

Analysis of Optimization Requirement of Mobile Application Testing Procedure

Manish Kumar, Kapil Kant Kamal, Bharat Varyani, Meghana Kale

Centre for Development of Advanced Computing

Mumbai, India

email: kmanish@cdac.in, kapil@cdac.in, bharatv@cdac.in, meghanak@cdac.in

Abstract— The rapidly changing and demanding mobile application ecosystem has resulted in explosion of mobile applications among the users across the world. This behavior of the mobile application ecosystem inspires the mobile industries to make high availability of mobile application and development over the different platforms like Android, iPhone, Windows, BlackBerry, etc. The rapidly changing and demanding mobile application ecosystem has made mobile application development more complex and critical. A mobile application should be responsive, stable and secure. This high expectation with mobile application requires the right approach of testing. There are various mobile application testing strategies available, and in the current scenario of mobile application testing, we normally apply all the available testing strategies for all the mobile applications even which are not necessarily required. To optimize this bulky mobile application testing, we have to first analyze the mobile application and decide which testing strategies are required. In this paper, we have described various mobile application testing strategies available and have presented a classification of mobile applications based upon which mobile application testing procedure can be optimized.

Keywords — *Mobile Application Testing; Optimization; Graphic User Interface (GUI); SQLite.*

I. INTRODUCTION

The rapidly changing and demanding mobile application development ecosystem can be understood by the fact that the number of mobile application download has increased by around 3000% in 2014 with approximately 138 billion downloads as compared to 4.5 billion downloads in 2010 [10]. The growing mobile application ecosystem has brought revolution in the mobile application development and it has made mobile application developers / providers to focus on creating mobile application testing strategies and road maps before releasing the mobile application for their users. It is important to build an application with all features and functionality required by the customer and which is beneficial to the application user, but it is even more critical to have a rigorous mobile testing plan before the mobile application is deployed or made available to its customer [1]. A comprehensive testing plan gives the confidence that the application will function as intended on different devices with varying screen sizes, resolutions, internal hardware, operating systems, and across any data transfer network. Today, there is a wide variety of mobile devices with different mobile operating systems, firmware updates and other possible customization that creates impossibly large sets of testing permutation and combination [8]. There is a large number of testing strategies available that makes mobile application testing more complex and a bulky procedure. Due to this complex process, extra use of resources increases the cost and

time of complete testing. For optimization of mobile application testing, the proposed classification of mobile applications can help to broadly characterize the mobile applications first and then, to choose appropriate testing strategy.

The below mentioned goals can be achieved by using this classification to optimize the mobile application testing:

1. *Reduce the completion time of mobile application testing life cycle.*
2. *Reduce the testing cost and overall application cost.*
3. *Faster release of mobile application in market.*
4. *Reduce the process involved in mobile application testing with quality assured.*

In this paper, we describe how classification of mobile applications helps in optimizing the mobile application testing procedure. In Section II, we describe some of the available testing strategies in a mobile application testing life cycle. In Section III, we propose classification of mobile applications to optimize the mobile application testing procedure. In Section IV, we describe how the proposed classification helps in optimizing the mobile application testing procedure. In Section V, we present our conclusion.

II. AVAILABLE TESTING STRATEGIES

Once a mobile application is developed, it undergoes various testings before it is released. There is a large number of testing strategies available today. Some of the mobile application testings that are performed before a mobile application is released, are described below.

A. Installation/Uninstallation Testing

Installation is one of the important strategies of the testing activity. It is performed to verify if the software has been installed with all the necessary components and the application in the all targeted devices is working as expected. Installation would be the first user interaction with the end users so it should be perfect. To verify if installation testing is successful or not, we perform the following two-step check: 1. if application can be installed by following normal installation procedure; 2. if application is seen in installed applications list. To verify if uninstallation testing is successful or not, we check if all the components of the application are removed as expected during the uninstalling process [2].

B. User Interface Testing

User interface testing is required to examine how easy or user friendly the application is, to use for the real users. It is essential that a user interface is interactive and more relevant to task of mobile application. The user interface testing is

performed on different devices with different form factor to check if the screen (including text, images, etc) is displayed as intended. This test also includes text visibility in the selected language, navigation between screens and verification of functionality online / offline, feedback from interaction with the system, i.e., downloaded application should be prompt with message [3].

