IJITKM 0973-4414 International Journal of IT & Knowledge Management (ISSN: 0973-4414)
Volume 11 • Issue 2 pp. 295-301 Jan 2018 - June 2018 www.csjournals.com

Mobile Software Testing: Strategies and Challenges

Er. Seema Rani

Assistant Professor, Computer Science and Engineering, CDLSIET, Sirsa

Abstract: The use of mobile devices has grown in our daily lives, eventually replacing the use of traditional computers for routine tasks such as web surfing, photo editing, audio and video playback, and file reading. Given the prominence of mobile devices in contemporary society, there is an increased need to focus on the dependability and interoperability of mobile applications. Thus, it has become clear that testing these apps is an essential step in the acceptance process for mobile platforms. This study looked into the basic ideas, parameters, features, and standards for assessing mobile applications in order to investigate several research directions in the field of mobile application testing. We draw attention to the parallels and discrepancies between testing native apps and mobile web apps. In addition, we highlight obstacles and new requirements in test environments and compare and evaluate different mobile testing environments and approaches. To find out how effectively a virtual device can replicate the entire client experience, we conducted a case study wherein we tested a hybrid app in an emulator vs a real-world device. Five Android mobile browsers were examined in a series of experiments, each running on an emulated device with the same specifications (CPU, RAM, etc.) as the real device. Metrics for functional/behavioral testing as well as performance were acquired for the application.

Keywords: Mobile Software Testing; Mobile Performance Testing; Hybrid Mobile Apps.

I INTRODUCTION

The previous couple years have witnessed a revolution in mobile devices. Mobile devices have evolved from simply being utilitarian tools to becoming intricate and advanced. Mobile applications have become extraordinarily popular among individuals and organizations for organizing and managing virtually every aspect of life [1][2][3]. The ubiquities, convenience, connectivity, personalize, and location-based services of mobile devices make them indispensable in our lives [4]. With over a billion smartphones sold, 6.8 billion mobile subscriptions, More than 300,000 apps have been created, with 76.9 billion downloads anticipated worldwide in 2014, mobile applications are ubiquitous and serve diverse purposes[5][6]. This prominence underscores the need to test these applications for confidentiality and security. The mobile application site continues to expand rapidly. The widespread availability, declining prices, and enhanced capabilities make mobile devices more powerful than PCs. Improved bandwidth, the handset culture, and other elements encourage the development of new operating systems and mobile applications. While testing mobile applications is expensive, time-consuming and complex, it remains imperative for ensuring consumer satisfaction. Skilled testers are vital not just for finding bugs but also evaluating quality before market release, thereby avoiding consumer frustration that could deter future use.

Testing mobile applications is crucial for ensuring usability, mobility, and security. The high price and restricted availability of physical equipment is a major challenge [7]. Many companies now rely on emulators for early development stages. The testing experience of a hybrid application on an emulator and a real device is compared in this study. Evaluating if a virtual device can replicate the entire client experience is the aim. Mobile engines exemplify hybrid applications combining web application and native application features.

Browsers leverage device hardware like GPS while benefiting from mobile web application server technology. The research paper is organized as follows: Section I Explain Introduction part. Section II gives a summary of mobile applications, Section III explains the types and scope of testing, Section IV discusses mobile application testing techniques, and Section V lists issues associated with mobile application testing; Section VI concludes the paper.

II A MOBILE APP: WHAT IS IT?

Software applications with a graphical user interface that are created and optimized to run on smartphones, tablets, and other mobile devices are known as mobile applications. They can be downloaded via USB, WiFi, or the internet from a web server, and they represent a rapidly expanding section of the global mobile market [8]. It's critical to distinguish desktop computer apps from mobile applications. Context-aware computing refers to mobile apps that are made for mobile devices and adjust to the user, computer, physical, and temporal contexts. Mobile applications therefore need specific testing methodologies and tools. Native applications, mobile web apps, and hybrid applications are the three primary categories of mobile applications. Native apps are installed and run locally on the device, depend on native APIs like the camera, and work offline without an internet connection [9]. They are built using platform-specific SDK's (Software Development Kits) and languages. Mobile web apps run in a web browser and require an internet connection, like YouTube. Hybrid apps combine elements of native and web apps, running locally but built with web technologies like HTML, CSS, and JavaScript [10].

