Cloud Based Mobile Application Testing

Rohini Temkar, Shailendra Gadekar, Dhruvil Shah
Vivekanand Education Society's Institute of Technology, Chembur

***Abstract* - In today's world all applications are available on mobile and therefore there is a certain need to develop all applications on mobile as well. Since there is huge demand for mobile applications these application needs to be tested thoroughly for its correctness. Testing of mobile application is the most difficult task due to its varieties and different operating systems. Although there are simulators and emulators available but they only simulate the working of operating system and cannot test the core functionalities for the mobile device. In order to overcome above testing issues cloud testing provides major solution to the problems faced during mobile testing. Making use of cloud infrastructure in which the service provider does the software testing activities of a given mobile application in a cloud infrastructure for customers as a service based on their requirements.**

Keywords– Mobile testing, Native apps, web apps, hybrid apps, testing as a service (TAAS), cloud security.

I. INTRODUCTION

The graph of mobile application usage is increasing day by day. Mobile applications have become the part of people's daily activity and therefore due to its necessity the quality of mobile application is the most important aspect that needs to be focused on. The quality of application can be improvised by testing every aspect of the application rigorously. In order to create an impact on the end users mind the application should not only provide accuracy but also should be compatible with variety of mobile devices.

The usage of mobile applications increases day to day. It even became a part of people's daily activity which means the end users expect quality in the application. Thus testing comes into the picture but unlike traditional applications testing the mobile application differs due to drastic changes in the hardware of the mobile device in which the application runs. Also the current testing models have to be adjusted according to mobile specifications to perform testing. Further mobile application testing requires more resources than normal testing here are many unique challenges that arise while testing mobile applications, like varied device environment, hardware limitation and differences. Making sure that a mobile application will be bug free and perform in a particular platform as expected is costly and time consuming.

Cloud computing is a network-based service which provides virtual hardware simulated by software running on physical server. Cloud testing uses cloud infrastructure for testing. The highlighting fact is that in cloud based testing the user need not be aware of the infrastructure and technology being used. Using cloud testing the time taken to set up the testing environment can be reduced. The key factors in cloud testing is that its shared resources with centralized memory, bandwidth, location independent – the service can be accessed from anywhere, utilization of existing infrastructure

II. MOBILE: NATIVE APPS, WEB APPS, AND HYBRID APPS

A. *NATIVE APPS*

Native apps live on the device and are accessed through icons on the device home screen. Native apps are installed through an application store (such as Google Play or Apple's App Store). They are developed specifically for one platform, and can take full advantage of all the device features — they can use the camera, the GPS, the accelerometer, the compass, the list of contacts, and so on. They can also incorporate gestures (either standard operating-system gestures or new, app-defined gestures). And native apps can use the device's notification system and can work offline.

B. MOBILE WEB APPS

Web apps are not real applications; they are really **websites** that, in many ways, *look and feel* like native applications, but are not *implemented* as such. They are run by a browser and typically written in HTML5. Users first access them as they would access any web page: they navigate to a special URL and then have the option of "installing" them on their home screen by creating a bookmark to that page.

ISSN: 2278 – 7798

C. HYBRID APPS

Hybrid apps are part native apps, part web apps. (Because of that, many people incorrectly call them "web apps"). 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 [1].

	NATIVE	**HTML5**	**HYBRID**
App Features			
Graphics	Native API's	HTML, Canvas, SVG	HTML, Canvas, SVG
Performance	Fast	Q	Slow
Native Look and Feel	Native	Emulated	Emulated
Distribution	Appstore	Web	Appstore
Device Access			
Camera	YES	NO	YES
Notifications	YES	NO	YES
Calendar	YES	NO	YES
Geolocation	YES	YES	YES
Gestures			
Swipe	YES	YES	YES
Pinch, Spread	YES	NO	YES
Connectivity	Online and Offline	Mostly Online	Online and Offline
Development Skills	ObjectiveC, Java	HTML5, CSS, JavaScript	HTML5, CSS, JavaScript

