

AN AUTOMATION FRAMEWORK DEVELOPED FOR WEB BASED APPLICATION USING SELENIUM WEBDRIVER

Ranjitha SR

Information Science and Engineering, Sri Jayachamarajendra College of Engineering, Mysuru, India.

Abstract: *Testing the last phase in the software development lifecycle is a gambling between cost time and quality. It has a major impact on the product quality that needs to be delivered to the end users. Testing can be carried out in two ways -Manual testing and Automation testing. This paper addresses the concepts of manual and automation testing and drawbacks associated with manual testing and advantages of test automation. The main objective of this paper is to focus on, effectiveness and importance of test automation. The test automation is acquiring more importance in an enterprise, as it is helping in improving the software or product quality at the expenditure of optimum time and cost. This paper also presents the case for one, developing their own automation framework for testing their enterprise products.*

Index Terms: *Manual Testing, Automation Testing, Automation Testing Framework, Test suits, Test cases, Selenium.*

I. INTRODUCTION

Testing is an activity of finding the defects in the product. It is the critical element of product quality assurance. Testing can be carried out in two ways - Manual testing and Automation Testing. Testing process will be fruitful if the defects are discovered at the early stages of the software development process. A fruitful testing has major impact on product quality, time, cost, Business value, and design schedule. Early detection of defects makes a major impact on the execution cost and schedule. In project environment where requirements get added on in an incremental fashion, chances will be high for defects to get injected in to the product. Frequent regression testing will be required to capture the defects at early stages of testing. It will cost more to do regression testing manually in projects that follow any increment models such as Agile, Scrum and others. Continuous Integration (CI) is a very effective process to bring high quality software products. Automating CI process will save maximum effort in the process of building the product and testing. Continuous Integration process can be automated with the help of an effective automation framework. An automation framework is a platform that provides a single workspace for integrating various hardware resources, software resources, tools and services. It allows the tester to design and develop an efficient test automation scripts for the reliable analysis of defects for the system under test. The automation framework also has other added advantages such as it enhances the automated tests scripts efficiency by enabling the reuse of code and avoiding the test case inter- dependency. It provides an efficient root-cause

analysis for the defects found in the system under test. Dynamic execution of tests according to the test scenarios reduces the dependency on subject matter experts.

In this paper our focus is to demonstrate how to build the product specific automation testing framework for web based applications using automation testing tool "Selenium WebDriver". The remnant of this paper is organized as follows. Section 2 explicates a brief description of related work. Then, Section 3 is a brief out of the methodology used. In Section 4, the experiments and results are captured and the paper ends with the Conclusion in the Section 5.

II. RELATED WORKS

Testing activity involves a systematic checking for software correctness. Software testing process attempts to evaluate the quality of the product and provides assurance about the quality of product to its consumers. Any product can be tested both manually as well as through automation.

A. Manual testing

Manual testing is an old rigorous method of software testing which involves human intervention. It requires a testing engineer to carry out all the test operations manually on the software which is under test without the help of Test automation [1]. Manual Testing is an activity in which a test engineer has a proper written test plan, where in the tester develops some set of test scripts which covers a set of important test cases that need to be performed on the software under test. A test case is nothing but a predefined set of conditions that will be applicable for particular software applications and the test engineer will execute all these test cases to ensure the correctness of the software applications. In order to check whether the software quality requirements are completely met, there must be two types of test cases: one is positive test cases and another is negative test cases. Manual testing is a strenuous activity which requires the test engineers to possess a certain set of qualities such as to be patient, creative, observant, innovative, and skillful [2]. Manual testing is mandatory for testing of any new application before its testing can be automated. Manual testing is necessary for checking the automation feasibility. Manual testing won't require the knowledge of any testing tool. There are many drawbacks associated with manual testing:

- Manual testing is not best suited for regression testing since it involves repetitive testing on very large set of datasets.
- Manual testing is time consuming and tedious job since it involves the test cases execution by testing

engineers.

- Manual testing is not cost effective since it involves testing engineers for the execution of test cases.
- The reliability factor will be less for manual testing than compared to automation testing because of human errors and it can become boring and hence error prone.

