

Review of Mobile Applications Testing with Automated Techniques

Anureet Kaur

Asst Prof, Guru Nanak Dev University, Amritsar, Punjab

Abstract: As the mobile applications and mobile consumers are rising swiftly, it is concern to researchers and testing professionals to devise viable testing techniques to guarantee unwavering quality of these mobile applications. A proper mobile quality charter would serve developers as a parameter for mobile quality confirmation. Due to smaller development life cycle of mobile applications, the developed apps tend to be faulty as little effort is put in ensuring quality of the app. Thus rigorous testing is required for ensuring its quality and that too in little time. Traditional life cycle models which are used for developing desktop/laptop application, same models are adjusted to work on mobile application development. But because of certain characteristics of mobile applications which are diverse from traditional applications, same models cannot be assimilated for mobile application development. For testing also which is a considered as a phase in mobile application development need to be addressed separately. There is various mobile application testing techniques used such as manual and automated techniques. But much importance is given to automated approach due to its various advantages. This paper discusses the characteristics specific to mobile apps along with comparison of manual and automated testing and comparison of various automated testing techniques precisely used for mobile applications.

Keywords: Software engineering, testing, Mobile applications, testing techniques, automated testing.

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 importance of application quality. According to Gartner, by 2017, over 268 billion downloads of mobile apps will generate cumulative revenue of \$77 billion [1].

Figure 1 shows the growth chart. From the chart it can be seen that mobile app downloads are growing yearly. So the testing of mobile application need to focus on functional testing, security testing, performance testing, usability testing, regression testing and compatibility testing. 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. The techniques used for testing desktop/laptop software are adapted in testing mobile apps too.

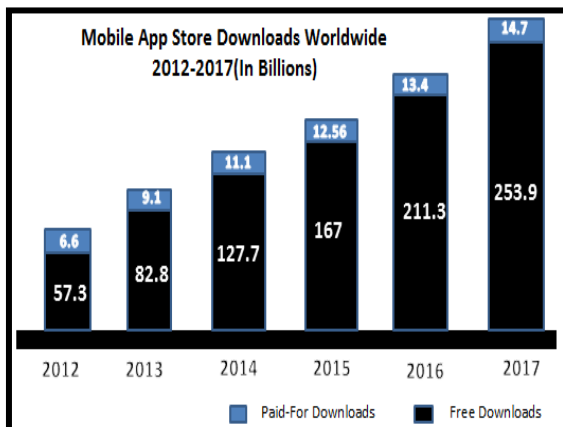


Figure 1: Mobile Apps downloads year –by- year [1]

But there are certain constraints that needs to be addressed specifically for mobile app development and testing, such as Performance Factor, Power Factor, Band Factor, Connectivity Factor, Context factor, Graphic Interface Factor, Input Interface Factor, etc. The objective of this paper is to answer fundamental questions about mobile app testing for new and experienced application testers alike. This paper will mainly focus on characteristics of mobile apps and how they further impact the testing technique selected for testing which will ultimately affect the quality of the application.

II. CHARACTERISTICS SPECIFIC TO MOBILE APPLICATIONS

The characteristics considered for developing all software applications which are called productivity factors such as Functionality requirements, Reliability requirements, Usability requirements, Efficiency requirements, Maintainability requirements and Portability requirements are common. But there are certain characteristics specific for developing mobile applications in addition to all other productivity factors. Laudson and Gibeon (2014) [2] have done a systematic review to identify characteristics that are inherent to systems and mobile. The 13 types of characteristics are observed by them.

The description of each characteristic identified is given below:-

- 1) Limited energy: Every mobile device is powered by battery and it has a certain lifetime period ,so the applications must be programmed to require the minimal amount of hardware resources possible, since the more resources consumed, the greater amount of energy expended.