C. Functionality Testing:

The objective of functionality testing is to measure the quality of the functional components of the system. The functional testing verifies that the system behaves correctly from the user / business perspective and functions according to the requirements, models, storyboards, or any other design paradigm used to specify the application. The functional test determines if each component or business event: performs in accordance to the specifications, responds correctly to all conditions that may be presented by incoming events / data, moves data correctly from one business event to the next (including data stores), and that business events are initiated in the order required to meet the business objectives of the system [2].

D. Compatibility Testing

Compatibility test involves validating the application with different Mobile platforms, with different mobile devices, screen sizes, hardware and resolutions as per requirements. The compatibility test determines that the mobile application works exactly as we want it to, across all supported devices, platforms, screen sizes, OS versions.

E. Performance Testing

Performance testing, a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing can be applied to understand the scalability of mobile applications. Performance testing includes the response time of application, application behaviour in change of different available networks, battery consumption and memory leaks, etc.

F. Security Testing

It includes the encryption/decryption techniques used for sensitive data communication, checks for multi users support without interfering with the data between them check for access to file saved in the app by any unintended users detect areas in tested application so that they do not receives any malicious content [3]. Also, the screen lock technique is such as Face Unlock, Voice Unlock, Pattern Unlock, Pin Unlock, Password Unlock, etc. also used to ensure security of mobile device.

G. Network Testing

It includes the testing of network availability, effect of speed of network on application and at not availability of network. We can check network testing in the scenario such as Signal loss, data loss over network, bandwidth, network delay, etc [5]. Unpredictable application behaviour, user-interface related errors, database corruption and functional issues are

some of the impact on mobile application due to Network variability.

H. Service Testing

The Service Testing process is responsible for planning and coordinating tests to ensure that the specifications for the service design are met and validated through delivery of the service and, including co-operation with the Release and Deployment process, to manage and limit risks that could result from insufficient utility and warranty of the service in operation [6].

III. PROPOSED CLASSIFICATION OF MOBILE APPLICATION FOR OPTIMIZED TESTING STRATIGY

Systematic software engineering techniques should be employed to maximize the probability of finding faults with minimal resources, i.e., time and money. In the mobile application testing lifecycle as shown in Figure 1, the first step requirement/design analysis is very crucial and it needs to be done carefully. It is important that all the aspects of the mobile application are captured in the first step so that the further steps can be optimized.

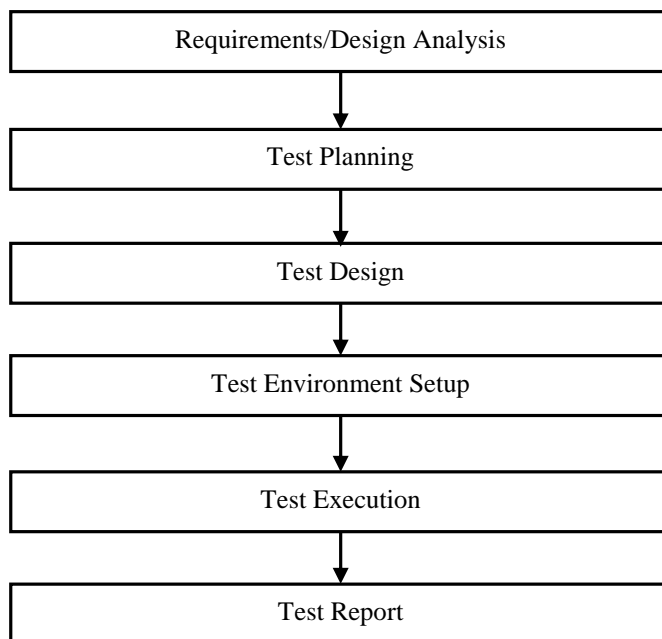


Figure 1. Mobile application testing Lifecycle

It is important to understand that mobile applications are different from traditional desktop applications because of the mobility, limited resources, context awareness, etc. [9]. Though the testing cycle may appear to be same for mobile application as compared to traditional application, the testing procedure for mobile application can be optimized if mobile application can be classified based upon various aspects. The mobile application runs on small screen size, limited resources (such as power, computational capacity, screen size, etc) as compared to a desktop. The mobile devices run on battery whereas desktops have continuous power supply. So for the optimization of mobile application testing, here we have presented mobile application classification and have analyze mobile applications on various aspects.

