ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

Cloud Enabled Test Evaluation on Mobile Web Applications

K.Vivekanandan[1], Rajkumar Bhojan[2], Subramanian Ganesan[3]

Professor, BSMED, Bharathiar University, Coimbatore, India[1]

Test Architect, Wipro Technologies, Bangalore, India[2]

Professor, Electrical & Comp. Engineering, Oakland University, Rochester, MI, USA[3]

Abstract: As the mobile applications and mobile users are growing rapidly, it is indeed for researchers and testing experts to come up with effective verification techniques to ensure reliability of these mobile applications. An appropriate mobile quality framework would serve developers as a guideline for mobile quality assurance. In order to mitigate mundane manual testing on Mobile application, we have come up with a customized mobile test automation framework using SeeTest and Selenium TestNg flavor. In this paper we describe a cloud based evaluation on a multi lingual mobile application, sample regressing test cases on multi language applications, Framework for handling complex test cases, and an example automated test.

Keywords: Mobile Test Automation, Cloud based testing, framework, Regression Testing

I. INTRODUCTION

According to authors, Dominik Franke, et al., Quality assurance helps the developer to keep a check on the quality of his software. But quality assurance methods, like testing or validation, vary between different kinds of software, in particular between software for mobile and desktop applications. They also stressed that mobile phones have often to deal with dynamic connectivity (3G, GPRS, WLAN, varying signal strength, ...), restricted user interfaces (small display, rarely full keyboard,) and specific application lifecycles (applications are often interrupted, e.g. incoming calls). These properties of mobile devices are one major reason for the lack of quality in software for mobile devices. Mobile platforms are so diversified that are available on the mobile market viz., Android, iOS, Windows Mobile, Symbian,). Each platform has its own architecture and design concepts. Furthermore the mobile market is currently booming and competitive, which is why development cycles are kept short. These circumstances make great demands on software and ask for specific approaches and methods for quality assurance.

Mobile device has become one of the most important equipment in people's daily life, which brings us not only convenience of communication, but more and more work and entertainment applications. But more complex software comes at a cost: the quality assurance becomes more difficult. And due to the short development and life circle of the mobile software, clear and nimble test is required absolutely. [1] As several languages are used in developing software systems, architectural rules may affect artifacts of multiple languages. To support cross-language rules, we need to have a method to build a unified model from different artifacts. [2]

With the advent of Web 2.0 applications and new browsers, the cross-browser compatibility issue is becoming increasingly important. Although the problem is widely recognized among web developers, there is no systematic approach to tackle it today. None of the current tools, which provide screenshots or emulation environments, specifies any notion of cross-browser compatibility, much less check it automatically. In this paper, we pose the problem of cross-browser compatibility testing of modern web applications as a 'functional consistency' check of web application behavior across different mobile web browsers and present an automated solution for it.

Our approach consists of (1) automatically analyzing the given mobile web application under different browser environments and capturing the behavior. (2) A customized framework for handling multi-language web applications using Selenium Web Driver. We validate our approach on several open-source and industrial case studies to demonstrate its effectiveness and real-world relevance.

Selenium Web Driver: In order to accomplish test automation suite, we used Selenium Web Driver. It is a tool for automating testing web applications, and in particular to verify that they work as expected. It aims to provide a friendly API that's easy to explore and understand, which will help make our tests easier to read and maintain. It's not tied to any particular test framework, so it can be used equally well with JUnit, TestNG or from a plain "main" method. [2].

According to seleniumhq.org [4], Selenium-Web Driver was developed to better support dynamic web pages where elements of a page may change without the page itself being reloaded. Web Driver's goal is to supply a well-designed object-oriented API that provides improved support for modern advanced web-app testing problems. Selenium-Web Driver makes direct calls to the browser using each browser's native support for automation. How these direct calls are made, and the features they support depends on the browser you are using.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

Advantages of Selenium Web driver

1. It is an Open source
2. Though there is no Record and playback option, it's easy to learn and implement.
3. It supports multiple browsers.
4. Supports multiple scripting/programming languages.
5. It also supports parallel execution.

