ORIGINAL RESEARCH PAPER

Computer Science

MOBILE APP TESTING TECHNIQUES, ENVIRONMENT AND CHALLENGES

KEY WORDS: Mobile, Testing, Mobility, Bluetooth

Uma Sharma

Asstt. Professor, Deptt. of Computer Science, Govt. College Hansi (Hisar) Haryana

ABSTRACT

In this paper I, investigated new research directions and ideas on mobile applications testing process, challenges and their mobility. When mobile applications testing are becoming so extraordinarily adopted in the field of IT and androids era, it is still unclear if they deserve any specific testing approach for their verification and validation. The term mobile testing refers to various types of testing techniques which is adopting. We use mobile applications testing to refer to "testing activities for native and Web applications on mobile devices using well-defined software test methods and tools to ensure quality in functions, behaviors, performance, and quality of service, as well as features, such as mobility, usability, interoperation ability, connectivity, security, and privacy.

I. INTRODUCTION

Mobile is a moving device and mobile applications refers to the software which is being used in Smart phones, which is also known as Smart mobile terminals, are high-end mobile phones that are built on mobile operating systems and offer advanced computing and connectivity. Modern smart phones have stronger processors, growing memories, high resolution touch-screens, richer sensors, GPS, high-speed data access through wi-fi. Mobile applications, also known as mobile apps in shorts, are software applications that are developed to run on smart phones and mobile devices.

The basic Comparison between desktop and web applications, mobile applications have to deal with specific challenges. Mobile applications have to process inputs from users as well as inputs from constantly changing contexts. In additionally, smart phones and mobile devices are still limited in their resources compared to modern personal computers and laptops using networks. Further, there is a large diversity of mobile operating systems, and the same operating system gets upgraded regularly and in relatively short time periods.

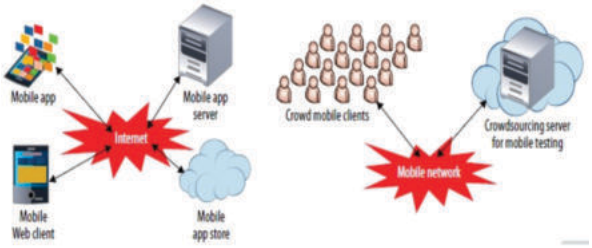

Figure: Mobile Networks

The exponential growth of this market and the criticality of the mobile app developed impose an increased attention to dependability aspects of applications which is being running on them. As demonstrated in some studies on mobile applications are not bug free and new software required to be test in technological aspect. Those applications are mobile applications different from traditional ones, so to require different and specialized new testing techniques.

2. Objective And Testing Environment

The main objectives is functionality and behavior including device-specific functions such as gesture interaction, QoS requirements, usability, better security features, and privacy. Web app testing aims to validate the quality of mobile Web apps using different Web browsers on diverse mobile devices. Web apps usually provide users with a thin mobile client to access online functions from the back-end server. Thus, in addition to functionality and behavior, QoS requirements, usability, security, and privacy, mobile Web app testing focuses on connectivity and interoperability.

The testing environment underlying mobile platform or operating system constitutes the testing environment. To achieve effective automation, testing solutions should be compatible, deployable, and executable on multiple platforms. In mobile apps, the underlying Web browser is the testing environment may become better one.

3. Major Implications On Testing

3.1. What A Mobile Application Is?

A mobile application is vaguely defined as an application running on mobile devices and/or taking in input contextual information. In order to improve our degree of knowledge on the topic, I decided to analyze the role of mobility and context-awareness on mobile applications.

In mobile computing an application is considered to be mobile if it runs on an electronic device that may move. An application is aware of the computing environment in which it runs, and adapts/reacts according to its computing, user, physical, or time context and traditional applications rewritten to run on mobile devices. A direct impact on testing, Apps for Mobile applications inherit the peculiarities of mobile applications i.e mobility, autonomy, and connectivity.

3.2. Mobile Connectivity And Autonomy

Peculiarities: mobile connectivity is one of the most peculiar, as well as critical, characteristics of a mobile application. Mobile applications typically connect to mobile networks, which may vary in speed, security, and reliability.

Implications on Testing: the application reliability, performance, security, and correct functioning strongly rely on the available connectivity type. Functional and extra functional testing has to be performed in different connectivity scenarios and networks. Bugs related to unreliable Wi-Fi, 3G, or Bluetooth connections have been reported.

The mobile differently from desktop or laptop computers, mobile devices can never rely on an electrical outlet for functioning. Different mobile applications may require very different energy consumption. Considering that a 200 hours stand-by autonomy of an iPhone 4S drops to 10 hours when a Wi-Fi connection is enabled, and to 7 hours when a 3G connection is active, an application requiring full time 3G connectivity strongly impacts the device autonomy. Reduced autonomy can be an issue autonomy improvement can instead be a business advantage.

3.3. New User Interfaces And Adaptation

Peculiarities: when developing a mobile application, developers have to follow some strictly defined guidelines for producing mobile applications GUIs. Mobile applications may in fact look differently depending on the mobile device screen resolution and dimension.

