



Automation Testing using Selenium IDE

Gundeep KaurResearch Scholar, Computer Engineering
Punjabi University, Patiala
India

Gundeepnarula0309@gmail.com

Himanshu AggarwalComputer Engineering Department
Punjabi University, Patiala
India

himanshu.pup@gmail.com

Gaurav GuptaComputer Engineering Department
Punjabi University, Patiala
India

gaurav.shakti@yahoo.com

Abstract – Testing is the process to uncover the defects in a program. Automation testing gives maximum results using minimum efforts. Selenium is a powerful tool for automating the testing procedure specially in the case of regression testing. Selenium IDE, which is a record and playback testing tool, helps to reduce human effort, chances of error and cost, and increases efficiency.

Keywords - Automation testing, web application testing, verification, validation, regression testing, Selenium IDE, Record and playback testing tool, open source testing tool.

I. INTRODUCTION

Testing is the process to uncover the defects in a program. The practice of quality assurance is growing many folds with the upcoming technology. As websites are getting complex, the testing techniques and tools need to evolve alongside. Selenium is the de facto tool for automating testing on a web browser. Selenium supports various programming languages, so it fits right into any skill set.

The practice of quality assurance is evolving with technology. As websites get more and more complex, test strategies, tools and practices need to grow and evolve along side. We need to invent ways of enhancing our techniques, building an automated test framework is the epitome in the world of software testing.

Test automation helps to avoid manual testing which requires a lot of time and effort. Automated tests are easily run for new product versions and automatically provide test results. It is important to choose the right test automation framework which would meet test goals best. Test automation framework has to provide test development and execution environment where white-box or black-box test cases are created and performed.

Functionality of Selenium IDE can be enhanced by making it work in areas which are not its core functions. This enhancement can be brought by building user extension and javascript functions. In Selenium, we can create our own function that can then be used throughout the tests to extend the functionality of Selenium IDE. User extensions are stored in a separate file that we will tell Selenium IDE or Selenium RC to use. To create our own user extension you need to create Java script methods and add them to the selenium object prototype.

Software testing is the method of evaluating software to find out mismatch between given input and required output.

Software testing is done during the development process. We can also say that software testing is a verification and validation process.

Verification

Verification is the process to ensure that the product is being built as per the conditions imposed at the beginning of the development phase. We can say that verification makes sure that we are building the product right.

Validation

Validation is the process which ensures that the product satisfies the particular requirements at the closing stage of the development phase. We can say that, validation makes sure that we are building the right product.

Once the application has been built, testing is performed. Based on the test results, the developer might have to change the application if desired output is not obtained. When the changes have been incorporated, the tester is required to perform the same tests again. This process of repeating the test cases over and over again is called regression testing. A huge number of variable constraints can be reused from the previous version's test cases, thus reducing the effort for resolving constraint values for those variables when we test a new version of the application [1].

II. WEB APPLICATION TESTING USING SELENIUM IDE

Testing a web application has two main objectives. First is to test the compatibility of the application with technical requirement. Second is to test the working of the web application itself [2]. Automation testing automates the process of manual testing using tools like QTP, Selenium etc. Automation helps to increase the test execution speed, is more reliable, effortlessly repeatable, easily programmable, and fluently reusable [3].

Automated software testing is an essential element component of successful development projects. Automation testing draws maximum benefits with minimum efforts by increasing the quality and reliability of the products.

There are various tools which are used in automation testing. These tools are able to predefined actions, compare the outcome to the expected behavior and report the hit or miss of these tests to a tester. Once automated tests are created, they can effortlessly be repeated and they can be extended to perform task unfeasible with manual testing.



There are various advantages of test automation like it improves accuracy, saves time and money, test coverage, it can do which manual testing cannot, and it improves team morale. There are number of commercial and open source tools are available for assisting with the development of test automation.

A. Selenium Test Package

Selenium is a package of various test components which consists of the following three major tools. Each one has a specific role in aiding the development of test automation for a Web application.[5]

Selenium IDE – A Firefox extension which has the capability to record test cases.

Selenium RC – overcoming the shortcoming of IDE, RC can run tests on various browsers.

Selenium Grid – has the capability of running multiple instances of Selenium RC in a single go.

B. Selenium IDE

Selenium IDE (Integrated Development Environment) is a tool used to develop automated test scripts. It has record-and-playback functionality [6], but we can also write the test scripts ourselves by manually entering the commands. It comes with a browser context menu with the most frequently used Selenium commands. It saves test execution time and is a good way to get familiar with test automation. Selenium IDE comes as a Firefox plug-in and has an easy user interface.

