

An Essential Guide to Automated GUI Testing Of Android Mobile Applications

Anureet Kaur

*(Dept of Computer Science, Guru Nanak Dev University, Amritsar)

Abstract:

Mobile platform has taken over all aspects of human activities, and researchers are emphasising on providing efficient and bug free applications running on mobile platform. As the time to market for mobile applications are very short so there is a need for techniques which are automated and provide robust and reliable applications. Although manual testing is the most popular technique for testing graphical user interfaces, but manual testing is often cumbersome and prone to errors. This paper presents a literature review of existing Automated GUI Testing of Android Applications techniques. Because of popularity and open-source nature of android devices the focus of research paper is on techniques used in Android devices.

Keywords — **Automated, GUI, testing techniques, Mobile Applications, Android**

I. INTRODUCTION

Mobile devices are rapidly taking over desktop computers and are becoming a very important part of our life. As the users of the mobile devices are growing, so does the significance of quality app is becoming obligatory. Automated testing is one of the important factors in increasing application quality. The apps written for mobile devices are becoming more and more advanced and complex, adjusting to the constantly improving computational power of hardware. In this paper we focus on reviewing the techniques available for automating the testing of GUI mobile applications on Android platform. This paper first provides a brief introduction to android and its underlying architecture. Then discusses challenges that are faced in testing mobile application and finally provides a comparison of existing techniques for testing based on some parameters. There are many mobile platforms available in market such as Symbian, Research In Motion, BlackBerry OS, Palm OS, PSP, Windows Mobile ,Apple iOS and

Android, But out of all these Android is predicted to become the largest mobile Operating System [1, 2], because of its open-source nature and many programmability features.

A. Android Overview

Android is java based which is used or developing mobile applications and has Linux kernel. Delvik is android compiler which is responsible for converting application code written in java to machine code. The basic use of android is in developing applications running on smart phones which are capable of having touch screen interface.

As shown in Figure 1 below, the Android platform is composed of 4 layers:

Android platform is divided into four layers namely

1. Top Layer is Applications. Applications are running at the top layer of architecture.
2. Second layer is Application Framework layer responsible for providing services to applications which include controlling activities for each application and loading the content provider

defined by each application while restricting data accessibility across applications

3. Third layer is Library/VM layer The Library/VM layer contains static libraries having common system and applications libraries for applications. It also has Android runtime environment which has runtime libraries and the Dalvik virtual machine

4. Bottom layer is the Linux kernel. It contains the OS and software. [3]

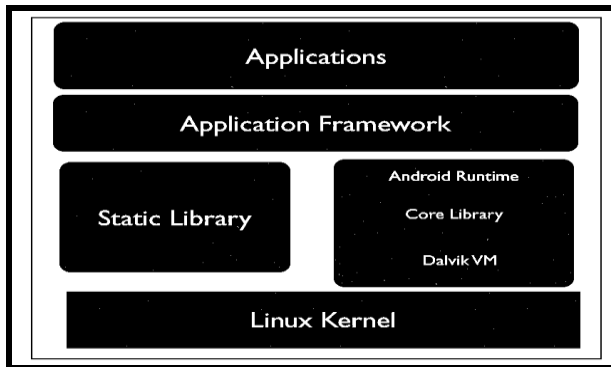


Fig.1 Architecture of Android platform [3]

B. Automated Testing of GUI Android Mobile Application

In GUI based automation testing of android applications, the test scripts are written which mimic or simulate the way user interacts with graphical interface of the application. These interactions are then forming the basis for testing them with actual interactions and then differences are corrected. There are many automation tools which can be capably used to test traditional PC applications or web applications .However, for android it is difficult to use these automated tools because of various reasons such as Performance Factor, Power Factor, Band Factor, Connectivity Factor, Context Factor, Graphic Interface Factor, Input Interface Factor etc. Also due to lack of experience of android developers, android development is liable to to several kinds of multifaceted bugs which cannot be tested automatically.

“Automated GUI testing is to develop testing scripts which simulate user interactions with the GUI application and verify the correct behavior, state and control flow in the GUI to discover possible deviations from the expected behavior. The

tests are structured against the GUI application to elucidate and clarify what is required from user perspective” [4].