Peculiarities: Mobile Apps may adapt and evolve driven by contextual information. Such an evolution may happen during the application operation and without stopping the service. The application dependability may be impacted by unforeseen, run-time adaptations. Implications on Testing: approaches for testing the application correct adaptation/ s evolution have to be developed.

3.4. New Mobile Programming Languages:

The mobile applications development programming languages have been designed for supporting mobility, managing resource consumption, and handling new GUIs format. Objective-C, and uses the UI Kit framework for implementing GUIs based applications. Mobile app language that can be extended at run-time in Android platform, instead, replaces the java.awt and java.swing APIs with new libraries.

In another way, the traditional structural testing techniques on mobile app development need to be revised in order to be applied to new mobile languages. Byte code analysis tools and functional testing techniques may be used to deal with binary code which is called new Mobile Operating System.

4. Testing Techniques

4.1. Crowd Based Testing Method:

The crowd-based testing approach involves using freelance or contracted testing engineers or a community of end users such as uTest, along with a crowd-based testing infrastructure and a service management server to support diverse users. Currently, a service vendor supports primitive test management, a testing service, and bug reporting. Most mobile test operations are managed in an ad hoc way with very limited mobile test automation tools. This approach offers the benefits of in-the-wild testing without the need to invest in a lab or purchase or rent devices, but at the risk of low testing quality and an uncertain validation schedule.

4.2. Cloud Applications Testing:

The approach based on testing through the cloud is typically supported by testing vendors (such as www.nttdata. com/ global/en). In a panel presentation at the Fifth International Workshop on Software Testing in the Cloud, Raj Rao, VP of software quality at NTT DATA, reported on his company's device-based cloud testing approach. The basic idea is to build a mobile device cloud that can support testing services on a large scale. This approach addresses the significant increase in demand for mobile testing services by using a pay-as-you-go business model. It also allows different mobile users to provision their required testing environments via a rental service model. Compared with other approaches, this can be more cost-effective than device-based testing for large-scale applications, and it is much more effective for supporting diverse testing activities on mobile devices.

5. Mobility Testing

A running and moving device how to use effective mobile network and ensure security, the mobility testing on a native device usually involves testing the device's location-based functions, features, data, profiles, and API. For example, a mobile travel app's content should be delivered and presented to users based on their current location; this would include airport information, rental service offices, maps, and attraction points and related data. A device cannot accept that data, the testing engineer needs to know that. In contrast, testing mobility for mobile Web apps focuses on testing the quality of location-based system functions, data, and behaviors.

6. Connectivity Testing

In the area of digital environment the diverse mobile wireless networks support the connectivity needs of both native and Web apps. Because connectivity affects application performance and interoperability, engineers must pay attention to it during testing. For mobile native apps, this means considering online content synchronization during testing as well as download, deployment, and customization issues related to app stores and marketplaces.

7. Challenges And Directions

A systematic software engineering approach to software testing is aimed at maximizing fault detection, making results reproducible, and reducing the influence of external factors, such as random selection of test cases or the testers mood. When dealing with systematic testing, different dimensions can be considered. For the purpose of this work, we focus on a subset of what we consider the most relevant aspects in systematic software testing of mobile applications.

Specifically, we analyze challenges and potential future research directions on i) the testing process (test selection and execution), ii) the testing artifacts (structural and functional),

8. Conclusions And Future Scope

This paper basically provides an overview on what testing mobile applications is and can be in the next few years. The mobile app security and mobility issues arise when users are moving and using public networks. Mobile applications different from traditional ones, so to require different and specialized new testing techniques. What are the new challenges and research directions on testing mobile applications, the challenges seems to be many, related to the contextual and mobility nature of mobile applications. The app Performance, security, reliability, and energy are strongly affected by the variability of the environment where the mobile device moves towards. Some potential for automation have been outlined, being aware that a much deeper and mature study shall be conducted. I am contacting colleagues with expertise on specific topics discussed in this paper, for making this paper a more through repository of knowledge and a reference for future research on mobile application testing.

REFERENCES

1. Henry Muccini and Patrizio Esposito "Software Testing of Mobile Applications: Challenges and Future Research Directions" Research gate Publication in June 2012
2. Jerry Gao, Xiaoying Bai, WeiTek Tsai and Tadahiro Uehara "Mobile Application Testing: A Tutorial, Research gate Publication in Feb 2014
3. L. Berardinelli, V. Cortellessa, and A. Di Marco, "Performance modeling and analysis of context-aware mobile software systems," in Fundamental Approaches to Software Engineering, ser. LNCS, D. Rosenblum and G. Taentzer, Eds. Springer Berlin / Heidelberg.
4. 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.
5. G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research," Hanover, NH, USA, Tech. Rep., 2000.
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. M. Sama, "Context-Driven Testing and Model-Checking for Embedded Context-Aware and Adaptive Systems," in ISSTA 2008 poster, 2008.
8. D. Amalfitano et al., "Using GUI Ripping for Automated Testing of Android Applications," *Proc. 27th IEEE/ACM Int'l Conf. Automated Software Eng*. (ASE 12), 2012, pp. 258–261.