C. When and Why to use Selenium IDE in a test cycle[7]

When-

A system testing life cycle comprises of frequent regression testing

Repeated test case Execution is required

User Acceptance Tests

Faster Feedback to the developers

Reduce the Human Effort

Why-

Selenium is an open source tool with Corporate backing.

Selenium deploys on Windows, Linux, and Macintosh platforms.

It allows recording, editing, and debugging tests.

Recorded tests can be exported in most language e.g. html, Java, .net, perl, ruby etc.

Selenium has the support of some of the largest browser vendors who have taken (or are taking) steps to make Selenium a native part of their browser.

TABLE I

Comparison between manual testing and automated testing using selenium ide[4]

PARAMETERS	MANUAL TESTING	AUTOMATED SELENIUM IDE FRAMEWORK
------------	----------------	----------------------------------

Reliability	not reliable and/or accurate at all times	Its more reliable and accurate.
Speed	Speed depends on human capability	Significantly faster
Cost	Requires high investment in human resources, testing infrastructure, etc.	Selenium IDE is an open-source tool and number of resources required is significantly less.
Frequent Executions Possible	Depends upon timelines.	can be iterated frequently.
Browser Compatibility	All browsers	Mozilla Firefox
Mundane & tiresome	Yes	No
Dealing with Code related information	No use of programming	Possible using programming.
Complexity of Test Data	Is utilized with very less variation in test data.	Can be used with extensive test data.
Languages supported	Not applicable	Supports several programming languages Java, C#, PHP, Python
Applicability	Applicable for any type of application (WebApp, system, web service, etc.)	Is applicable only for web applications
Specific Application component testing	Testing of third-party APIs is possible. Testing of dialog boxes is possible.	Testing of third-party APIs is not possible. Testing of dialog boxes is possible partially.
Drawbacks	Time consuming and very tedious.	For small change in web application UI the script may have to be rewritten.

Features:

**D. Selenium IDE components[10]**

Stable release	Version 1.0
Supported languages	C#, Java, Perl, PHP, Python, and Ruby
Supported browsers	Firefox
Ease of writing tests	Excellent
Scope of test operations	Good
Performance	Sufficient
Ease of installation	Good

Selenium IDE is divided into sections which are marked in the screenshot. Each section is explained below:

1. The menubar has options to open, add and save test scripts and suites. It has edit options and a preferred programming language can also be specified.

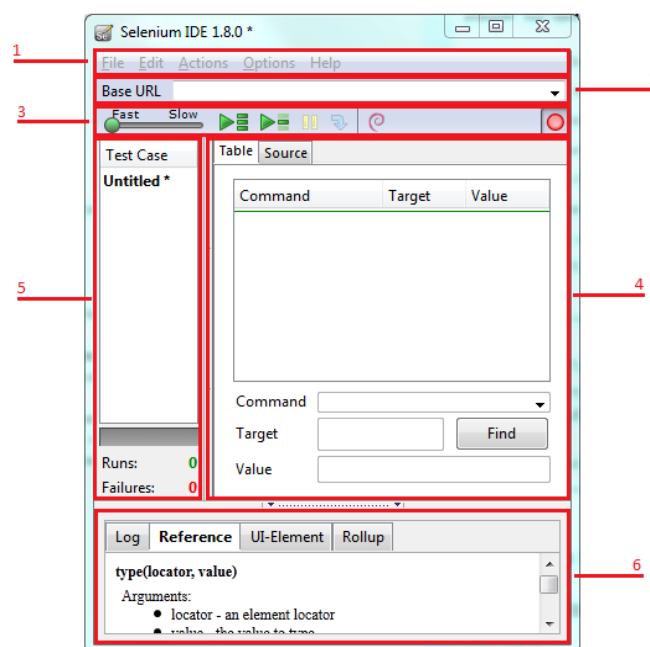
2. Base URL - Allows tests to be executed across various domains.

3. Toolbar has buttons which can control the test scripts. The function of each button is narrated here:

- Speed Control: checks the speed of execution of test scripts
- Run All: Run all test scripts in a single click.
- Run: Runs only the specific script which has been selected.
- Pause/Resume: used to Pause or resume a running test script.
- Step: Step through the test script once it has been paused.
- Apply Rollup Rules: Allows us to group a sequence of actions into a single action.
- Record: helps to record whatever action the user is performing on the screen.