II. CHALLENGES WITH GUI TEST AUTOMATION IN ANDROID

Some of the factors responsible for these challenges are:

(1)Diversity:-The diversity of Android-based handsets in terms of screen size, OEM, operators. Etc.

(2)Fragmentation:

-Multiple devices in the market with different versions of Android OS

-Testing applications on different OS versions is a huge challenge.

(3)Testing on multiple devices:

-Due to large number of devices available in the market, it is difficult to test app on every single device and Simulators are not reliable.

(4)Time to market:

-Time to market is reduced greatly

-Very short and rapid release cycles.

(5)Newer Versions

-The upgraded version (typically there is a new Android release every 6-7 months)

Overcoming these challenges is an opportunity for software testing to reach to the next level of testing, i.e. fully automated testing. Android has provided frameworks and tools for automated testing, and these tools can be broadly categorized as:

- Instrumentation-based test framework
- GUI-based testing tools
- Third party cloud-based testing services.

III. VARIOUS AUTOMATED GUI TESTING TECHNIQUES FOR ANDROID MOBILE APPLICATIONS

1. A GUI Crawling-Based Technique for Android Mobile Application Testing.[5]
2. Automated GUI Testing on the Android Platform. [4]
3. Automating GUI testing for Android applications. [3]

4. Experiences of System-Level Model-Based GUI Testing of an Android Application [6]
 5. GUI Testing using computer vision.[7]
 6. Using GUI Ripping for Automated Testing of Android Applications.[8]
 7. Testing Android Apps through Symbolic Execution. [9]
 8. Automated Concolic Testing of Smartphone Apps. [10]
 9. Combining Model-based and Combinatorial Testing for Effective Test Case Generation [11]
 10. A Grey-box Approach for Automated GUI-Model Generation of Mobile Applications.[12]
 11. Keyword-Driven Testing Framework for Android Applications.[13]
 12. Guided GUI Testing of Android Apps with Minimal Restart and Approximate Learning. [14]
- [3] Hu, C., Neamtiu, I.: Automating GUI testing for android applications. In: Proc. 6th International Workshop on Automation of Software Test. AST '11 (2011) 77-83
 - [4] Kropp, M. & Morales, P. 2010. Automated GUI Testing on the Android Platform. Proceedings of the 22nd IFIP International Conference on Testing Software and Systems: Short Papers, Montreal: CRIM, pp. 67-72
 - [5] D. Amalfitano, A. Fasolino, and P. Tramontana. A GUI crawling-based technique for android mobile application testing. In IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops (ICSTW), pages 252-261. IEEE, 2011
 - [6] Takala, T., Katara, M., Harty, J.: Experiences of system-level model-based GUI testing of an android application. In: Proc. 2011 Fourth IEEE International Conference on Software Testing, Verification and Validation. ICST '11 (2011) 377-386
 - [7] Chang, TH., Yeh, T. & Miller RC. 2010. GUI testing using computer vision. CHI '10 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1535-1544.
 - [8] Amal_tano, D., Fasolino, A.R., Tramontana, P., De Carmine, S., Memon, A.M.:Using GUI ripping for automated testing of android applications. In: Proc. 27th IEEE/ACM International Conference on Automated Software Engineering. ASE 2012 (2012) 258-261
 - [9] Mirzaei, N., Malek, S., P_as_areanu, C.S., Esfahani, N., Mahmood, R.: Testing android apps through symbolic execution. In: SIGSOFT Softw. Eng. Notes. Vol-ume 37. (November 2012) 1-5
 - [10] Anand, S., Naik, M., Harrold, M.J., Yang, H.: Automated concolic testing of smartphone apps. In: Proc. ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering. FSE '12 (2012) 59:1-59:1
 - [11] Nguyen, C.D., Marchetto, A., Tonella, P.: Combining model-based and combi-natorial testing for effective test case generation. In: Proc. 2012 International Symposium Software Testing and Analysis. ISSTA 2012 (2012) 100-110
 - [12] W. Yang, M. Prasad, and T. Xie. A grey-box approach for automated GUI-model generation of mobile applications. In FASE, 2013 Proceedings of the 16th international conference on Fundamental Approaches to Software Engineering Pages 250-265
 - [13] Zhongqian, Wu, Liu Shu, Li Jinzhe, and Liao Zengzeng. "Keyword-Driven Testing Framework For Android Applications." Proceedings of the 2ndInternational Conference on Computer Science and Electronics Engineering(ICCSEE 2013)
 - [14] Choi, Wontae, George Necula, and Koushik Sen. "Guided GUI testing of android apps with minimal restart and approximate learning." Proceedings of the 2013 ACM SIGPLAN international conference on Object oriented programming systems languages & applicatio ns. ACM, 2013

A review of the Automated GUI testing Techniques for Android Mobile Applications are shown in table 1 based on certain parameters.

