A Model for Classification of Issues and Strategies Related To Smart Phones Applications Testing

Deepika Dhamija
Research Scholar
Computer Science Department
Jaipur National University, Jaipur
aghi.deepika@gmail.com

---**ABSTRACT**---

With the everyday decline in costs, and with their ever increasing demand and popularity, the makers of smart phones have really been smart in the sense that they have understood the needs of the people who love the madness of never ending games and its graphics, to remain up to date about any topic in this world by just tapping, pinning, zooming and scrolling the giant screens of their phones with the touch of a the few fingers!! Needless to mention that the smart phones are 'smart' because they run so many smart applications for almost every purpose. The existence of smart phones is solely dependent on these mobile applications because most of the other features are available in a normal feature phone as well. Now this is a reason to worry. The makers are relishing this period but with so much diversity in devices and platforms, with the constant pressure of delivering the applications in shorter time span, with the lack of testing tools for mobiles and with a variety of network plans availability, the pressure is on the applications developers to develop full proof and bug free applications. There are so many areas of concern for mobile application developers: from OS and device fragmentation to security of personal data in this e-commerce era, from connection speed to data usage, from installation to launching and finally, from handling interruptions while usage to the handling of error messages. Each of the above mentioned area is of massive importance and with the decrease in the patience level of the user, the future of your business can be in dark if these concern areas are not addressed carefully. Testing of mobile applications can't just be done in the same manner as it is done for PC's as it is more complex than traditional web applications and desktop applications. A variety of software platforms and versions, diverse hardware platforms and different network connectivity conditions are there on which mobile applications are required to be tested.

With this insight, we propose a model for classifying all these issues into formal categories & areas and accordingly, we propose appropriate testing strategies to resolve these issues.

The paper is further divided into following sections. Part I give us an introduction to the area concerned and lay emphasis on categorizing the different types of mobile applications. In part II, we identify some challenges that are posed by mobile applications and we also present some statistical data related to mobile app crashes on various operating systems and in part III, we propose our model of classifying different mobile applications testing issues and strategies to test them. Part IV includes the conclusion of the paper.

Keywords: mobile, apps, testing, software, applications

I: INTRODUCTION

Mobile apps are basically little, self-contained programs, used to enhance existing functionality, hopefully in a simple, more user-friendly way. Take one of today's modern smartphones. They all come with powerful web browsers, meaning you can do pretty much anything you can do on a desktop computer in a phone's browser. Normally, when people talk about apps they are almost always referring to programs that run on mobile devices, such as Smartphones or Tablet Computers. There are thousands of apps available falling into many different categories. Nowadays there seems to be an app for everything. Mobile apps can be categorized broadly into three different types: **Web Apps**, **Native Apps** and **Hybrid Apps**

Basis of Differentiation	Native Apps	Web Apps	Hybrid Apps
Purpose /usefulness	For use on a particular platform or device. Native apps reside on the mobile and are accessed through icons on the mobile's home screen. These are installed through an application store (such as Google Play or Apple's App Store	are not real apps but just a compressed version of a website that appropriately fits into a smartphone's screen area	are like native apps, run on the device. These are web apps built into a native mobile framework
Development Technology/ Programming Language	coded in a specific programming language, such as Objective C for iOS and Java for Android operating systems	They are run by a browser and typically written in HTML5 and Java Script.	are written with web technologies (HTML5, CSS and JavaScript)
Access to mobile phone's features	YES	NO	YES
Internet Connectivity required	Not at all time	Required at all times	Not at all time
Browser Requirement	Never	Everytime	Sometimes
Popularity	Very Much	Not Much	Very Much
App Store Presence	Yes	No	Yes
Cross Platform Development	No	NO	YES
Used across different platforms	No	YES	YES
Reliability	Most Reliable	Not Reliable	Somewhat Reliable
Method of Delivery	Downloaded onto a mobile device. Installed and runs as a standalone application (no web browser needed)	Accessed through a mobile device's web browser. No need to install new software. Updates are made to the web server without user intervention	Run inside a native container, and leverage the device's browser engine (but not the browser) to render the HTML and process the JavaScript locally. Like native apps, they live in an app store and can take advantage of the many device features available. Like web apps, they rely on HTML being rendered in a browser, with the caveat that the browser is embedded within the app.
Examples	Games like Angry Birds, using a calculator or calendar, listening to music etc.	Mobile version of any website like rediff mobile, facebook mobile, Safari Browser etc.	Yelp, Foursquare, twitter, Games like Temple Run, Subway Surfers etc which are on app store as well as use browser for updates.