4. Test script pane – used to display the test script commands. There is one tab which shows us the script in the three-column table format. The other tab shows us the script in a preferred language.

The Command, Target and value text fields are corresponding to the three-column table structure. It is possible to edit the values by clicking on a specific row. Those three fields are explained below:



• Command: This select box has a list of all the Selenium commands. It is possible to type into it to use the auto complete functionality or use it as a dropdown. Some reference information is shown in the reference pane at the bottom once a specific action is selected.

• Action: Parameter specified for the command in the Reference tab of bottom pane always goes in Target field.

• Value: If a second parameter is specified by the Reference tab, it always goes in the Value field.

5. Test suite pane- Shows the test scripts in the current test-suite along with the results.

6. Log/Reference pane - The bottom pane is used for four different functions divided in separate tabs, like: Log, Reference, UI-Element, and Rollup information. Each function is explained below:

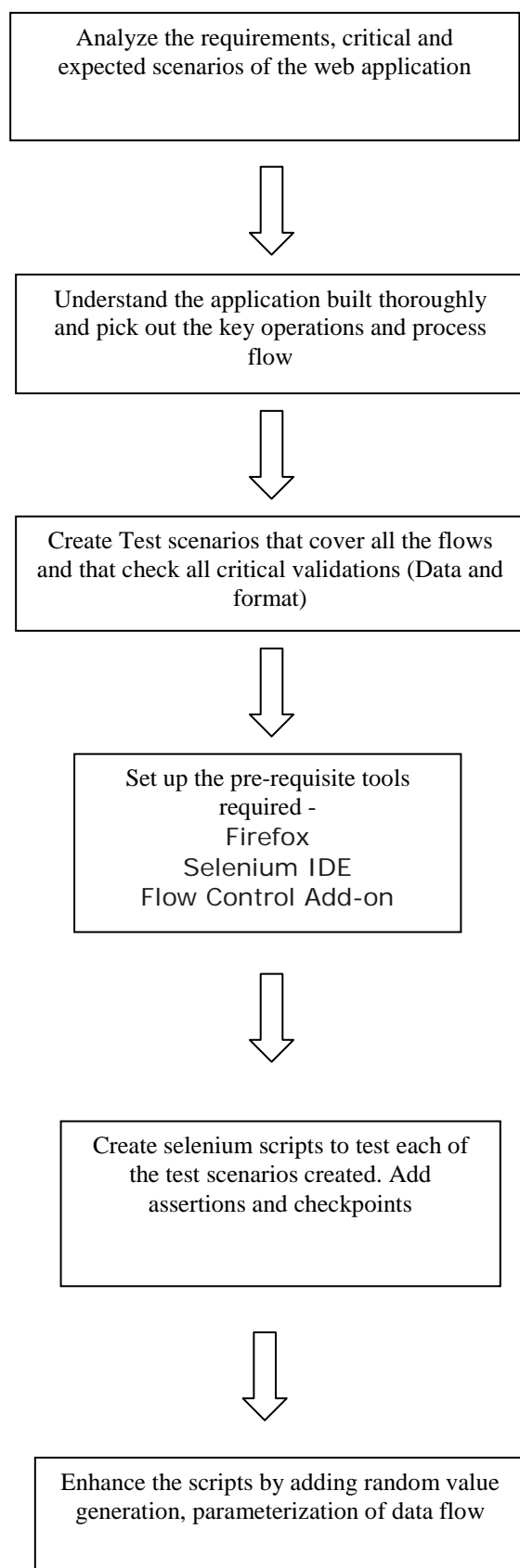
• Log: Error messages and information messages are shown while the test is running.

•Reference: Displays the documentation of the command actually selected.



III. MATERIAL & METHODS

Blueprint of building an automated test framework using Selenium IDE is given below:



Selenium IDE Commands

A command tells Selenium what is to be done. There are three basic types of Selenium commands: Actions, Accessors and Assertions.

Actions work to manipulate the state of the application, such as "click this link" and "select that option". The execution of the current test is terminated, if an Action has an error or fails.

Many Actions can be called with the "AndWait" suffix (e.g. "clickAndWait"). This suffix tells Selenium that it should wait for a new page to load.

Accessors observe the status of the application and store the outcome in variables. They are also used to generate Assertions.

Assertions are like Accessors, but they verify that the state of the application conforms to what is expected. For instance to "make sure the page title is X" and "verify that this checkbox is checked".