IV. CONCLUSIONS

The Exhaustive study of various GUI based testing techniques was done on various parameters which revealed their characteristics, field of application and scope of application. Furthermore, critical limitations of the testing method were identified. So the study is helpful in selecting an appropriate methodology for requisite testing needs.

REFERENCES

- [1] <http://thenextweb.com/mobile/2013/10/31/strategy-analytics-android-smartphone-shipments-81-3-q3-2013-ios-13-4-windows-phone-4-1/>
- [2] <http://techcrunch.com/2013/08/07/android-nears-80-market-share-in-global-smartphone-shipments-as-ios-and-blackberry-share-slides-per-ic/>

TABLE 1
EXISTING AUTOMATED GUI TESTING TECHNIQUES FOR ANDROID MOBILE APPLICATIONS

No	Method Used to automate testing?	Testing Tool Used?	Tests supported by tool?	Can testers Write their own scripts?	Weakness of the method?
1.	GUI Crawler	Automated android Testing tool (Robitium test framework)	Crash testing, Regression Testing	Does not support flexibility for testers to write their own tests	Lack of scripting available to testers, Limited usefulness of tool in many testing activities
2.	Android GUI Testing on the Android Platform	Android Instrumentation Framework and Positron Framework	Unit testing, functional testing	Testers are able to write their own test cases	Positron Framework would not be used for performance testing due to its slow execution times.
3.	Automating GUI testing for Android Applications	Monkey	Unit testing, functional testing	Yes	Additional bugs are to yet to reveal.
4.	Experiences of System-Level Model-Based GUI Testing of an Android Application	Monkey and TEMA tools	Regression testing	N/A	Need to make the test modelling easier.
5.	GUI Testing using Computer Vision	Sikuli Test	Regression testing, Unit testing, Test-driven development	Yes	Method is designed to test the GUI's visual feedback and not checking the internal functionality. Can't be used for any time-critical testing. Limited usability because Sikuli test is limited to emulator environments.
6.	GUI Ripper	Robotium and Android Instrumentation class	Stress testing, system testing, Acceptance testing, functional testing	N/A	Not a re-usable model for use in future testing ,only for stress testing
7.	Testing Android Apps Through Symbolic Execution	Extended Symbolic Pathfinder (SPF) tool, Java Pathfinder (JPF)	Automated generation of test cases	Yes	Integration of SPF tool suite with cloud testing infrastructure to detect both functional and security vulnerabilities.
8.	Automated Concolic Testing of Smartphone Apps	Android Instrumentation framework.	Unit testing, performance testing, functional testing	Yes	Works only for touch screen and does not handle input values problem effectively.
9.	Combining Model-based and Combinatorial Testing for Effective Test Case Generation	M[agi]C http://selab.fbk.eu/magic .	Functional testing	Yes	The improvement required in current approach is to investigate the automated test cases which were otherwise not automatically produced in these

					executable test cases.
10.	A Grey –box Approach	ORBIT and Robotium	Functional testing, system testing, Acceptance testing	Yes	Needs controlling the order of event sequences in crawling algorithm used.
11.	A Keyword driven approach	Robotium and Android Instrumentation Testing Framework	Functional testing, Unit testing, Acceptance testing	Yes	Requires more planning and longer initial time investment than going directly to the test creation stage
12.	Guided GUI Testing of Android Apps with Minimal Restart and Approximate Learning	SwiftHand	Regression testing, Functional testing	Yes	First , the current implementation does not support Apps whose main entry routine is native. Second , the current implementation works correctly only with devices with Android 4.1 or higher versions. Third , SwiftHand cannot handle apps that use internet Connectivity to store data on a remote server.