- 2) Graphical Interface: Due to the reduced screen size, the interface design is limited.
- 3) Input Interface: How the user will interact with the application, in other words, if the user will interact via keypad, stylus, touch screen or voice and image recognition. The latter makes the task of developing applications that offers all these interaction options more complex.
- 4) Bandwidth: Given an application that requires the maximum, the minimum or a reasonable bandwidth, one must consider its enormous variation. A mobile application might have the maximum band at times and the minimum in other moments. Some types of applications need to realize this and act differently in each situation.
- 5) Change of context: This characteristic take into account not only the data entries explicitly provided by users, but also the implicit entries concerning the physical and computational context of the users and the environments that surround them.
- 6) Limited Performance: Due to its size and technological advancement all mobile devices, even the most advanced in its class, have limitations of specific resources such as processing power, memory and connectivity. Because of this, the performance is limited.
- 7) Reduced memory: Due to limited hardware size, less memory is available.
- 8) Connectivity: Mobile applications can be developed to support different types of connectivity such as 3G, Bluetooth, infrared, Wi-Fi, Wireless, NFC and others. In addition, a single application can support multiple types of connectivity simultaneously.
- 9) Constant Interruption of Activities: Some applications are developed to work offline and are synchronized when becomes online. Mobile applications should be prepared for different scenarios which are interrupted constantly. Receiving a call, lack of connection and low battery are examples of such interruptions, which makes the applications become much more complex.
- 10) Portability: It can be divided into two characteristics: Hardware Portability, a mobile application should be able to run on the largest number of devices possible. Software Portability, a Mobile application should be able to perform same way on all types of operating systems.
- 11) Response time: The applications must be initialized and finalized immediately or in other words, any development should be focused in the time variable. It requires the applications to be developed with a possible resource optimization for a better efficiency and response time.
- 12) Processing: Response Time is directly related to the power of Processing.
- 13) Native vs. Web Mobile: It must be defined if the application will be designed to be installed on the device itself, which is known as native applications, or used on the web.

III. TESTING LIFE CYCLE OF MOBILE APPLICATION

Software Test Life cycle process consists of various activities that help in smooth testing of the software. Various phases in testing cycle are shown in figure 2.



Figure 2: Testing Life Cycle of Mobile App [3]

IV. CHALLENGES IN TESTING MOBILE APPLICATION

The mobile apps testing have different set of problems such as:

- 1) Fragmentation: Multiple devices in the market with different versions of OS.
- 2) Testing on multiple devices: As there is large number of mobile devices, it is not feasible to get new device every time and simulators are not reliable.
- 3) Time to market: Time to market is reduced greatly .Very short and rapid release cycles.
- 4) Newer Versions: The upgraded version (e.g. typically there is a new Android release every 6-7 months)
- 5) Multitasking: Because of the small screen size, it is hard to show many applications at same time. With multiple applications open and running in background consumes a lot of battery.
- 6) Form Factor: There are different form of Mobile devices such as phones and tablets. Developing the apps for different forms is related but designing apps for them is very different.
- 7) Diversity: The diversity of handsets (e.g. Android-based) in terms of screen size, OEM, operators. Etc.
- 8) Emulator Vs. real device: Testing on emulators is cost effective as testing can be done quickly and efficiently. While testing on real devices helps to understand the application activities in real-life setups.
- 9) Manual Vs. Automated: Automated testing is cost effective, fast and more but it requires a large amount of initial cost.

V. SCOPE OF MOBILE APPLICATION TESTING

Before developing a testing strategy, decision has to be made on scope of testing. Various testing levels include:-

- 1) Unit Testing: It is performed to check app code structures to find bugs and errors.
- 2) Integration Testing: It is performed to verify that the interfaces between components works as expected.
- 3) System Testing: It is performed to ensure that the entire system of the mobile application meets all the specification specified by the application.
- 4) Regression testing: It is performed to ensure that the basic functionality is not affected due to any changes made and new features works fine as expected.
- 5) Compatibility Testing: It is performed to assure that application works as intended with the selected device, operating system, screen size, display and internal hardware.
- 6) Performance Testing & Stress Testing: It is performed to load and stress test the mobile application and database servers.
- 7) UI Testing: It is performed to ensure that the end user's experience is efficient, effective and satisfactory for user application.

For performing testing on any given level it is best to use some combination of real device and emulator.

The figure 3 shows the combination.

