RESEARCH ARTICLE OPEN ACCESS

Comparative Study of Software Automation Testing Tools: OpenScript and Selenium

Vaibhav Jain*, Dr. Kumar Rajnish**
*(M.Tech. in Computer Science (Research Scholar), Birla Institute of Technology, Mesra, Ranchi)
** (Associate Professor, Computer Science and Engineering Department, Birla Institute of Technology, Mesra, Ranchi)

ABSTRACT
Software Testing ensures delivery of good quality software to the customer. With the increase of product functionality, testing becomes challenging as it is time-consuming to perform manually, cost also increases as test suite size grows and human error can creep into a system which may lead to losses for the organization. Therefore, automation of software testing for the product is needed using appropriate Automation tool, which can enable developers and testers to easily automate the entire flow or process of automation testing. The objective of this paper is to conduct a comparative study of automated tools i.e. the OpenScript and Selenium-based on different criteria`s.

***Keywords* -** Oracle OpenScript, OATS OpenScript, Selenium, Automation, Testing

Date of Submission: 28-02-2018 Date of acceptance: 13-03-2018

I. INTRODUCTION

Software testing is performed to make software defect free and to ensure that every effort used in development as per requirements works in a defined manner. It involves execution of a software component or system components to evaluate one or more properties and verify that it satisfies specified requirements or to identify differences between expected and actual results. Software Testing is of two types - manual and automation. Java-based application automation is been used for functional testing by using above mentioned two automated testing tools.

Manual testing [1] is testing of the product by a human manually as per the procedures defined or documented to test the proper functioning of the software product. Manual Testers discover the imperfections manually which expects them to act as end client and utilize highlights of the application to guarantee its right and correct functioning. They follow a written composed test plan that leads them to an arrangement of imperative experiments. The issues with manual testing are, it is the extremely tedious process, not reusable, great human efforts required, and few mistakes may still remain unrevealed. Automation testing covers most of the issues of manual testing.

Automation testing [2] automates the steps of manual testing using automation tools. Automating the test case execution for given software is a need for software on time delivery as it improves reliability factor, saves time and human efforts increase productivity and also decreases cost in long run. It expands the test execution speed, make them more reliable, programmable, exhaustive, and reusable. Standard Practices [3] need to be followed for testing applications in correct manner.

II. PROBLEM STATEMENT

Various software testing automation tools are available currently in the market but, not every tool has the capabilities to perform all type of testing like Functional, Load, Performance, Unit etc. and lack capabilities to interact/detect most of software development frameworks. So, it is of utmost importance to understand the application to be automated and using appropriate software testing automation tool to simplify code and testing framework development. For this study, software testing automation tool are compared on different criteria`s which are discussed further [4] [5].

2.1 Literature Survey

This section contains a brief mention of existing research papers which are used as references for this study. User Interface has changed a lot over recent years. The paper “Testing Tools (software)” by M. Lutz [8] surveys a set of tools that support the testing process in different ways. Some tools simulate the final execution environment as a way of expediting test execution, others automate the development of test plans, and still, others collect performance data during execution. In these tough economic times, more and better testing done faster. The automated testing tools facilitate higher quality and more productive testing, but acquiring such tools is often complicated. The paper “Evaluation and selecting testing tools” by Poston and Sexton [5] had proposed the evaluation criteria for selecting the testing tools that not only help the managers but technical team to select the appropriate automation tool.

The paper “A survey on testing and reuse” by Torkar, Richard and Stefan [4] gives a survey which tries to give an account of what type of trends exist today in software reuse and testing. Book “Selenium Testing Tools Cookbook” by Unmesh Gundecha [6] and paper “Design and implementation in selenium ide with web driver” by Uppal, Nidhika and Vinay Chopra [9] gives a detailed in-depth about Selenium, its four components i.e. IDE, RC, Web Driver, and Grid. While official document of Oracle OpenScript [7] is programmers guide of OpenScript which describes primarily the technical implementations for automation engineer and about the tool. The focus was to try to find out how developers use different tools today and what is lacking, especially in the field of reuse and testing. The paper “Classification of Software testing Tools Based on the Software Testing Methods” by Khaled, Rafa and Mohammad [10] classify and distribute a set of testing tools over the types of testing (testing methods) for three types of software products (web application, application software, and network protocol).