Mobile Web Application

A mobile web app is a web application formatted for smartphones and tablets, and accessed through the mobile device's web browser. Like a traditional web application, a mobile web app is built with three core technologies: HTML (defines static text and images), CSS (defines style and presentation), and JavaScript (defines interactions and animations). Since web apps are browser-based, they're intended to be platform and device independent, able to run on any web-enabled smartphone or tablet. A mobile web app is normally downloaded from a central web server each time it is run, although apps built using HTML5 (described below) can also run on the mobile device for offline use. [5]

In this project, we took a sample customer registration application which was developed in five different languages. This application can be used any of the large scale automotive industries.

In order to register in this portal, customers use specific settings, such as user name and password and also offer the convenience of applying some default settings to users. Customers can register themselves either through websites or mobile browsers.

Mobile Testing Challenges:

1. Mobile Apps are expected to receive inputs from different context providers (i.e., users, sensors and connectivity devices), inputs that vary from the different (and changing) contexts the mobile device can move towards. This may lead to the unpredictability and high variability of the inputs the application is potentially receiving.
2. Mobile application languages add some specific constructs for managing mobility, sensing, and energy consumption.
3. Mobile applications functional testing requires specifying both the application and the environment. (e.g., low battery, meeting, and flying mode)
4. As far as GUI testing is concerned, the challenges are a) to test whether different devices provide an adequate rendering of data, and b) to test whether native applications are correctly displayed on different devices.
5. Performance and reliability of mobile applications strongly depend on the mobile device resources, on the device operational mode, on the connectivity quality and variability and other contextual information.
6. Security is particularly relevant due to the mobility of the device into networks with different security levels. [6]

Our Framework

In order to meet the above said mobile testing challenges, we have come up with a customized automated testing framework as shown in Fig.1for implementing automated regression testing suites. Our customized Software Testing framework is a multi-platform, multi-language framework designed around the idea of reusable components. Some of the reusable components are Account Login(), AccountInfo(), SelectBrowser().

We used open source tool Selenium (2.0) TestNG in Eclipse IDE as a base platform for our framework. TestNG is a testing framework designed to simplify a broad range of testing needs, from unit testing (testing a class in isolation of the others) to integration testing (testing entire systems made of several classes, several packages and even several external frameworks, such as application servers).[7]

We developed a Hybrid framework for attaining goal of this project. Hybrid framework is the combination of one or more frameworks. This framework is what most frameworks evolve into overtime and multiple projects. Most of the industries use Data-driven Framework in combination of Function decomposition method in their real time projects.

Modularity-driven framework: In Modularity framework common task in test script are grouped together as Modules. Based on the application business flows, the modules will be derived. In this project, we divided modules based on the languages for our regression testing.

Data-driven framework: This framework supports data-driven testing by importing data from an external data. Test Data is read from the external files (Excel Files, Text Files, CSV Files, ODBC Sources, and ADO Objects) and are loaded into the variables inside the Test Script. Then variables are used both for Input values and Verification values in AUT (Application under Test).

In this project, we used Data Table or Excel sheet for feeding test data into the application as shown in the Fig.2. The data in the "Run" column identifies whether the test case is to be run (automated) or not.

Run Flag	Test Case	Username	Password
Y	TC1001	XXXX	YYYY
Y	TC1002	AAAA	BBBB
Y	TC1003	AAAA	CCCC
N	TC1004	BBBB	DDDD
END			

Figure 2. Data Table for Test Suite

This column has the letter "Y" which denotes that the test case should be run. If any step in the test suite is not being run, then corresponding row in the first column is "N" or be left blank. The second column of the test suite indicates the name of the Test Case that has to be

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

executed. So, if the column A is specified "Y", it means that the particular test case is going to be executed.

The Java Driver Class opens the application based on the language selection and Run flag status, fetches the data from Data Table and Sends it to the AUT (Application Under Test). Then corresponding test code will run the application based on the test steps. Once the first cycles gets completed, the controller goes to next record in the Data Table, it searches for Run Flag Status, if it is "Y", it executes the second test scripts and so on until it reaches the "END".

We also used Experitest's SeeTest tool for capturing the mobile applications objects [8]. We extracted the application elements that we wanted to run a test and set the object identification methods. Once it was extracted, the elements were showing up in the object repository (OR) and SeeTest generated the scripts. Then we exported scripts to our TestNg framework.