Table 1: A comparison of three types of apps

All these three types of apps are used by users but while using them, they are not bothered about their type. The botheration part is for the developers and testers only because whether to build a native app, a mobile app or a hybrid app depends on many factors such as business objectives, target audience, technical requirements and so on. Companies like facebook maintain both native apps and a mobile web app. However, for others, budget and resource constraints will require them to decide if they need to build a native app or a mobile web app or a hybrid one.

II- CHALLENGES OF MOBILE APPLICATION TESTING

Testing mobile applications is more complex and time consuming compared to traditional desktop and web applications. The majority of desktop applications need to be tested on a single dominant platform – Windows. The lack of a similar dominant platform for mobile apps results in many apps being developed for and tested on Android, iOS and sometimes even more platforms.

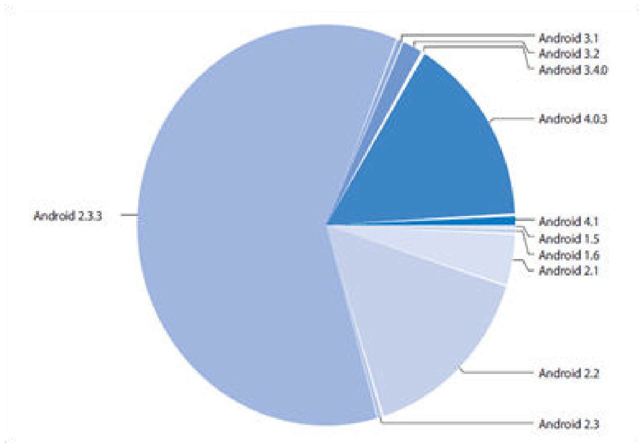

Figure 1: Android Versions

The slow pace of OS updates on Android devices (see figure above) and the resulting OS fragmentation results in the need to test apps on various versions of Android. Unlike the desktop world, where PCs are established as standardized reference hardware, the wide variety of device form factors (e.g. phones and tablets of various screen size) adds another layer of complexity in testing mobile apps.

Device diversity is an especially acute problem for Android devices – even the official Android device gallery includes over 60 devices of various screen sizes, resolutions and form factors.

The ease of upgrading apps over the air combined with increased user expectations about quicker releases (both for bug fixes and new features) result in frequent application releases. Adding multiple major and minor OS updates on top of this, test teams are continuously tasked with testing new app features or recertifying the app against a new OS version.

Mobile apps operate in a unique environment where application behavior can be affected by changes in network conditions (bandwidth change, dropped connections), alerts and notifications, as well as touch screen responsiveness. This unique environment requires additional testing to ensure acceptable app behavior in real world conditions. Also, there is a big difference between a mobile and desktop platform:

- Mobile device hardware is smaller and generally tends to have lower hardware resources than desktops/laptops.
- Smaller screens bring about different design considerations and challenges.
- Touch screen technology introduces new interaction concepts that differ from traditional input devices (keyboard and mouse).
- With a mobile device, internet connectivity is not always as reliable as a hard-wired broadband connection, which means internet connectivity is a concern and data transfer could be significantly slower.

So, all the above mentioned challenges must be kept in mind before creating mobile apps and while testing them.