Table 1: Comparison between Native, HTML5 and Hybrid apps

III. COMMON ISSUES IN MOBILE APP FACED BY USERS

A. COMMON MOBILE USABILITY ISSUES

The goal of usability testing, simply put, is to make sure that a user can complete the tasks they are expected to complete. More importantly, they must be able to do so easily and without becoming frustrated. A good user experience can be thwarted by number issues, including:

- Functionality: A mobile application must present the user with the appropriate functionality. If the functionality of an application is perceived as incomplete or inadequate, customers may be lost.
- Layout & Design: Good layout and design allow a user to easily complete tasks. If a button is placed in the perceived 'wrong' place, users will get frustrated and might look for an alternative product.
- Interaction: The flow of an application must be natural and allow the user to easily complete tasks. If a user believes they have been sent to the wrong page and have to manually backtrack in order to find the page they were looking for, they can become frustrated and abandon the application.[2]

B. COMMON MOBILE LOAD & PERFORMANCE ISSUES

Load testing is another critical step in the successful launch of a mobile application. The key factor to be aware of here is mobile carriers and data usage, which can affect the speed and ease-of-use of the application. If the application performance is poor for the end user, they will be likely to divert to other apps and services. The application must therefore be tested with different devices and carriers depending on the county and region.

C. COMMON MOBILE SECURITY ISSUES

The explosion of mobile applications presents an entirely new set of security challenges. While most of the tools and practices of traditional web and desktop applications are equally applicable to mobile, there are some unique concerns to keep in mind, including lost or stolen devices, mobile malware, targeted attacks on devices and more. No application can ever be 100% secure from threats, but as developers, these are questions that you should always be asking yourself when it comes to security testing for mobile apps:

• Confidentiality: Does your app keep your private data private?

• Integrity: Can the data from your app be trusted and verified?

• Authentication: Does your app verify you are who you say you are?

• Authorization: Does your application properly limit user privileges?

• Availability: Can an attacker take the app offline?

• Non-Repudiation: Does your app keep records of events?

ISSN: 2278 – 7798

D. COMMON MOBILE LOCALIZATION ISSUES

The worldwide proliferation of mobile applications (and the mobile web) requires your products to be accessible and contextual for users in all markets and regions. But with the localization process exceedingly difficult to verify, many companies are excluding entire sub-sets of users without even knowing it [2].

IV. TYPES OF TESTING FOR MOBILE APPLICATIONS

1. *USABILITY TESTING*

This includes text visibility in the selected language, navigation between screens, and verification of functionality online/ offline, feedback from interaction with system, i.e., downloaded application should be prompt with message.

2. *COMPATIBILITY TESTING*

This entails validating the application for different mobile devices, OS versions, screen sizes, and resolutions as per the requirements, checking if integration server changes, checking for the app isolation with other apps on the device.

3. *OPERATIONAL TESTING*

This entails checks for back-up of necessary information in the app, save and recovery plan if battery goes down, data lost in case of app upgradation from appstore market, app access if user gets any alarm, call, message, reminder, etc., and battery power usage while app is being accessed.

4. *SERVICES TESTING*

This includes checking for mobile app not to act as a server, checking if a service takes too long or is used offline, and checking if a service goes down and returns malformed responses.

5. *PERFORMANCE TESTING*

This includes checks on server connection changes to WIFI from 2G/3G or vice versa, shared images size used for the Application is as per the requirement, application response time, code optimization for the CPU cycle, battery consumption, memory leaks, resources like GPS, CAMERA, etc., freed.

6. *SECURITY TESTING*

This includes encryption/decryption techniques used for sensitive data communication, checks for multi-user support without interfering with the data between them, checks for access to files saved in the app by any unintended users, detect areas in tested application so that they do not receive any malicious content.

V. CHALLENGES IN MOBILE APPLICATION TESTING

Mobile applications though have restricted computational resources at disposal are designed to be as functional as the traditional application counterpart. That's why when it comes to mobile application testing; there are various unique factors which challenge the testing process.

A. Device Diversity Device diversity in the sense is that each and every mobile phone differs in screen size, screen orientation, free memory, chipset, architecture. The mobile application should satisfy the seven software quality models as described by Dominik Franke [3].