B. Automation testing

Automation Testing is a process which involves automating of test scripts using scripting languages such as Python, JavaScript or Perl [3]. The automated test cases enter the test data into the system under test, compare expected outcome with the actual results and generates the detailed test reports hence the automated test cases can be executed quickly and repeatedly without the human intervention. Test Automation requires considerable investments on necessary software and hardware resources. Automation testing is best suited in the Incremental developments which involves the repeated executions of same set of test cases. Automation testing is well suited for the environment where the requirements changes frequently and more cycles of regression testing needs to be performed. Various advantages of test automation:

- Automation testing is much faster than compared manual testing since the execution is carried out by the computer.
- Automation testing reduces the cost investments on Test engineers.
- Automation testing is more useful in repeated testing and it can be reused on different versions of software.
- Automation testing is more reliable in terms of accuracy.
- Test automation increase wider test coverage of application features.

III. METHODOLOGY

Selenium is a freeware portable software testing suite used for developing Test Automation framework. The test cases can be written using any of these popular programming languages such as C#, Java, Groovy, Perl, PHP, Python and Ruby and can be executed against most modern web browsers. The web browsers supported by the selenium are Google Chrome, Internet Explorer 7 and above, Firefox, Safari, Opera, HtmlUnit, phantomjs, and Android (with Selendroid or appium). Selenium can be run on operating system platforms like Windows, Linux, and Macintosh etc. Selenium Suite is composed of following components: Selenium IDE, Selenium Core, Selenium 1 (also addressed as Selenium RC or Remote Control), Selenium 2 (also addressed as Selenium Web driver), and Selenium-Grid [4]. Each tool has their specific roles and is used as regression testing tool for web and mobile applications.

A. Selenium Webdriver

Selenium Webdriver is also addressed as Selenium 2 is a freeware software testing tool used for testing web based as

well as mobile applications [4]. Webdriver in selenium was designed to overcome some of the major limitations of Selenium Remote Control (selenium RC). Selenium RC has some of the major drawbacks such as

- It requires separate selenium server to execute the automated tests.
- It supports only Java Scripting language for executing automated scripts on browsers.
- It supports only web based applications and
- It fails to support the latest versions of browsers.

All these drawbacks of selenium RC are overcome using Selenium Webdriver. Selenium WebDriver doesn't require separate selenium server to execute the automated tests, it supports different programming languages such as Python, Java, Ruby, Perl, Net PHP for executing automated scripts on latest versions of different browsers, it supports both web based and mobile applications. Its interaction with the browser is very fast since it handles it from operating system level.

B. Steps to Automate Test Cases to Implement Framework

Automation frameworks are classified into different types such as Keyword driven framework, Data driven framework, Hybrid framework and others [5]. Page Object Model with Selenium Webdriver can be used with any of the framework to test any web based application. The framework selection depends on the software web application that needs to be automated. Data driven framework or Hybrid framework are selected when there is a huge amount of data is needed in a test automation process of the software under test. Below steps demonstrate the creation of Data Driven Framework using Selenium webdriver. Architecture is the basic foundation of any software application. It should be robust enough to handle the desired functionalities in an effective manner. Having a test script with embedded test data has its own limitations for building platform independent framework. It has major drawbacks such as small changes in the test data needs test case corrections, it may not be a difficult task for the person who developed it but it will be a hectic job for others to handle it. Keeping all these factors it would be more helpful by having the test data in an external data source and using it for test execution, this is nothing but Data driven Approach. A good framework has qualities such as reusable, application- independent and data driven. The Architecture of the data driven components consists of following components such as Framework Components, Internal components and external data source components.

C. Role of Framework Components

Selenium Webdriver and Java are the main components of this framework. When the test scripts are executed these components carry out the logic processing of data driven scripts on the application that is under the test. Selenium Webdriver interacts directly operates at the operating system level to launch the browser. These components are involved in importing the excel sheets which consists of object repository, test scripts and test suite. It holds all the functionalities that are needed for reporting.