III PROPOSED TESTING MODEL

There are so many parameters to test typical software which is supposed to run on a desktop/laptop, a proper sequence of testing activities has to be followed and it must be aligned with the software development life cycle. When it comes to testing a software or application that is about to run on a mobile device, the issues and their complexity increases manifold because of the dynamic nature and different types of mobile devices and their users. So many testing techniques are there like usability testing, mobility testing, component testing, functional testing, QoS testing, features testing, service testing, user interface testing, system testing and so on. An iterative mobile app testing cycle includes the three key components: Test automation which allows developers to test a function many times without having to manually test a program. Performance Testing, for ensuring the best possible user experience, and lastly the production monitoring where by monitoring the application 24/7 on real devices, your operations team can obtain metrics and detailed reports on application behavior and understand exactly what end users are experiencing. Our focus is on categorizing mobile apps broadly in two different categories and then tries to combine all the above mentioned testing techniques at one level or one step and then move on to the next step or level.

Therefore we propose two different mobile app testing models: one for the apps that are native to the mobile devices and one for the apps that are browser based, or typically called web apps.

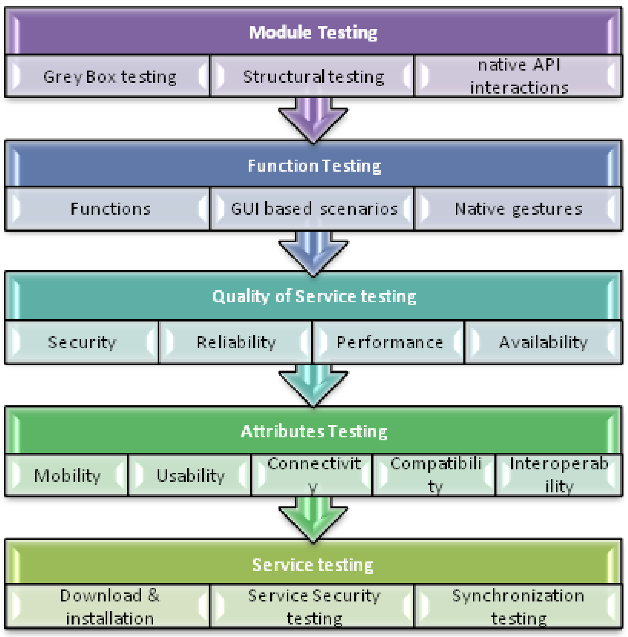

Figure 2.Testing model for Native App

The above five step model is created keeping in mind the quality of functions and behaviors' as well as the quality of service parameters .It include the following:

1. **Module Testing**: Here, we test those test objects which are separately testable as a isolated unit without integrating with other components (e.g. modules, programs, objects, classes, etc.).It also includes behavioral or black box testing, white box testing and testing of native API interactions.

a. **Grey box testing**: Mobile Application Gray Box Tests are conducted with partial knowledge about the applications and having test logins to the same. These attacks are targeted to determine flaws related mainly to three categories - Local Storage of Data, Hard-coded Sensitive Data in the Source Code and Data in Transition.

b. **Structural testing**: It is done when the internal design of the device being tested is known to the tester. It can be started at an early stage and there is no need of complete GUI to be available. It involves auditing the code base of the application for security flaws. An Android application code review is conducted on its .java files and tested via its. apk files or Android Marketplace download. An iOS (Apple OS) application code review is conducted on its .h and .m files and tested via the Apple App Store download.

c. **Native API interactions**: It is advisable to check the native API interactions like Android SDK with the mobile app being developed to ensure that it is compatible with and fulfill all the requirements of the native API.

2. **Functional testing**: It is performed to test your app through the real world experiences it needs to stand up against. For example, it tests for GUI based scenarios and native gestures like understanding how your shopping cart works under low connectivity or how your navigation app handles a cell tower hand-off. Discover what works and what doesn't on an Android tablet running Ice Cream Sandwich or the newest batch of iPhones, and so on.

3. **Quality of Service Testing**: It covers the tests related to security, reliability, performance and availability at the same time, thereby, improving the quality of service.

4. **Attributes testing**: Issues related to mobility, usability, connectivity, interoperability and compatibility are resolved at this stage.

Service Testing: At last, tests are conducted to ensure that the app exists in the store, is easy to download from the store and gets installed in considerably less amount of time. Also, tests are conducted to ensure synchronization with all other running apps and mobile devices.

