Estimating Non-Functional Requirements Using Different Mobile Operating Systems

B. Sai. Abhinay  
BTech 3rd year, Department of CSE  
Gokaraju Rangaraju Institute of Engineering and Technology, Telangana, India

T. S. K. M. Abhishek  
BTech 3rd year, Department of CSE  
Gokaraju Rangaraju Institute of Engineering and Technology, Telangana, India

K. Madhavi  
Professor, Department of CSE  
Gokaraju Rangaraju Institute of Engineering and Technology, Telangana, India

DOI: 10.23956/ijarcsse/V7I6/0210

Abstract---Mobile application testing is a process of testing applications developed for hand held devices. It is the process of checking the application for functionality, usability and performance issues. Mobile application testing is different from testing of desktop applications, as apart from regular functional and UI requirements we also have to consider factors like device hardware, screen size, platform, connectivity issues and many more. A Test Case is a set of actions executed to verify a particular feature or functionality of a mobile application. This paper addresses how an application acts in different ways in different mobile operating systems, by considering non-functional requirements such as performance, compatibility, usability, network issues, security and implementing the designed test cases on a mobile application using different mobile operating systems.

Keywords—mobile application testing, Test case design, Test case verification

I. INTRODUCTION

The number of consumer and enterprise mobile apps have grown exponentially over the last few years, leaving the end user with a humongous number of apps to choose from. App quality is the key to any application’s success and it can only be achieved through mobile application testing. This practice allows you to deliver better software and helps your app to be successful by testing its functionality, usability and consistency, growing your user base[2].

Mobile application testing is a process of testing applications developed for mobile devices. It is done by testing functional and non-functional requirements. The growing number of mobile devices is leading to a massive fragmentation of OS, screen sizes, variations of standard OS and more. With agile processes, software testing is performed every so often to assure the best possible quality[9].

New features and bug fixes need to be released with short intervals, so users don’t lose interest and new features should not bring new bugs. Testing becomes vital for an app’s survival. Since the non-functional requirements are common to many of the applications, we chose to test them as they are used to judge the operating of the system. Following are the Key Challenges for Mobile Application Testing:

1. Should be Downloadable from Particular Store
2. There are different mobile operating systems in the market. The major ones are Android, IOS, Symbian, Windows Phone, and BlackBerry (RIM). Each operating system has its own limitations. Testing a single application across multiple devices running on the same platform and every platform poses a unique challenge for testers.
3. Access to the right set of devices when there is an ever-growing list of devices and operating system versions is a constant mobile application testing challenge. Access to devices can become even more challenging if testers are spread across different locations.
4. There are over 400 mobile network operators in the world. Out of which some are CDMA, some GSM, whereas others use less common network standards like FOMA, and TD-SCDMA. Each network operator uses a different kind network infrastructure.
5. The variety of devices makes executing a test script (scripting) a key challenge. As devices differ in keystrokes, input methods, menu structure and display properties single script does not function on every device. 6. There are two main ways of testing mobile applications: testing on real devices or testing on emulators. Unfortunately, neither method is flawless. Emulators often miss issues that can only be caught by testing on real devices, but because of the multitude of different devices in the market, real devices can be expensive to purchase and time-consuming to use for testing.
7. It is necessary to test the Compatibility: Suppose an Application can work on the high resolution and it doesn’t on the fewer lower resolution.
8. It is necessary to check, While executing the app Application should be able to pick up call. 9. Mobile devices differ in screen input methods (qwerty, touch, normal) with different hardware capabilities.

II. BACKGROUND

Mobile apps have been around for a long time now. Back in the 1990s they were usually created by device manufacturers like Nokia and Motorola. These apps performed certain basic tasks. Later on, wireless service providers started making apps to differentiate the devices sold on their network to others. At the same time, third party companies started making apps for the mobile platforms like the Windows mobile OS and the Symbian OS[2].
However, there was no centralized place where end users could acquire these apps. The most modern iteration of the mobile apps started in 2007, when Apple announced the first generation of the iPhones. At the same time Apple also announced the centralized market for mobile apps called the ‘App Store’, through which, the end users had to download all their apps. Soon after in 2008, Google deployed their own platform (Android) and their own app market the ‘Android Market’ (which was later renamed as ‘Google Play’). Similar app markets were released for the mobile phone platforms developed by Microsoft, and BlackBerry as well. With these other app markets, now the mobile app developers have an even larger customer base to sell [4].

It is estimated that there are currently 2.6 Billion mobile phone users, who mostly own smart phones. With the introduction of app markets for each platform, now developers have the ability to manage the distribution of their software through one centralized market for each platform. With the increased use of smart phones and mobile apps by end users, and development of these mobile apps by software developers, mobile apps became an obvious area for software engineering researchers to examine. Non-functional requirements specifies how the system works, these are the properties that a mobile application must have. Think of these properties as the characteristics or qualities that make the application attractive, or usable, or fast, or reliable [1].