Testing Scope	Manual Testing		Automated Testing
	Using Real Device	Using Emulator	
Unit Testing	No	Yes	No
Integration Testing	No	Yes	No
System Testing	Yes	No	No
Regression Testing	Yes	No	Yes
Compatibility Testing	Yes	No	Yes
Performance Testing	Yes	No	Yes
GUI Testing	Yes	No	No

Figure 3: Combining Manual Testing and Automated Testing for various testing Scopes

VI. AUTOMATED TESTING OF MOBILE APPLICATIONS

As the development of applications goes through a very short cycle, it is a requirement that testing life cycle should also be fast. But with manual testing acquiring speed in processing is difficult. Various automation techniques are adopted for testing. Testing Automation allows improving efficiency and scope of application for better updates. Various advantages of automated testing:

- 1) Time Saving
- 2) Defects are identified
- 3) Higher quality software
- 4) Accuracy is high
- 5) Test cycles are also fast
- 6) Lower cost even though initial cost is high but they are compensated with long term use.

VII. LITERATURE OF VARIOUS EXISTING MOBILE APPLICATION TESTING TECHNIQUES

1) Keyword-Driven Testing
Zhongqian, Wu, Liu Shu, Li Jinzhe, and Liao Zengzeng (2013) [5] proposes an Android-based keyword-driven automated testing framework. This technique is based on Robotium and it is combined with keyword-driven testing framework which separates testing logic, testing scripts, and testing data in design. A keyword library is designed which is reusable. In this library test data and business logic is integrated. By using this library stores in a database table, the testing can be simplified and thus minimize the manual operations.

2) Test Driven Mobile Applications Development
Kim, HaengKon (2013) [6] presents a technique in which while developing the app, automated testing is run in parallel with development. For working of this technique object oriented and function oriented approaches are used together along with Model Driven Testing ideas.

3) Compatibility Testing Service for Mobile Applications
Zhang, Tao, et al. (2015) [7] presents an approach for compatibility testing of mobile apps keeping in view various mobile devices and platforms (OS) they run. The authors propose a testing strategy by using statistical approach in order to cut test costs and improve efficiency. An optimized compatibility test sequence for mobile apps is prepared using the K-Means statistical algorithm.

4) A Strategy to Perform Coverage Testing of Mobile Applications
Delamaro, et al., (2006) [8] presents a technique which support coverage testing for mobile app. This technique allows app be tested on emulators and also on real target mobile devices with help of structural coverage assessment. The authors presented this technique using a tool JaBUTi/ME.

5) Novel Approach of Automation Testing On Mobile Devices
Nagowah and Sowamber (2012) [9] presents a mobile test automation technique in which connection to a computer is not required. In this technique testing is done on the phone itself as some testing cannot be simulated in emulator/simulator. The authors have shown the results of this technique using a tool MobTAF.

6) Performance Testing of Mobile Applications at the Unit Test Level

Kim, et al., (2009) [10] presents a technique for performance testing of mobile applications using a resultant database of unit testing run on emulator. The authors have implemented this technique in a tool which verifies the reliability of the technique.

7) A GUI Crawling-based technique for Android Mobile Application Testing
Amalfitano, et al., (2011) [11] presents a technique that automatically execute GUI tests on android applications. In this technique the main aim is to find runtime crashes or user-visible faults on modified versions of the application. The authors have implemented this technique using a tool for testing a real small size Android application which shows the effectiveness and usability of the proposed testing approach.

8) Testing Conformance of Life Cycle Dependent Properties of Mobile Applications

Franke et al., (2012) [12] presents a technique for testing the conformance of Mobile apps development according to the specific properties of mobile applications. For this, the authors state a unit testing based approaches which make use of assertions in life cycle callback-methods.

Table1 shows the comparison of various automated testing techniques for mobile applications.