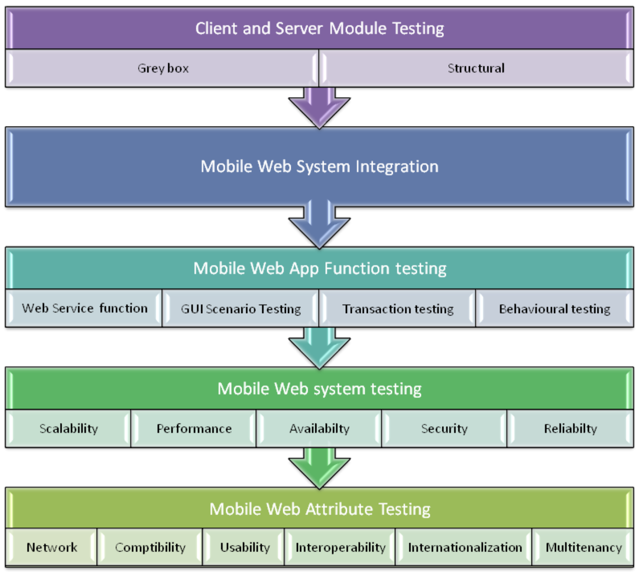

Figure 3: A Model for Mobile Web Apps

The model for mobile web apps is slightly different from the model for native apps but it aims to achieve the same goals. It assures the quality of web systems under test on networks via web browsers.

1. It is same as the native app model. Here component testing validates the quality of software components in mobile web clients.

Then, integration testing is performed to integrate the entire system which includes mobile client and server.
Functional testing is same as in native apps.
System testing is same as QoS testing
Feature testing is same as in native apps.

IV CONCLUSION AND FUTURE WORK

With so many applications from so many vendors competing for user's attention, it is important to maintain a high quality of the application as poor quality application will not only result in loss of revenue but can also affect the customer adoption and brand image. Also, the defective applications require more work on them which results in loss of productivity.

So many issues arise during mobile applications testing and it becomes very difficult for testers to categorize the issues because all these issues and problems seem to be connected and dependent on each other. There has to be some sort of differentiation among the nature of these issues so that these can be treated as a single unit of problem to be resolved.

In this paper, we have focused on creating testing models for native and web apps. However, with the advancement in technology and innovations in this area, we see some apps that are a mix of both native and web apps. We call them hybrid apps which run on your mobile as well as on the browser. So we aim to create a testing model for such hybrid apps that makes the job of testers easy.

References

[1] J. Harty. A Practical Guide to Testing Wireless Smartphone Applications, Morgan & Claypool, 2009.

[2] J.Gao, X.Bai, W. Tsai, T. Uehara, "Mobile Application Testing: A tutorial", IEEE computer society, 2014.

[3] S. Anand et al., "Automated Concolic Testing of Smartphone Apps," *Proc. ACM SIGSOFT 20th Int'l Symp. Foundations of Software Eng*. (FSE 12), 2012, pp. 1–11.

[4] R. Mizouni et al., "Performance Evaluation of Mobile Web Services," *Proc. 9th IEEE European Conf. Web Services* (ECOWS 11), 2011, pp. 184–191.

[5] I. Satoh, "Software Testing for Wireless Mobile Computing," *IEEE Wireless Comm.*, vol. 11 , no. 5, 2004, pp. 58–64.

[6] C. Hu and I. Neamtiu, "Automating GUI Testing for Android Applications," in Proc. of the 6th Int. Workshop on Automation of Software Test, ser. AST '11, 2011, pp. 77–83.

[7] "GoogleCode Android open issues," http://code. google.com/p/ android/issues/list.

[8] J. Bo, L. Xiang, and G. Xiaopeng, "MobileTest: A Tool Supporting Automatic Black Box Test for Software on Smart Mobile Devices," in Proc. of the Second Int. Workshop on Automation of Software Test, ser. AST '07, 2007, pp. 8–.13.

[9] E.J. Weyuker, F.I. Vokolos. Experience with Performance Testing of Software Systems: Issues, an Approach, and Case Study, *IEEE Trans. Software Eng*., vol. 26, No.12, Dec. 2000, pp. 1147–115

[10] M. Cundy. Testing mobile application is different from testing traditional applications. Veritest Tester's Network, Aug. 2001.