D. Role of Internal Components

The internal components mainly deal with Object Repository and the Global Variables that are used across the test cases. Knowing about the application objects, their property value and the description are carried out by Object Repository. Selenium recognizes the object properties corresponding to the application objects such as value and type and stores it in Object Repository. Knowledge of element locator is necessary to work with selenium automation testing tool. When selenium test run, it will first locate element from page and then performs the given action on it like click, type, select and others as specified in the test case. The selenium driver with the help of Object Repository identifies the application and executes the test cases according to the flow coded in the test scripts. If the test script contains some object property that are not added in the Object Repository then the test script will fail. Therefore it is necessary to have a good Object Repository that contains property value as it is in the application. Object Repository also contains separate Global Variables which are kept in common place so that they can be used by any test script. This is done to add object once and reuse it as needed.

E. Role of external components

Embedding data to test script results in redundant code if there is need to check same operation with different set of data hence to avoid this the different set of data that needs to be performed on single test case will be stored in the spreadsheet. The spread sheet holds different cases of inputs that need to be fed to the application under test. The good practice is to have these sheets in a common place. The report generation will be configured with the framework components so that the summary report and the detailed reports will be executed after the test execution. The reports consists of details such as execution duration, start time end time and the total number of test cases that and passed and the failed with appropriate root cause for the failure. These results help in performance analysis.

IV. EXPERIMENTAL RESULTS

Automated software testing is broadly accepted for businesses because of its added advantages such as it reduce the more amount of resource utilization, time efficient and contributes more for product quality assurance, but the automation process has its own set of investments and risks hence it is important to calculate the return on investment (ROI) before making a move to test automation process. A good test planning must be performed to make sure that the investments on test automation does not equal or exceed the benefits that will come from it.

4.1 Calculation of ROI

The return on investing in automation can be calculated by considering tangible and intangible factors. Tangible costs can be calculated by dividing the benefits by investment cost. The ROI must be calculated in comparison with the cost that will be incurred if done through manual testing. Tangible cost calculation is easy because it can be shown in terms of

sums and its measurable entity for example cost investments on tools that are needed for automation, return investments from automation. Calculation of intangible factors is not a straight forward process since it is non quantifiable. It deals with the factor such as lower compliance risk, product quantity and others. ROI is always an essential important element in making decision for test automation. Different types of costs and benefits are involved with both tangibles and intangibles. There are significant benefits associated with automated testing which are listed below

- Time Efficient- This factor is a most appropriate tangible factor which is more quantifiable for comparison. Automation testing is more time saving than compared with manual test execution.
- Repeatability - Automated tests executes repeatedly in the same fashion, every time when they are executed. It may not look like a big return, but when we consider how many steps would be executed manually when automated tests were not present matters and then we need to consider the count of mistakes might be made with manual effort.
- Traceability - Automated tests generates some logs as evidence of execution. These automated tests keep a good track of what was measured during execution and the results provided by the automation will have less variation when compared with manual testing.
- Availability - Automated tests can be started off manually or automatically to run at any hour of the day. They don't get tired and they can be made to work weekends without special arrangements.

V. CONCLUSION

This paper discussed about implementing a robust product specific automation framework for testing of web based and mobile applications using a test automation tool "Selenium webdriver". Using this framework the regression testing will be carried out easily. It also talks about ROI calculations before going for test automation and this metric provides a clear picture of the benefits of test automation against the manual testing process. Automated testing framework eases the testing activity by increasing the efficiency, test coverage and product quality.

REFERENCES

- [1] "Study and Analysis of Automation Testing Techniques" Vishawjyoti and Sachin Sharma Dept of Computer Applications, ManavRachna International University, Faridabad.
- [2] Prof. (Dr.) V. N. Maurya, Er. Rajender Kumar "Analytical Study on Manual vs. Automated Testing Using with Simplistic Cost Model" International Journal of Electronics and Electrical Engineering ISSN: 2277-7040 Volume 2 Issue 1, pp.24-35 January 2012.
- [3] J. Fernandes and A. Di Fonzo, "When to automate your testing (and when not to)", July 2010.
- [4] SeleniumHQ Browser Automation

“<http://www.seleniumhq.org>”

- [5] Vivek Kumar “Comparison of Manual and Automation Testing” *International Journal of Research in Science and Technology (IJRST)* 2012, Vol. No. 1, Issue No. V, Apr-Jun ISSN: 2249-0604.