Successful mobile apps, regardless of their intended use, have some characteristics in common. These include context awareness, quality design, and support for a variety of devices, connectivity, and the usage of new programming languages. Localization, reachability, security, and personalization are further features. Mobile devices are constantly connected to the network; apps must be tested on different networks for speed, reliability, and security. The graphical interface must be tested across devices for ease of use and access. Apps should also seamlessly install and intuitively operate. Supporting the diversity of mobile hardware, software, and Operating systems increase the difficulty of testing. Apps must be tested across combinations of these variables for compatibility and functionality. The Cap Gemini Quality Report states that methods rather than tools are becoming the main obstacles to mobile application testing [11].This means conventional testing techniques must be adapted for new mobile programming languages. Context awareness also introduces the need for context-specific testing techniques and coverage criteria. The variety of mobile app types and characteristics increases testing complexity and challenges. The next section will explore mobile testing scopes, types, and strategies in more depth.

III MOBILE TESTING SCOPES AND TYPES

- The evaluation of mobile apps on mobile devices with specific test tools and procedures is known as mobile application testing. Testing guarantees that the required features—security, usability, compatibility, and quality of service—are operating as intended [12]. Due to their increasing complexity, mobile apps must be tested for resilience and stability [13]. Due to the rapidly evolving mobile device market, regularly updated software, and lack of unified test operations due to the distinct processes in each app, mobile app testing is challenging. Many goals, including as

device compatibility, software dependability, and app functioning, are served by different mobile app testing methodologies. Among the objectives of testing mobile apps are:

- Verifying mobile features and actions in every circumstance.
- Testing mobile app scalability, reliability, availability and performance.
- Checking app compatibility across platforms for cross-platform functionality.
- Reviewing app design for easy adaptation across languages and cultures without reprogramming, and testing text visibility, navigation and offline functionality.
- Testing data privacy, encryption and security vulnerabilities in communications channels and software.

Mobile apps have distinct testing requirements and features:

- Testing anytime, anywhere since mobile apps are accessible round the clock.
- Testing across diverse mobile platforms, devices and browsers due to differing operating systems, displays, and hardware and battery life.
- Testing for optimal user experience through multiple inputs, rich features, native interfaces and hardware.
- Utilizing extensive virtualization and simulation to assess scalability and performance while cutting hardware expenses.
- Testing with different network connectivity types that mobile devices support.

IV. TESTING STRATEGIES FOR MOBILE APPLICATIONS

Testing of mobile applications can be categorized in a number of ways and is intended to satisfy quality standards. Based on the underlying client and server architecture, there are several classifications. This classification can be approached in four ways:

1) **Emulator for Devices**: 30–40% of exams employ this technique. Here, an emulator on a mobile device is used by the quality assurance team to verify mobile apps [14]. Emulators come with capabilities like the ability to use efficient scripting languages and go around the network. By loading the right profile, testers can quickly transition between different types of devices [13]. As there is no need for actual devices, this is inexpensive. Emulators facilitate testing of user interface, performance, and stress levels and are generally free. The same testing on a real mobile device might not yield the same results, though, as emulators lack device quirks. Furthermore, Emulators facilitate testing of user interface, performance, and stress levels and are generally free. The same testing on a real mobile device might not yield the same results, though, as emulators lack device quirks. In addition, the few emulators available in comparison to the large number of phones, as well as emulator problems that result in spurious faults. Emulation and simulation-based testing are comparable, but simulation-based testing mimics mobile behaviours on selected devices. Nevertheless, neither have native mobile testing features.

2)Mobile Cloud Solutions: In these cases, a web interface is used to access mobile devices. Applications are maintained, tested, and deployed remotely. Because of its pay-as-you-use strategy, which minimizes complexity and keeps tasks and data on the internet rather than on individual devices, it is also cost-effective. Among the advantages are

- Rent reduction via hourly or exchange devices.
- No cell plan needed for testing calls/texts.
- Investigating failures by recording test executions.