No.	Automated Testing Technique	Tool Used	Test Coverage	Platform
1.	Keyword-Driven Testing Framework For Android Applications	Robotium and Android Instrumentation Testing framework	Functional Testing, Unit Testing, Acceptance Testing	Android
2.	Test Driven Mobile Applications Development	Qtronic tool, DOORS tool	Unit Testing, Requirement specification testing	Just a model(No implementation shown)
3.	Compatibility Testing Service for Mobile Applications	N/A	Functional, behavioural, Regression testing	Android
4.	A Strategy to Perform Coverage Testing of Mobile Applications	JaBUTi/MW	Structural testing, Coverage testing	Java apps on any platform
5.	Novel Approach Of Automation Testing On Mobile Devices	QTP(Quick Test Professional)and test complete	Test execution on real device testing	Symbian operating system
6.	Performance Testing of Mobile Applications at the Unit Test Level	PJUnit	Performance testing, Unit testing	N/A
7.	A GUI Crawling-based technique for Android Mobile Application Testing	Automated android testing tool(Robotium test framework)	GUI testing. Regression testing, crash testing	Android
8.	Testing Conformance of Life Cycle Dependent Properties of Mobile Applications	Android developer tool along Logcat tool	Unit Testing	Android

Table 1 : Comparison of various Automated testing techniques for mobile applications

V. CONCLUSION

In this review paper comparison between mobile devices applications to other applications that run on desktop/laptop is done .Through this paper, it is emphasized to plan a testing strategy which depends on specific mobile. The available testing techniques must adjust to accommodate new characteristics of mobile apps. Manual testing for mobile apps is time consuming and due to short development life cycle of mobile apps it is difficult to use manual techniques. Various automated testing techniques has replaced manual testing .A comparison between these techniques is reviewed which helps the mobile app tester in choosing the appropriate method keeping in view the tool support, platform on which they want to work on and test coverage supported by the technique.

REFERENCES

- [1] [http://www.businessinsider.in/GARTNER-Mobile Apps-Will-Have-Generated-77-Billion-In-Revenue-By-2017/articleshow/29293386.cms](http://www.businessinsider.in/GARTNER-Mobile-Apps-Will-Have-Generated-77-Billion-In-Revenue-By-2017/articleshow/29293386.cms)(last accessed on 13/04/15)
- [2] De Souza, Silva L., and de Aquino G.S., (2014), "Mobile Application Development: How to Estimate the Effort?" , Computational Science and Its Applications–ICCSA 2014.Springer International Publishing, pp.63-72.
- [3] "Software testing Fundamentals", <http://softwaretestingfundamentals.com/software-testing-life-cycle> (last accessed on 13/04/15)
- [4] <http://www.qaac.org/wp-content/uploads/2012/07/Mobile-Testing-QAAC.pdf> (last accessed on 13/04/15)
- [5] Zhongqian, Wu, Liu Shu, Li Jinzhe, and Liao Zengzeng. "Keyword-Driven Testing Framework For Android Applications." Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)
- [6] Kim, HaengKon. "Test Driven Mobile Applications Development." Proceedings of the World Congress on Engineering and Computer Science. Vol. 2. 2013.
- [7] Zhang, Tao, et al. "Compatibility Testing Service for Mobile Applications." 2015 IEEE Symposium on Service-Oriented System Engineering, San Francisco Bay, California, USA; 03/2015
- [8] M. E. Delamaro, A. M. R. Vincenzi, J. C. Maldonado, "A Strategy to Perform Coverage Testing of Mobile Applications" 2006 ACM 1-59593-085-X/06/0005
- [9] Nagowah, L.; Sowamber, G., "A novel approach of automation testing on mobile devices," Computer & Information Science (ICCIS), 2012 International Conference on , vol.2, no., pp.924,930, 12-14 June 2012 doi: 10.1109/ICCISci.2012.6297158
- [10] H. Kim, B. Choi, W. Eric Wong." Performance Testing of Mobile Applications at the Unit Test Level". Proc. of 2009 Third IEEE International Conference on Secure Software Integration and Reliability Improvement, IEEE Comp. Soc. Press, pp. 171- 181
- [11] In IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops (ICSTW), pages 252–261. IEEE, 2011
- [12] D. Franke, S. Kowalewski, C. Weise, and N. Prakobkosol, "Testing conformance of lifecycle-dependent properties of mobile applications," in 5th International Conference on Software Testing, Verification and Validation (ICST). IEEE Computer Society, 2012, pp. 241 – 250.