• Screen sizes: xxhpdi, xhdpi, hdpi, mdpi, ldpi are some of the screen resolution. The mobile application should be flexible enough to be used in all the resolutions.

• Screen orientation: There are only two types of screen orientation they are portrait and landscape. The mobile application should perform with full functionality in both orientations.

• Free memory: Latest mobile devices ranges from maximum 3GB to minimum 512MB RAM. Thus the mobile application should be capable of scaling up in devices with higher memory and scale down in devices with lower memory.

• Chipset: Qualcomm's Snapdragon, Mediatek's MT, ST-Ericsson's NovaThor, Nvidia's Tegra and Samsung's Exynos are different chipsets used by different mobile devices. So the application should be optimized for all these chipset.

• Architecture: The major architecture in use for mobile devices is ARM and Intel's x86. The application should be adaptable to both the architecture and thus testing is tedious for mobile application.

ISSN: 2278 – 7798

B. Operating System Though Android and iOS dominate the mobile OS market, former leading by quite a huge margin, one should not ignore the overcoming windows phone OS and other under development Firefox OS, Ubuntu touch and less active OS like Blackberry OS and symbian. For a mobile application to become successful it should play well in and be compatible in all mobile platforms. One can make a note of the popularity of Whatsapp mobile application; it is because it is available for almost all mobile OS platforms.

C. Network for each and every mobile device varies from GSM to LTE. Each varies in speed and the way in which the device connect to the network infrastructure. The application should use the networks effectively without interrupting the flow of data. D. Runtime Environment Since mobile devices use variety of mobile operating system their run time environment also differs. Runtime is where the mobile application gets executed. Android uses just in-time (JIT) runtime environment while iOS uses native Mach-O runtime environment.

VI. MOTIVATION

Traditional testing methodologies requires high budget to set up resources and time consuming. This is where cloud testing comes into place. The resources here refer to the mobile devices. A company can't spend huge money for a high end mobile device to use it for testing for short period of time. Testing efficiency is boosted drastically in cloud computing. Since cloud computing can use virtualized environment, performance testing can be conducted more effectively. Also the testing tools in cloud are available anytime and anywhere which makes it more reliable with high accessibility. Gao, J, Bai analyzed and proposed a research framework for large number of research challenges in testing an application. In order to overcome these challenges, a new Testing Framework which is implemented by Testing as Service (TaaS) in Clouds.

VII. PROPOSED MODEL

In this proposed model we are making use of Testing as a Service (TaaS). TaaS cloud infrastructures is considered as a new service model, in which the service provider does the software testing activities of a given mobile application in a cloud infrastructure for customers as a service based on their requirements. Our model provides the following services as a part of Testing as a Service (TaaS)

- Real Devices: Allocated to the customer based on their specification and requirement. Customers are charged based on the usage time of the device. Using real devices in cloud greatly reduces the testing cost of the application by paying only for what you use in the device unlike buying the new device only to test for a specific period of time.
- Emulator: Emulator creates a virtual mobile device environment of any platform or any manufacturer's specific brand. In this way, we can speed up the testing process. Since it's an emulator it can be set in no time and can be customized easily. Only basic testing like acceptance and functionality testing can be done in emulator since other testing methods will be effective on the real device hardware
- Automated testing tools: Various automated testing tools like HP-Quick Test, IBM's RQM, and Robotium can be installed in the cloud server. Using this service the license cost for using the tools use the above mentioned tools except Robotium as it is an open source. More such tools can be installed as per as the requirement[4].