- Logging device to troubleshoot.
- Availability of numerous test devices.
- Parallel testing on multiple devices.

3)Real Devices with Real Networks: To verify functionality, dynamics, and quality of service, a corporate test lab purchases real mobile devices. Given that actual instruments are needed, it is pricey. To avoid manual labor, automation tools are employed. To cut expenses and save time, the quality team automates jobs. Android, Android instrumentation, QTP (Paid), and iOS testing tools are a few examples [15, 16].

V CHALLENGES OF TESTING MOBILE APPLICATIONS

Determining the challenges and difficulties with the application is necessary to improve its quality. To present a variety of challenges at different testing levels, the testing procedure and the outcomes must be trustworthy and objective. Right now, we might have to choose between test efficacy and strategy. A variety of tests could be necessary to ensure complete satisfaction because no exam is flawless or comprehensive. In Figure 1, barriers are surrounded by four dimensions.

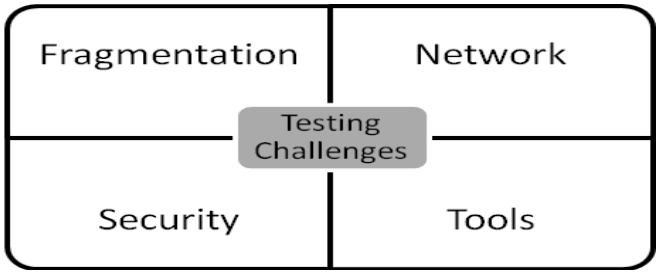

Figure 1: Testing Challenges

The first difficulty is fragmentation due to the large number of mobile devices running various operating systems, as seen in Figure 1, upper left. Compatibility issues create difficulty with different user interfaces. User experience is impacted by network performance; each operator may support a variety of technologies and unusual standards. It is costly and time-consuming to travel to each network in order to test mobile apps. By avoiding lower network levels and testing online with emulators, we can get around the network problem and save money and travel time. Bypass, however, is unable to replicate timing or network effects. Another aspect of test validity and efficacy is security; it is crucial to make sure the application is safe and does not compromise user information. The range of testing tools is the final challenge dimension. Another aspect of test validity and efficacy is security; it is crucial to make sure the application is safe and does not compromise user information. The range of testing tools is the final challenge dimension. Considering the abundance of tools accessible, it is imperative to handle the possibility of overlooking the right one.

The Cap Gemini Quality Report states that techniques, rather than tools, are now the testing hurdles. A lack of the appropriate procedure or approach affects 56% of businesses, along with a lack of easily accessible gadgets (52%), specialists (48%), in-house environments (38%), tools (37%), and time (33%). However, research reveals that in 2013, as opposed to 2012, mobile testing increased significantly, with 55% of firms adopting new techniques and instruments to verify security, functionality, and performance, up from 33% in 2012.

Testing process, Testing artifacts, and Testing kinds are the three primary categories into which some scholars divide testing issues. The problems with the mobile application testing process are outlined in Figure 2.