Exported scripts were customized as shown in Appendix-1, for adding features like fetching data from xls sheet, merging one or more executable methods and sending the reports in html/xslt results.

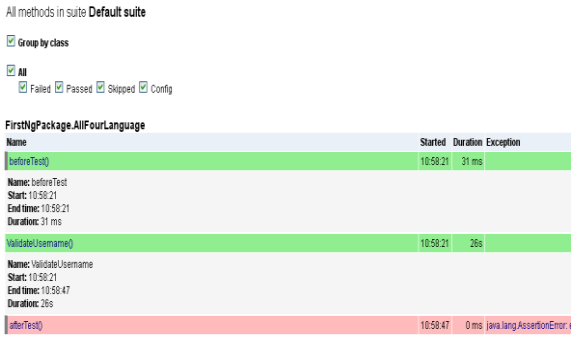

Figure 3. HTML Result

HTML/XSLT Reports: After the execution is completed, our framework produces an HTML report automatically as shown in the Fig 3. In this report, TestNg writes each and every methods used in the test and time taken for the each execution. With the help of ant, we run the build from the command prompt for the .xslt report.

Apache Ant is a Java library and command-line tool whose mission is to drive processes described in build files as targets and extension points dependent upon each other. The main known usage of Ant is the build of Java applications. Ant supplies a number of built-in tasks allowing to compile, assemble, test and run Java applications.

Here in this project, Ant is used in the context of plug-in development in the build.xml. [8] Once the build is successful, it produces an interactive XSLT sample report as shown in the Fig 4 & Fig 5. In this report, we will be able to find out how tests are passed and how tests are failed in a test suite. The failed test steps will also have links which redirects to screen shots of the errors captured during the execution.

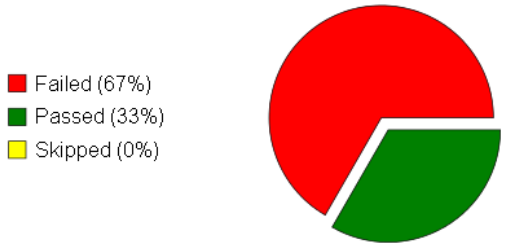

Figure 4. XSLT Result

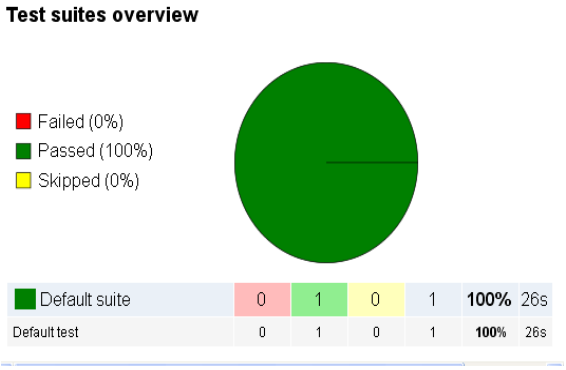

Figure 5. XSLT Result

Several times, a failing Selenium test case does not show enough information through console output. That makes very difficult to understand neither what went wrong nor what the required action to fix it. If build is successful, no screenshots will be taken. If build is unsuccessful, framework will trigger a screenshot. It will be very helpful for test engineers to monitor where and when errors occur during the execution.

S.No.	TC_ID	Test Scenario
1	MA_T001	Invalid username Or do not enter username
2	MA_T002	Login incorrect temporary password
3	MA_T003	Login with US Account in Spanish Locale set
4	MA_T004	Subscriber Logs Into m.xxx.com with Valid credentials from Email connections
5	MA_T005	Subscriber navigates to m.123.com from Email link

Figure 6. Sample Test Cases

Sample Test Cases: Some of the sample test cases are given in the above Table

Compatibility Testing using Framework: In General, Compatibility testing is a type of software testing used to ensure compatibility of the system/application/website built with various other objects such as other web browsers, hardware platforms, users (in case if it's very specific type of requirement, such as a user who speaks and can read only a particular language), operating systems etc. This type of testing helps find out how well a

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

system performs in a particular environment that includes hardware, network, operating system and other software etc. It is basically the testing of the application or the product built with the computing environment. It tests whether the application or the software product built is compatible with the hardware, operating system, database or other system software or not. [10]