III. PROPOSED SYSTEM

Smart phones are rapidly becoming the primary method of interaction for consumers and businesses worldwide. Explosion of consumer apps can be seen in just about every industry and mobile apps are running rampant in the B2B world as well, including spaces such as Business Apps, Productivity, Collaboration. But, Failure of a mobile application depends on performing the quality of test. In present existing system, Test case designs for estimating non-functional requirements are implemented mainly on mobiles with only android operating systems. There are very limited test cases focusing on other mobile operating systems.

To design the test cases for estimating non-functional requirements to implement on different mobile operating systems. Apart from existing test cases, new test cases are designed for each non-functional requirements like performance, usability, compatibility, security etc and implemented on different mobile operating systems like Android, windows, ios, blackberry through manual and automated test.

Non-Functional Requirements:
A Non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed to be. Functional requirements are usually in the form of "system shall do <requirement>", an individual action of part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of "system shall be <requirement>", an overall property of the system as a whole or of a particular aspect and not a specific function. The system's overall properties commonly mark the difference between whether the development project has succeeded or failed. Non-functional requirements are often called "quality attributes" of a system [9][10].

a) Usability:
Usability, a non-functional requirement, testing the usability is a measure of how easily the system can be used by end users. Time required to get used to in using the application. Assessment of a user's attitude towards using the application. A usability test makes sure that the interface of an AUT is built in a way that fits the user’s expectations with respect to meeting requirements (effectively) easily (efficiently) in a simplistic satisfying manner. The primary focus is on [10]:
1. Ease of use
2. Ease of Learning or familiarizing with the system
3. Satisfaction of the user with the entire experience

b) Compatibility:
Testing the Compatibility, a non-functional requirement, is a non-functional testing technique conducted on the application to evaluate the application's compatibility within different environments. It is performed to ensure that since mobile devices have different size, resolution, screen and version. So, the application should be tested across all the devices to ensure that the application works as desired. Mobile compatibility testing refers to validating whether the under-test mobile app can work well on different mobile devices with various platforms, appliances and features, and in different environments. Higher costs of mobile compatibility testing - Due to thousands kinds of mobile devices, different mobile platforms, and various native APIs, Engineers have to spend a lot of time and effort on testing mobile compatibility [10].

c) Security:
Testing the Security , a non-functional requirement is a testing technique to determine if an information system protects data and maintains functionality as intended. Testing data can be proceeded as different combination of usernames and passwords and its purpose is to check that only the authorized people are able to access the application. Security is set of measures to protect an application against unforeseen actions that cause it to stop functioning or being exploited. Unforeseen actions can be either intentional or unintentional. Security Testing is a variant of Software Testing which ensures, that system and applications in an organization, are free from any loopholes that may cause a big loss. The goal of security testing is to identify the threats in the system and measure its potential vulnerabilities [10].

© www.ijarcsse.com, All Rights Reserved

Page | 469
Network Issues:

Network variability refers to the changing behaviour of a wireless network. The variability is caused by the following factors:

- **Signal loss**: Dead spots on a wireless network lead to a partial or total loss of network signal. This leads to abrupt snapping of the client server communication.
- **Data loss over network**: A data loss occurs when data packets sent by client/server over a wireless network fail to reach the intended destination because of network congestion. Even though data can be retrieved with a mobile operating system, the data loss has a severe impact on the user’s experience, when the application doesn’t handle the resulting error conditions properly.
- **Bandwidth**: The bandwidth can fluctuate significantly when multiple users are sharing the same network.
- **Network delay and jitter**: The time taken by each packet from a source to the intended destination is defined as network delay. The network delay experienced by each packet can vary and is known as jitter.
- **Impact of network variability on mobile applications**: The variability of a network can have the following impacts on mobile applications:
  - **Unpredictable application behavior**: The mobile application might freeze, abrupt application exit, inconsistent application behavior or irrelevant error messages might occur due to the inability of the mobile application to handle the variation in the network[10].

Performance:

Testing the performance, a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage. Performance testing is mainly useful for the following[11]:

1. To ensure that its response time doesn’t lower under peak load conditions
2. To ensure that the application performs in the same way in different versions.
3. To ensure that the application will work efficiently at different bandwidths.

This paper is mainly focused on manual testing. A Mobile application (Naukri-job search application) is chosen, considering each non-functional requirements like Usability, compatibility, security, network issues. Test cases were designed and verified manually on this mobile application using different mobile operating systems like Android 6.0.2, IOS 9, Windows 8.1, Blackberry 10.3.1. For example test case - To validate whether User interface of application is compatible with the mobile. Verifying this test case in mobiles with different operating systems whether the application is compatible with the mobile screen. Performance is tested using automation tool. Neoload is the tool used for testing performance. Results are shown in the result section[IV] in the form of a table 1.