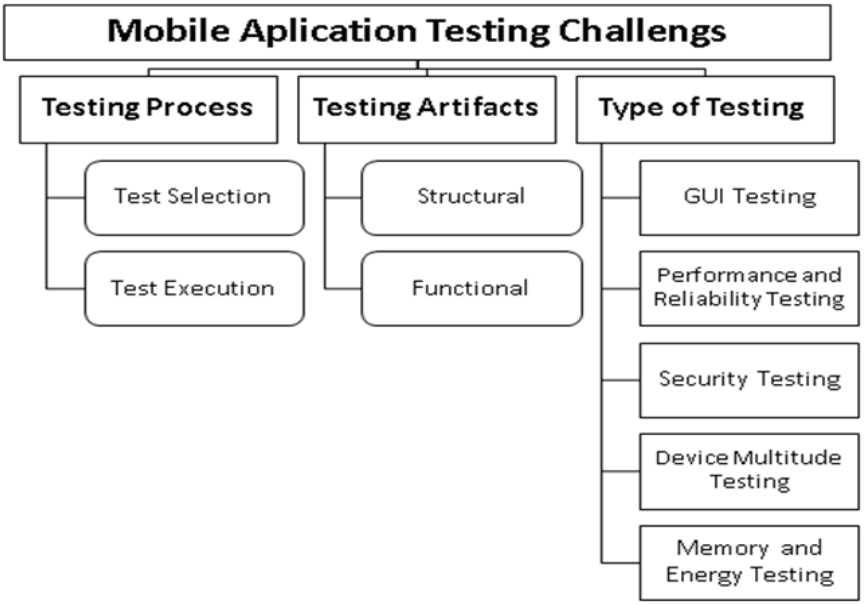

Figure 2: Mobile Application Testing Challenges

As mentioned, there are three types of mobile apps: native, web, and hybrid. Each has a different structure and function. Figure 2 divides the testing process difficulty into two categories: test execution and test selection. Given the variety of app kinds and features available, additional criteria could be needed to guarantee thorough testing coverage of elements like type, platforms, and capabilities. This is the challenge of test selection. The execution of tests addresses the wide range of unique mobile situations. It is challenging to test on every device, even slightly different ones. Introducing meaningful, trustworthy test cases with rich contextual inputs becomes difficult as a result. Current simulations are unable to replicate real-world phones with sensors, GPS, and networking. Contextual inputs might be implemented using new automated capture-replay systems. We concentrated on structural and functional methods for testing artifacts:

1) **Code-Based Structural Testing**: In Code-Based Structural Testing mobile languages include movement, sensing, and energy constructs. An efficient, low-cost method of quickly detecting errors and inconsistencies is automated crawling. Mobility, sensing, and energy structures could be better taken into account in new coverage criteria.
2) **Model-Based Functional Testing :** Model-Based Functional Testing necessitates application and environment specification. State-based methods can simulate several execution modes, such as airplane mode, meeting mode, and low battery. Techniques for testing mobile apps are popular.
3) **GUI Testing** - Testing display and data on different devices. Automated script capture and replay, even on different devices, can efficiently test GUIs.
4) **Reliability and Performance Testing**: This is dependent on context, connectivity, modes, and resources.

5) **Security Testing** - Networks have different security; viruses could access personal context. Conventional techniques should consider simulated contextual factors and transmitted data.

6) Multitude testing - Different devices have different features and OSs, making "test many devices" expensive and time consuming. This should be replaced by effective automated techniques. We can use real devices, emulators, or both. Real devices expose hardware/firmware limitations but are costly and time-consuming. Emulators are easy to manage but lack real device limitations and quirks. A prototype released for free public testing could collect runtime data and failures. Using both leverages emulation speed and cost with real device functionality verification. At a comparatively modest cost, an imitated environment improves speed and device variety. The actual devices confirm that all development goals have been reached and validate the functionality of the applications.

The mobile testing landscapes and gaps have been emphasized by researchers Sahinoglu et al. [17]. There are metrics available, along with definite publications. In order to organize and classify the research data that has been published in the field of mobile application testing techniques and challenges, Zaein, et al. [18] carried out a systematic mapping analysis. They reported that testing techniques and challenges have been organized and classified. There are gaps found and particular problems for practitioners highlighted.

VI CONCLUSION AND FUTURE WORK

Outsourcing, cloud, and crowd-based testing are examples of automated testing techniques that are gaining traction because they offer more affordable options than traditional application testing. Additionally, they facilitate layer-by-layer testing and make it evident how application-level errors differ from operating system or framework errors. Some companies may adopt testing services in an As-a-Service approach, providing specialized skills and labs to thoroughly test mobile apps affordably. Choosing the right test approach depends on various factors. Organizations may decide based on economics, quality assurance needs, or other constraints. In a growing world with daily mobile manufacturing advances in software, operating systems, and hardware, efficient mobile app testing is increasingly important. There are various mobile app testing approaches, which can confuse organizations. Different researchers propose different models and types. Each has pros and cons. To combine the benefits of different approaches, Google proposes a new modular phone concept. Users can update with ease when components are user-swappable. The primary components are all detachable, click-in modules. This makes testing easier and allows users to participate. In further research, we will examine the variety of testing approaches and tactics by employing a "hardware/software compatibility approach." The goal is to produce mobile devices that, like computers, can run many operating systems. This would resolve difficulties with device support and incompatibility between hardware and software.