We started validating the web 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. [11]

Compatibility Testing is non-functional testing in which we ensure that application/website/system is capable of running on various objects like on various mobile Browsers, on various Resolutions, on various Operating Systems, with some other application, on network.
In this paper, we also found the following conditions for Compatibility testing:

- In general when algorithm of page is very complex then cross browser testing become the need of testing as soon as we can start
- When page have a lot of animation, ActiveX , Java Applet and Dynamic content then Cross browser testing should be started as soon as it is possible
- If algorithm is running on Client side then Cross Browser testing should be start early
- Other wise Cross browser testing is started when all functional testing get to end and Acceptance testing is just on the way for the final touch [12]

As part of our mobile test automation framework, we developed number of reusable components (eg., AccountInfo()) for executing test suites. Those test suites were converted to no. of .jar files based on language selections. We also parameterize browser names for different compatibility testing using major browsers like Safari, Google Chrome and Opera. When surfing the Web on a smartphone, most of us stick with the browser that came with our handset. Mobile browsers like Opera, Rockmelt, Dolphin, and the brand new Futureful are sparring with the built-in browsers on the iPhone, iPad, and Android-running smartphones and tablets, hoping to grab a percentage of the growing market for surfing the Web on these smaller screens. [13]

As the mobile market continues to grow sharply, those brands that make real-world testing coverage a priority will enjoy ROI in terms of increased market share, profitability and above all, user loyalty. Those who neglect testing will struggle to keep up in a world filled with app stores, social media and increased user expectations. [14]

Implementation with Cloud

According to world quality report 2013-2014, Test environment can be difficult and expensive to build and maintain, are often underutilized or idle, and maintenance and downtime issues can impact on testing availability and deadlines. Running load and performance tests creates additional pressure on QA resources, particularly when testing efforts need to be scaled up only for short periods of time. But test environments need to be constructed so that they accurately represent production conditions, and kept up to date to avoid inconsistent and misleading results. Test environment management is covered later in the report in more detail.

Besides, this report further extends on Cloud infrastructure and solutions. Testers can use the inbuilt scale-up/scale-down elasticity of a cloud ecosystem to generate load on their applications or access a shared resource pool on demand. Infrastructure components and resources, such as networks, applications, databases or software licenses, can be provisioned almost instantly -reducing downtime, lowering costs through usage-based pricing and helping deliver applications to market faster.

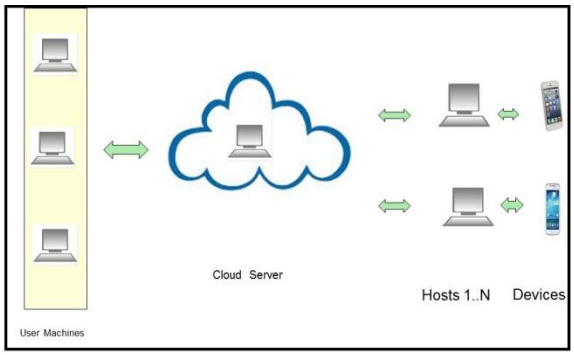

Figure 7. Cloud Framework

The above figure-7 shows our cloud based framework adopted for our implementation. Here User machines will be connected to License server and Jenkins Dashboard. When a user clicks on an execution job in Jenkins Dashboard, it invokes execution on host machine through cloud server. The host machines will have "N" number of mobile devices. Each device will have AUT (Application Under Test) build. The selenium scripts which come from our framework will be executed on the applications. Based on our execution criteria, the test scripts will be executed on real devices. If needed, we execute the script concurrently on AUT and appropriate execution results will be sent back to Jenkins dashboard.

The rapid adoption of software applications delivered on demand has created the need for QA organizations to offer specific strategies to validate the functionality, security and performance of software as a Service (SaaS) applications. SaaS vendors perform their own rigorous testing to ensure that their applications are delivered free of major problems, but individual organizations still need to perform a variety of testing activities to verify all customizations, API integration components, upgrade paths and end-to-end security of the business processes. Nearly three-quarters (71%) of respondents confirm that they have developed specific approaches for testing cloud-based services, paying special attention to performance (54%) and data security (48%). [15].