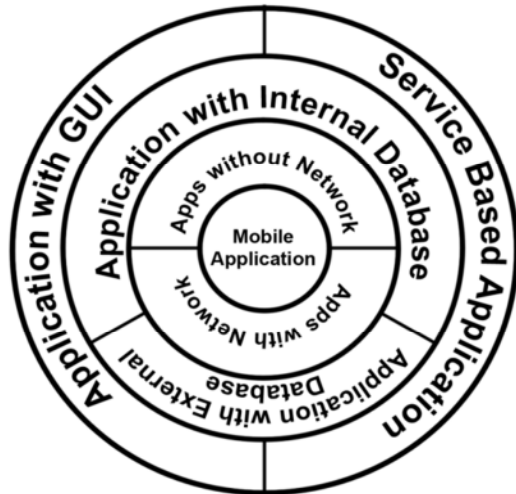


Figure 2. Proposed Classification of Mobile application

Figure 2 shows the proposed classifications of mobile application in perspective to optimize the mobile application testing.

A. Mobile application With GUI-With Internal Database-No Network[GIDNN]

It includes the mobile applications with Graphical User Interface (GUI) with internal database (database within device for example in Android SQLite) but no network requirement. This type of application does not require data network. For example, tic-tac game, which has the ability to store last scores and save the previous state of the game played by user in the past. Here, it is not required to test network and test the security of data communication channel because the data resides within the mobile device itself.

B. Mobile application With GUI-With Internal Database-With Network[GIDN]

It includes the mobile applications with GUI with internal database and also requires data network to submit the data to a remote server to get some required response. For example the application available on Mobile Seva AppStore named Ministry directory [4], initialized with its own internal database but when clicked on update it quickly repopulates its internal database with response data received from the update server. Another example is m-Indicator app on Google play store which stores the data in device database and also if we need exact location of express, rail alerts, etc. and then, it requires network connectivity.

C. Mobile application With GUI-With External Database-With Network[GEDN]

It includes mobile applications with GUI with external database and also requires data network to submit the data to remote server to store in external database. Mobile applications such as compliant registration or feedback for specific department need to store the data at department's end, so this type of application is GEDN. The mobile application of state government for birth registration, available on Mobile Seva AppStore [5], is another example of mobile application classified under GEDN, where the data filled by user in the

GUI forms are sent over internet to state centralized server and the data is stored in centralized database.

D. Service based Mobile application With Internal Database-No Network[SIDNN]

These types of applications may or may not have any GUI. They start in background and run as background process. For example, in most of the smart phones there are various sensors which capture various data based upon various parameters. There are various internal applications which keep monitoring these sensors and send s information to other application which require information about these sensors. Internal database is used to store the data of past running.

E. Service based Mobile application With Internal Database-With Network[SIDN]

These types of mobile application may or may not have any GUI, they start in background, and the captured data is stored in internal database and may also communicate with remote servers. Example is Map applications, in this the latitude-longitude information is stored locally and location details are obtained from external GIS providers and the output is shown on map.

F. Service based Mobile application With External Database-With Network[SEDN]

These types of mobile applications do not have GUI, they capture data, send it to remote servers and the data is stored in the database of the remote server. Example of such application is version checking of mobile applications. If a new version or update is found at the remote application server, then the new version or the update is downloaded from the remote server and the mobile application is updated.

G. Service based Mobile application-With GUI-With Internal Database-With NoNetwork[SGIDNN]

These types of mobile applications have GUI to start or stop the application, and then they run as background service processes. For example the Contact application, which shows all contacts, recently dialed, missed call, received call, etc., which stores the contact info in the internal database, also shows time, duration and also there is no need of network in this type of applications.

H. Service based Mobile application-With GUI-With Internal Database-With Network[SGIDN]

These types of mobile applications have GUI to start or stop the service; this type of application, store data in internal database but require data network to communicate with the remote server or application. Example location based services; the maps application with Global Positioning System (GPS) will start your route mapping service then update it with its current locations and required data on map itself and also stores it in the internal database.

I. Service based Mobile application-With GUI-With External Database-With Network[SGEDN]

These types of mobile applications have GUI to start or stop the service; these type of applications do not store data internally and send the data collected to remote server. The data is stored in the external database. Example of this type of

mobile application is Drop Box application in which it has GUI from which we can see the data in our drop box and also we store new data on external database, i.e., on cloud directly through network connection.