All Selenium Assertions can be used in 3 modes: "assert", "verify", and "waitFor". For example, you can "assertText", "verifyText" and "waitForText". When an "assert" fails, the test is aborted. When a "verify" fails, the test will continue execution, logging the failure.

"waitFor" commands wait for some condition to become true. They will succeed immediately if the condition is already true. However, they will fail and halt the test if the condition does not become true within the current timeout setting.

For commands related to asserts, value can be used to compare the value. For example :-

command=assertText, Target=Label and Value=somethingtocompare.

* Element Locators tell Selenium which HTML element a command refers to. Many commands require an Element Locator as the "target" attribute. Element Locators include "elementId" and "document.forms[0].element".

Some of the frequently used Selenium commands are as follows :- [9]

1. open - opens a page using a URL

Eg : - Open Firefox browser → Tools → Selenium IDE → Give following command and click on run

Command	Open
:	
Target:	http://www.google.com
Value:	

2. openWindow - opens a page using a URL in a new tab

Eg : - Open Firefox browser → Tools → Selenium IDE → Give following command and click on run



Command:	openWindow
Target:	http://www.google.com
Value:	

3. click/clickAndWait - performs a click operation, and waits for a new page to load. Useful for form submissions. The “wait” portion will wait to continue the testing when the next page has fully loaded.
4. verifyTitle/assertTitle - verifies an expected page title
5. verifyTextPresent - verifies expected text is someplace on the page
6. verifyElementPresent - verifies an expected UI element, as defined by the HTML tag, is present somewhere on the page
7. verifyText - verifies expected text is present on the page
8. assertTextPresent - checks to see if the text is present in the last rendered page
9. click - Clicks on a link, button, checkbox or radio button
10. doubleClick - Double clicks on a link, button,checkbox or radio button
11. close - Simulates the user clicking the “close” button in the title bar of a popup window or tab.
12. goBack - Simulates the user clicking the “back” button on their browser.
13. verifyValue - Verifies the value of an input field. Provide the id of the input as the target and the value is the text expected.
14. select - select an option in a select box by providing the id of the select box and the option value.
15. Start - Launches the browser with a new Selenium session.
16. stop - Ends the test session, killing the browser .
17. submit - Submit the specified form.
18. type - Sets the value of an input field, as though you typed it in.
19. unCheck - Uncheck a toggle-button (checkbox/radio)
20. getLocation - Gets the absolute URL of the current page.

IV. RESULTS & DISCUSSIONS

Selenium IDE has the following limitations:

Selenium IDE tool can only be used in Mozilla Firefox as it is an Add-on for only that browser and hence it cannot be used with other browsers.

It supports limited function.

It is a record and playback tool and the script format can be written in Selenese.

Cannot run concurrent tests.

REFERENCES

- [1] Md. Hossain, Hyunsook Do, Ravi Eda, “Regression testing for web applications using reusable constraint values” IEEE International Conference on Software Testing, Verification, and Validation Workshops, 2014
- [2] Harry M. Sneed, Ancon GmbH, Vienna, Austria & Budapest, Hungary, “Testing a Web Application”, Proceedings of the sixth IEEE International workshop on the website evaluation, 1550-4441/04 \$ 20.00 IEEE, 2004
- [3] Richa Rattan, Shallu, “performance evaluation and Comparison of software testing tools”, International Journal of Information and Computation Technology, ISSN 0974-2239 Volume 3, Number 7 (2013), pp. 711-716
- [4] Chandraprabha, Ajeet Kumar, Sajal Saxena, “Systematic testing of web testing tool : Selenium”, International Journal Of Advance Research In Science And Engineering, IJARSE, Vol. No.2, Issue No.11, November 2013.
- [5] Selenium page on Wikipedia [Online]. Available: http://en.wikipedia.org/wiki/Selenium_%28software%29 as on 19-01-2015
- [6] Selenium Website [Online]. Available : <http://www.seleniumhq.org/>
- [7] Navaraj Javvaji, Anand Sathiyaseelan, Uma Maheswari Selvan, Avacorp technologies pvt. Ltd, Chennai, “Data Driven Automation Testing of Web applications using Selenium”, Conference preceedings, STEP-AUTO, ISQT, 2011
- [8] Antawan Holmes and Marc Kellogg, Digital Focus, “Automating Functional tests using Selenium”, Proceedings of AGILE 2006 conference, 0-7695-2562-8/06 \$20.00 © 2006
- [9] Selenium Tutorials on <http://www.guru99.com/selenium-tutorial.html>
- [10] <http://selenium.polteq.com/category/selenium-ide/>