According to authors in [16] Cloud Computing is emerging as a promising distributed computing paradigm. It delivers a wide range of services like Infrastructure as a

Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The cloud resources and services are dynamically provisioned like utilities. The elastic nature of the cloud facilitates users to have as much or little of service depending on their need at any given time. [16]. In our framework, we used SeeTestCloud for testing multiple devices concurrently. In order to execute a suite level test scripts in multiple devices, we used this private cloud for parallel execution.

II. CONCLUSION

In this paper, we described our new cloud based test automation framework which performs automation regression testing of mobile applications directly on the multilingual mobile web browsers. Since we developed a more robust framework, we believe, this framework forces encapsulation by hiding technical detail for manual test team reduces the complexity and provides ease of handling test execution. Our framework can easily be extended for additional requirements for additional languages and future versions of the mobile browsers.

REFERENCES

[1] Hwang, S.M. Chae, H.C. Design & Implementation of Mobile GUI Testing Tool. International Conference on Convergence and Hybrid Information Technology, 704-707., 2008.
[2] Razieh Rahimi, Ramtin Khosravi, "Architecture Conformance Checking of Multi-Language Applications", ieeexplore.ieee.org, 2010
[3] http://code.google.com/p/selenium/
[4] http://docs.seleniumhq.org/projects/webdriver/
[5] http://testng.org/doc/documentation-main.html#introduction
[6] B. Kirubakaran, Dr. V. Karthikeyani, "Mobile Application Testing - Challenges and Solution Approach through Automation", International Conference on Pattern Recognition, Informatics and Mobile Engineering, Feb 2013
[7] www.lionbridge.com/files/2012/11/LionbridgeWP_MobileApps2.pdf
[8] http://experitest.com/
[9] http://ant.apache.org/
[10] http://istqbexamcertification.com/what-is-compatibility-testing-in-software/
[11] http://www.infosys.com/flypp/resources/Documents/mobile application-testing.pdf
[12] http://abodeqa.wordpress.com/2012/11/09/browser-compatibility-testing-when-where-and-how-it-is-done/
[13] http://www.technologyreview.com/news/510486/the-browser-wars-go-mobile/
[14] www.utest.com
[15] http://www.capgemini.com/thought-leadership/world-quality-report-2013-14
[16] R. Jeyarani, N.Nagaveni, and R.Vasanth Ram "Self Adaptive Particle Swarm Optimization for Efficient Virtual Machine Provisioning in Cloud", International Journal of Intelligent Information Technologies, 7(2), pp: 25-44, April-June 2011.

BIOGRAPHIES

Dr.K.Vivekanandan, Professor of Management and Dean of Faculty o f Social Science,Bharathiar University, India. He has been working with Bharathiar School of Management and Entrepreneur Development, for the past 25 years. He has guided 15 Ph.Ds. and Published 30 articles in reputed International Journals. He received his Masters and Ph.D (Computer Science) from Bharathiar University, Coimbatore, India. He has a long term association with National University of Rwanda for collaborative research work in Information System. vivekbsmed@gmail.com

Rajkumar J.B, is a Solution Test Architect, Wipro Technologies, Bangalore India. He has over 20 years of experience in both IT and Academics. He holds M.Sc., (Physics) from B harathidasan University, MCA., from Bharathiar University and M.Phil (Computer Science)from Manonmaniam Sundranar University, India. Currently he is pursuing Ph.D (Computer Science) from Bharathiar University, India. He has executed IT projects in diverse geographies including India, APAC & USA. He is a Certified Scrum Master and has rich experience in Agile/scrum Methodologies. He is a member in IEEE and ACM. jbrkumar@gmail.com

Dr.Subramaniam Ganesan is a Professor of Electrical and Computer Engineering, Oakland University, Rochester, MI, 48309, USA. He has over 25 years of teaching and research experience in Digital systems. He served as the chair of the CSE department from 1991 to 98. He is with Electrical and Computer Engineering Department since 2008. He received his masters and Ph.D. from Indian Institute of Science (IISc) Bangalore, India. He worked at National Aeronautical Laboratory (NAL) India, Ruhr University, Germany, Concordia University Canada, and Western Michigan University before joining Oakland University in 1986. ganesan@oakland.edu