### IV. RESULTS

The following Table 1 shows the behaviour of a mobile application using different test cases for different non-functional requirements on mobile operating systems such as Android 6.0.2, IOS 10.2, Windows 8.1, Blackberry 10.3.1

<table>
<thead>
<tr>
<th>NON-FUNCTIONAL REQUIREMENTS</th>
<th>TEST CASE ID</th>
<th>TEST CASES</th>
<th>WINDOWS</th>
<th>IOS</th>
<th>ANDROID</th>
<th>BLACKBERRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT_001</td>
<td>To validate whether User interface of application is compatible with the mobile</td>
<td>✓  ✓  ✓  ✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT_002</td>
<td>To ensure that text is readable for all users of that application</td>
<td>✓  ✓  ✓  ✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT_003A</td>
<td>To ensure that the call functionality is enabled whenever the application is running</td>
<td>✓  ✓  ✓  ✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compatibility</td>
<td>CT_003B</td>
<td>To ensure that the alarm functionality is enabled whenever the application is running</td>
<td>✓  ✓  ✓  ✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_001</td>
<td>To ensure that the buttons should have the required size and suitable for end user</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_002</td>
<td>To ensure that the buttons are placed in the same section of the screen to avoid confusion to the end users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_003</td>
<td>To ensure that the icons are natural and consistent with the application</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_004</td>
<td>To ensure that the buttons which have the same function should also have the same colour</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_005</td>
<td>To ensure that the validation for the tapping zoom in and zoom out facilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_006</td>
<td>To ensure that the keyboard input can be minimized in an appropriate manner</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_007</td>
<td>To ensure that the application provides a method for going back or undoing an action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_008</td>
<td>To ensure that the contextual menus are not overloaded</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_009</td>
<td>To validate the display of splash screen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_010</td>
<td>To ensure that text is kept simple and clear to the user for visibility</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_011</td>
<td>To ensure that the short sentences and paragraphs are readable to the end users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_012</td>
<td>To validate unmapped keys</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_013</td>
<td>To validate that user is able to exit application using slide exit option</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT_014</td>
<td>To validate that application is not hampered when other applications are running in background</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>UT_015</td>
<td>To ensure that the end user is provided with a user manual which helps the end user to understand and operate the application who may be not familiar with the application's proceedings</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>NT_001</td>
<td>To validate application performance when network is changed to wifi from 2g/3g or vice versa</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>NT_002</td>
<td>To validate network performance while moving around with the device</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Network issues</td>
<td>NT_003</td>
<td>To validate response of application when there is a signal loss</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NT_004</td>
<td>To validate response of application at different bandwidths</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>NT_005</td>
<td>To generate proper error message when there is no internet connection</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>PT_001</td>
<td>To validate the response time of the application under different bandwidths</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PT_002</td>
<td>To validate the response time of the application under different user load conditions</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td>ST_001</td>
<td>To Verify that whether the app provides users with access to a remote service, an acceptable form of authentication such as username/password authentication is performed</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ST_002</td>
<td>Verify that the remote endpoint terminates the existing session when the user logs out</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>ST_003</td>
<td>Verify that a password policy exists and is enforced at the remote endpoint</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
V. CONCLUSION

In conclusion, we believe that due to popularity of mobile apps, and the impact that research can have on developers from both small and large organizations, combined with the abundance of publicly available data, software engineering research for mobile apps is a great place for young researchers to start. Designing the right test strategy, choosing the right mobile simulators, devices and testing tools can make sure that we have 100% test coverage and help us include security, usability, performance and compatibility based tests into our test suites. Non-functional requirements plays a prominent role in enhancement of an application. The scope of the mobile testing will be bright until mobiles are going to be used in real life. Increased popularity of smart phone has increased the use of mobile application or increased popularity of applications have increased the use of smart phone. Whatever reason is, but one thing is very clear that Mobile applications are as popular as mobiles. Due to this hard hitting competition among mobile application developers lead them to develop an application which serves a different purpose of the users and make their daily task easy. But developers overlook a complete software development life cycle and skip or allot less time for testing an application. Further requirements and improvements can easily be done. Improvements can be appended by changing the existing module or adding new modules. As Security and performance are the two main aspects in application testing and we found that there are few automation tools on security testing that are commercial tools. So, developing free source automation tools might help testers of mobile application to achieve their requirements.

REFERENCES


[5] Problems and solutions in mobile application testing by Triin Samuel, Product-Focused Software Process Improvement