IV. DISCUSSION

The paper presents a classification of mobile application based upon various aspects such as graphical user interface, data network requirement, database requirement, etc.. The mobile application testing can be optimized by using this classification and all the testing strategies may not be required for all the mobile application. Table 1, as shown below, describes various mobile application testing strategies which are required for different categories of mobile applications.

TABLE I. OPTIMIZATION TABLE FOR MOBILE APPLICATION TESTING USING PROPOSED CLASSIFICATION

TESTING STRATEGY \ CLASSIFICATION	A	B	C	D	E	F	G	H
GIDNN	✓	✓	✓	✓	✓	x	x	x
GIDN	✓	✓	✓	✓	✓	✓	✓	x
GEDN	✓	✓	✓	✓	✓	✓	✓	x
SIDNN	x	✓	✓	✓	✓	x	x	✓
SIDN	x	✓	✓	✓	✓	✓	✓	✓
SEDN	x	✓	✓	✓	✓	✓	✓	✓
SGIDNN	✓	✓	✓	✓	✓	✓	x	✓
SGIDN	✓	✓	✓	✓	✓	✓	✓	✓
SGEDN	✓	✓	✓	✓	✓	✓	✓	✓

As shown in Table 1, it is not required to apply each and every testing approach for all the mobile applications but only those testing strategies which are required should be applied. For example, a mobile application classified under GIDNN category does not require service testing, security testing and network testing; so, these testings can be skipped from the complete cycle of testing process. This will reduce the testing time and cost. This in turn will reduce the overall cost of the mobile application and the reduced testing time helps in faster release of mobile application. There are some other aspects such as energy consumption, memory requirement, context awareness, etc., based upon which, the classification can be enhanced further.

V. CONCLUSION & FUTURE SCOPE

The paper presents a combined study of various mobile application testing strategies available and proposed classification of mobile application based upon various aspects such as graphical user interface, data network requirement, database requirement, etc.. The classification is simple yet the mobile application testing can be optimized by using this

classification. This helps in bringing down the overall mobile application development cost and faster release of the mobile application. The proposed classification is not concrete and the classification can be further enhanced on other parameters. Testing strategies can also be defined specific to each classification to optimize the mobile application testing procedure further. The performance and reliability of the mobile application greatly depends upon the mobile device resources. We have not included the memory and energy requirements of mobile application which are crucial for performance and reliability testing.

ACKNOWLEDGEMENT

We are thankful to all the members of Mobile Seva team of C-DAC, Mumbai and Dr. Zia Saquib (Executive Director, C-DAC), Department of Electronics and Information Technology, Govt. of India for their direct as well as indirect contribution for this paper. We also thank the anonymous reviewers for their valuable insights and comments.

REFERENCES

- [1] "Mobile Application Testing: Step by Step Approach", Retrieved from <http://www.rapidvaluesolutions.com/mobile-application-testing-step-by-step-approach/>, accessed on 21 Sept 2015
- [2] "Install / Uninstall Testing", Retrieved from http://www.tutorialspoint.com/software_testing_dictionary/pdf/install_uninstall_testing.pdf, accessed on 21 Sept 2015
- [3] B. Kirubakaran and Dr. V. Karthikeyani, "Mobile application testing — Challenges and solution approach through automation", 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME), Feb. 2013, pp. 79-84
- [4] M. Kumar and M. Chauhan, "Best practices in Mobile application Testing", White Paper, Infosys
- [5] "Mobile Seva Application store", <https://apps.mgov.gov.in>, accessed on 21 Sept 2015
- [6] M. Jandial, A. Somasundara and Karthikeyan, "Enhance Mobile Application Performance with Successful Network Impact Testing", White Paper, Infosys
- [7] M. Ryder, "Service Validation and Testing: A CA Service Management Process Map, Mar. 2010", Retrieved from http://acolyst.com/wp-content/uploads/2010/12/itil-service-validation-testing-tb-v2_234995.pdf, on 21 Sept 2015
- [8] R. R. Nimbalkar, "Mobile Application Testing and Challenges", International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064, Vol. 2 Issue 7, July 2013, pp. 56-58
- [9] H. Muccini, A. D. Francesco and P. Esposito, "Software Testing of Mobile Applications: Challenges and Future Research Directions", 7th International Workshop on Automation of Software Test (AST), June 2012, pp. 29-35
- [10] "Number of mobile app downloads worldwide from 2009 to 2017", retrieved from <http://www.statista.com/statistics/266488/forecast-of-mobile-app-downloads/>, on 30 Sept 2015