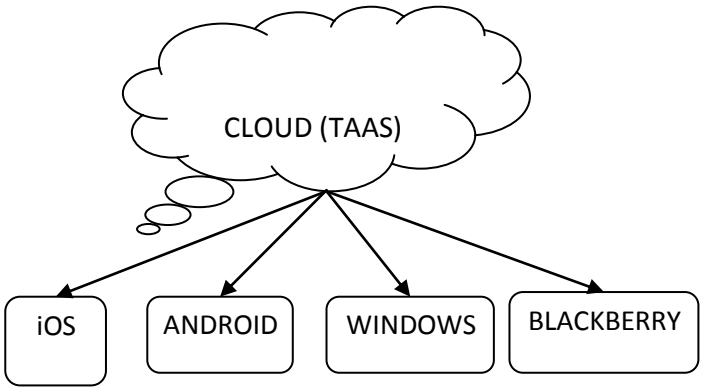

Figure 1: Testing as a Service (TAAS)

Steps involved in using TAAS:

1. Develop user scenario that need to be tested.
2. Design test cases.
3. Select the cloud service provider.
4. Select subscription based on the duration to access cloud services.
5. Setup infrastructure and the configuration required to test the result on the device.

ISSN: 2278 – 7798

6. Start testing the scenarios on the virtual devices (cloud based).
7. Monitor testing goals.
8. Deliver results and redevelop test cases if any.
9. Deployment

Above steps are graphically represented as follows:

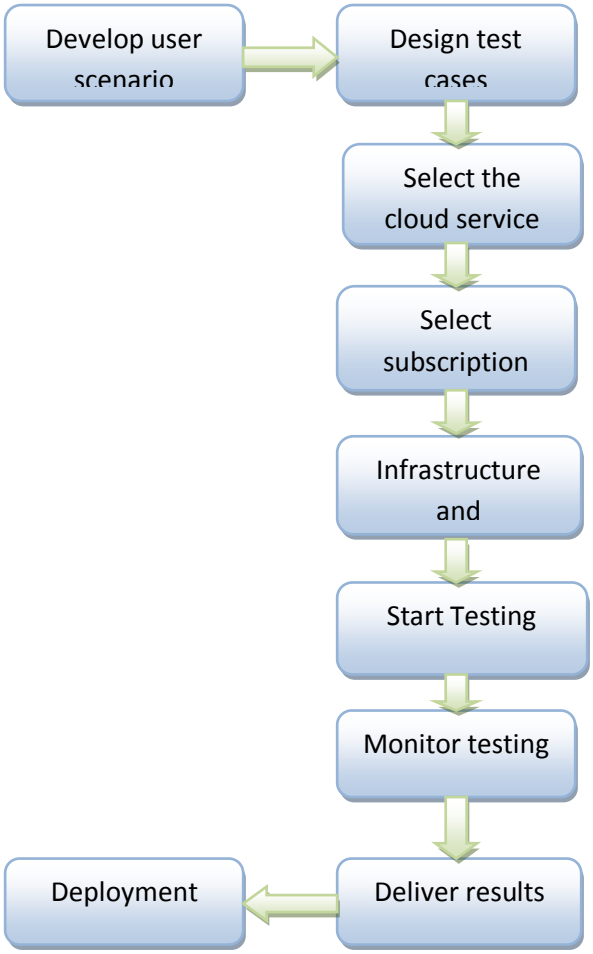

Figure 2: Steps involved in using TAAS

There are several benefits of using a cloud solution:-

- Rent per hour, swap devices
- Test incoming calls and text without the need of cell plan
- Automated test execution is recorded to video to investigate failures
- Device logs are recorded to help with troubleshooting
- Large number of devices available for testing
- Tests can be run on several devices in parallel
- Web based interface
- Build integration

So, given these market demands, why is the cloud an increasingly viable option for mobile testing? Well, because, cloud-based testing is highly scalable and asset light, as it allows on demand, pay per use models for services hired. Assurance service providers and professionals can set up their own infrastructure, or rent it from cloud testing service providers. They can test complex mobile apps that run in distributed component based environments, using various scripting languages and integrated development environments (IDEs). By doing so, they can achieve numerous benefits for their customers. Cloud based testing services are pre-standardized. Thus, they free testing teams from spending costly resources in configuring test environments. Moreover, cloud-based testing has the advantage of providing mobile device management, an otherwise complex and time consuming activity.

