XML visual editor. *International Conference on Computer Engineering & Systems ICCES* '07., 313–318.
31. Hussain, A.; and Mkpojiogu, E.O.C. (2015a). An application of ISO/IEC 25010 standard in the quality-in-use assessment of an online health awareness system. *Jurnal Teknologi (Sciences & Engineering)*, 77 (5), 9-13.

32. Berner, S.; Weber, R.; and Keller, R.K. (2005). Observations and lessons learned from automated testing. *Proceedings of the 27th International Conference on Software Engineering*. New York, USA, 571– 579.
33. Weiss, D.; and Zduniak, M. (2007). Automated integration tests for mobile applications in Java 2 micro ed. Lecture Notes in Computer Science, 4439, 478–487.
34. Puhakka, T.; and Palola, M. (2006). Towards automating testing of communicational B3G applications, Mobility '06. *Proceedings of the 3rd International Conference on Mobile Technology, Applications & Systems*. New York, USA, 27.
35. Dustin, E.; Rashka, J.; and Paul, J. (1999). Automated software testing. Addison Wesley Longman.
36. Jiang, B.; Long, X.; and Gao, X.P. (2007). Mobile test: a tool supporting automatic black box test for software on smart mobile devices. *Proceedings of the 2nd International Workshop on Automation of Software Test* (AST'07). IEEE Computer Society, Washington, DC, 8.
37. Hussain, A.; Abdullah, A.; Husni, H.; and Mkpojiogu, E.O.C. (2016). Interaction design principles for edutainment systems: Enhancing the communication skills of children with autism spectrum disorders. Rev. Tec. Ing. Univ. Zulia, 39(8), 45-50.
38. Mkpojiogu, E.O.C.; and Hussain, A. (2017). Usability metrics and methods for public transportation mobile applications: a systematic literature review. Journal of Engineering, Science and Technology.
39. Hussain, A.; Mkpojiogu, E.O.C; and Hassan, F. (2014). Usability dimensions and sub-dimensions for the evaluation of m-learning apps for children: a systematic review. Jurnal Teknologi (Sciences & Engineering).
40. Hussain, A.; and Mkpojiogu, E.O.C. (2015). A systematic review on usability evaluation methods in m-commerce apps. *Journal of Technology, Electronics & Computer Engineering*.
41. Hussain, A.; and Mkpojiogu, E.O.C. (2015). Antecedents to user adoption of interactive mobile maps. *Journal of Technology, Electronics & Computer Engineering.*
42. Hussain, A.; and Mkpojiogu, E.O.C. (2016). Mobile video streaming applications: a systematic review of test metrics in usability evaluation. *Journal of Technology, Electronics & Computer Engineering.*
43. Hussain, A.; Mkpojiogu, E.O.C.; and Hassan, F. (2016). Systematic review of mobile learning applications for children. *Proceedings of the 2nd International Conference on Information and Communication Technology for Transformation* (IC-ICT4T'16). Kota Kinabalu, Sabah, Malaysia.
44. Baker, S.; Au, F.; Dobbie, G.; and Warren, I. (2008). Automated usability testing using HUI analyser. *19th Australian Conference on Software Engineering* (ASWEC), 579-588.