REFERENCES

[1] Muccini, H., Francesco, A., and Esposito, P., “Software Testing of Mobile Applications: Challenges and Future Research Directions”. Proceedings of the 7th International Workshop on Automation of Software Test (AST), 2012.

[2] Huang, J., and Gong, Y., “Remote Mobile Test System: A Mobile Phone Cloud for Application Testing”, IEEE 4th International Conference on Cloud Computing Technology and Science.2012.

[3] Gao, J., Bai, X., Tsai, W., and Uehara, T., "Mobile Application Testing" IEEE Computer, vol. 47, no. 2, Pp. 46-55, Feb. 2014.
[4] Franke, D., Kowalewski, S., Weise, C., and Parkobkosol, N., "Testing Conformance of Life Cycle Dependent Properties of Mobile Applications", IEEE Fifth International Conference on Software Testing, Verification and Validation, Pp 241-250, 2012
[5] Information Technology for Management, 6th Ed., by Turban, Leidner, McLean, and Wetherbe, Wiley, 2008.
[6] Anne Kaikkonen, Aki Kekäläinen, Mihael Cankar, Titti Kallio, and Anu Kankainen., "Usability testing of mobile a between laboratory and field testing", Journal of Usability Studies, vol. 1, Pp 4-16. 2000
[7] Dongsong Zhang & Boonlit Adipat, "Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications", International Journal of Human-Computer Interaction, Volume 18, Issue 3, 2005.
[8] Liu, Z., Gao, X., and Long, X., "Adaptive Random Testing of Mobile Application", 2nd International Conference on Computer Engineering and Technology, Vol. 2, Pp 297-301. 2010.
[9] Kirubakaran, B., and Karthikeyani, V., "Mobile Application Testing – Challenges and Solution Approach through Automation", International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME) February 21-22,2013.
[10] She. S., Sivapalan, S., and Warren, I., Hermes,"A Tool for Testing Mobile Device Applications", Australian Software Engineering Conference,Pp 121-130, 2009.
[11] Amalfitano, D., Fasolino, A., and Tramontana, P., "A GUI Crawling based technique for Android Mobile Application Testing", 4th International Conference on Software Testing, Verification and Validation Workshops, Pp 252-261, 2011.
[12] Capgemini , Mobile Testing Ashift from tools to Methods, available: http://www.capgemini.com/resource-file-access/resource/pdf/wqr_2013 14_mobile_testing_pullout.pdf , Access on 2016.
[13] Liu, Z., Liu, B., and Gao, X., "SOA based Mobile Application Software Test Framework, Reliability" , 8th International Conference on Maintainability and Safety, 2009, pp.765-769.
[14] Nagowah, L., Sowamber, G., and Mauritius, R., "A Novel Approach of Automation Testing on Mobile Devices" International Conference on Computer & Information Science (ICCIS), Pp 924-930, 2012.
[15] Dantas, V., Marinho, F., Costa, A., and Andrade, R., "Testing Requirements for Mobile Applications", International Conference on Computer & Information Science (ICCIS), Pp 555-560, 2009.
[16] Kim, H., Choi, B., and Wong, E., "Performance Testing of Mobile Applications at the Unit Test Level",Third IEEE International Conference on Secure Software Integration and Reliability Improvement, Pp 171-180, 2009.
[17] Harleen K Flora, Xiaofang Wang, " An Investigation on the Characterstics of Mobile Application", International Journal of Technology and Computer Science,Vol 11, Pp 21-27, 2014.
[18] Sahinoglu M, Incki K, Aktas MS, " Mobile Application Verification: A Systematic Mapping Study", Computational Science and Its Applications--ICCSA, Jun 22, Pp. 147-163, Springer International Publishing.