TaaS can be used in the following scenarios:

1. Functional testing – In a continuous integration kind of scenario, TaaS could be a platform for creating an agile based functional testing environment.
2. Load Testing – TaaS could be used for creating various kinds of loads to stress test applications. The scale-in/scale-out nature of cloud comes in handy for generating variable loads.
3. Performance and Benchmark testing – For ISV's, looking to create benchmark reports for their products with their standardized test suites, TaaS can be used.
4. Regression testing – Applications which are in maintenance mode, can make use of TaaS to run regression tests of previously written test scripts.
5. Mobile application testing – Mobile applications can be tested on TaaS. TaaS provides a realistic platform to test mobile applications, and leverages the key advantages of cloud to test application performance using cloud based content delivery networks (CDN) distributed across multiple locations around the world, effect of network latency and this will be testing on a live network

Many cloud based mobile testing service providers today offer comprehensive test capabilities – performance testing; load testing; stress testing; capacity testing; failover testing; app security testing; governance, risk and compliance testing; latency testing; and browser testing.

ISSN: 2278 – 7798

Cloudy issues:
Here are five aspects you need to consider before you choose a service provider suited to your enterprise needs:

- Capabilities
- Performance on priority
- Security
- Contracts
- 'Per Use' models and pricing
- Other allied costs

VIII. CONCLUSION

This paper presents importance of testing mobile applications then various challenges faced in testing a mobile application compared to traditional application and further more discussed the benefits of cloud computing and introduced cloud testing to reduce the mobile application testing cost, decrease the testing time to meet deadlines, cutting off license cost and for its extensibility using the service model Test as a Service (TaaS). We believe this approach with trending cloud technology can simplify the hectic process in mobile application testing.

IX. FUTURE SCOPE

Although using this approach one can test the user-level application on mobile devices with the services and resources provided using cloud computing but there are security issues in cloud like privacy, unauthorized access and leakage of sensitive information which needs to be cited and the data analysis (i.e.) collecting the test data can be cumbersome due to the shared resources in cloud environment which produces latency in request.

Protection of cloud infrastructures:

A. Identity management

Identity management schemes in cloud computing use active bundle schemes, where predicates are evaluated over encrypted data and multiparty computing. This presumes that the used encryption schemes allow the execution of predicates without violating confidentiality and privacy, which is often hard to fulfill. These techniques do not need trusted third party (TTP) for the verification or approval of user identity. As a result, the user anonymity is guaranteed and the identity is not disclosed. As an alternative to existing public key infrastructures, ID based encryption schemes may also be used in the cloud computing context. A shortcut of such identity management schemes is that active bundle may not be executed at all at the host of the requested service. It would leave the system vulnerable. The identity remains a secret and the user is not granted permission to his requests [5].

B. Software isolation

To address the security of the hypervisors, different domains are used for providers and users, each with a special trust agent. This encompasses the use of different trust strategies for service providers and customers so as to take time and transaction factors into account for trust assignment. Despite the efficiency of this approach, its scalability is questionable. Software isolation in a very large scale cross cloud environment is hard to guarantee. This scheme is able to handle only a limited number of security threats in a fairly small environment. In addition, they often have a negative impact on the system performance because of the important computational load.

IX. REFERENCES

[1]. RalucaBuidiu, "Mobile: Native Apps, Web Apps, and Hybrid Apps", http://www.nngroup.com/articles/mobile-native-apps

[2]. The Essential Guide to Mobile App Testing by utest, Ebook,http://core.ecu.edu/STRG/materials/uTest_eBook_Mobile_Testing.pdf

[3]. Dominik Franke, Stefan Kowalewski. "A Mobile Software Quality Model" 2012 12th International Conference on Quality Software.

[4]. Jerry Gao, J., Bai, X., & Tsai, W. T. (2011). Cloud testingissues, challenges, needs and practice. Software Engineering: An International Journal, 1(1), 9-23.

[5]. Q. Liu, G. Wang, J. Wu, "Efficient Sharing of Secure Cloud Storage Services," International Conference on Computer and Information Technology, GB, 2010

[6]. T. Vengattaraman, P. Dhavachelvan, "A Model of Cloud Based Application Environment for Software Testing", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 3, 2010

[7]. Mohan Kumar, Manish Chauhan, "Best Practices in Mobile Application Testing"

ISSN: 2278 – 7798