The paper “Logical capture/replay” by Silverstein [11] told us if there exists a reasonably well-structured system implementation, it is very easy to add in a mechanism to capture interactions with operations that system provides and to generate playback that are meaningful. Performing operation-centric capture/replay avoids many of the pitfalls of traditional GUI centric capture/replay. Paper titled “Comparative Study of Automated Testing Tools : Selenium, Quick Test Professional and Testcomplete” by H. Kaur and G. Gupta [12] has describes the comparative study between tools i.e. Selenium, HP Quick Test Professional and TestComplete functional testing tools on nine criteria`s.

III. EVALUATION STUDY

For this study OATS OpenScript 13.1.0.1 Build 363 and Selenium 3.4.0 versions were used. Comparison between these two tools is made on the basis of following parameters [12]:

1. Recording Efficiency of Tool
2. Capability of generation of scripts
3. Hybrid-Driven Framework
4. Test result report
5. Reusability of code
6. Execution speed of scripts
7. Scripts Playback
8. IDE features
9. Licensing Cost
10. Learning Ease
11. Application Support

3.1 Recording Efficiency of Tool

Both tools have capabilities of recording and playing back scripts [6] [7]. Using record feature, the tool will automatically insert the commands/code in the script. Both tools have easy access to controls while recording, Selenium recording engine icon present in the tool blinks which indicates it recording the user actions. Recording toolbar was also there which has all the controls. So, checkpoints can be easily applied, also add text and also see screen coordinates and window coordinates. Selenium provides different types of recording such as keyword, script, low level procedure based on screen or window coordinates, and HTTP task. Selenium and OpenScript both are good for web based applications/products automation only and are very stable. When the record button is pressed, application is started. It records all the actions as performed by user. Inserting checkpoint during recording in OpenScript is available.

3.2 Capability of generation of Scripts

Selenium can generate various types of scripts i.e. C#, Java, Python, Ruby for Web Driver and Remote Control. But, OpenScript generates only Java code. Selenium do provide the capability for code conversion by Exporting Test Case or Test Suite to Ruby, Python 2, Java and C#.

3.3 Hybrid-Driven Framework

Nowadays, Hybrid-Driven Framework becomes very important part of testing framework. Primarily, it involves use of combination of various other frameworks i.e. Functional Decomposition Framework, Data-Driven Framework, Keyword-Driven Framework and Page Object Model (POM). Combination of all these Frameworks/models are used these days and both tools support all these frameworks thereby supporting functional libraries to be part of Asset Scripts which will be used to call common methods. It is possible to make the scripts access the different sets of input data from external source line databases and CSV files etc. OpenScript [10] allows various types of Data Sources to be added as part of Assets (see Fig. 1) Databanks, it can be a CSV File or a database connection string. In our study, input is taken from database. After mapping, on clicking the Playback button to run the script and the result will be listed in Result tab and output in Console Output tab.

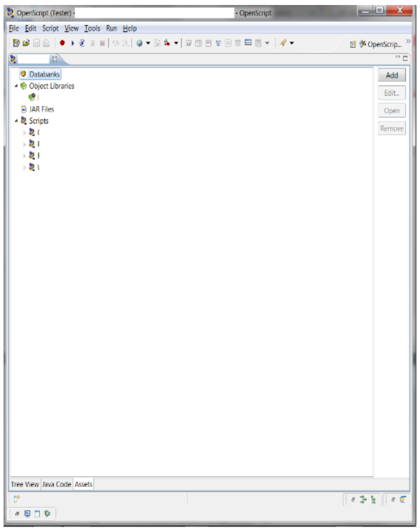


Figure 1 - Assets in OpenScript

Selenium [10] also supports all these frameworks by using inbuilt data tables which have functionality like excel sheet, XML, JSON, YML, Database. Using these sources we can reduce efforts of maintaining and easy mapping of columns to the input elements by even a novice user. We can insert foundation table data as parameters into our test scripts so that it will run several times on with different combinations of datasets. Each test run on a different set of data is called iteration. Data tables are of two types: static and dynamic data tables. Selenium IDE don`t support Data Driven testing.

3.4 Test Result Reports

Output after executing the test scripts should be evaluated whether script has passed or failed while running a test suite or a test case. OpenScript provides the result report under Results tab in html format that will be automatically saved in the folder of script repository on system. The generated result file can be expanded to check the script`s execution flow, OpenScript saves report details of previous 10 runs by default and runs can`t be compared. Generated reports are not graphical based and hence not easy to understand (see fig.3). Selenium [9] has no reporting by default bundled with the tool, external plugins like TestNG, JUnit etc. need to be used to generate reports for good graphical reports.

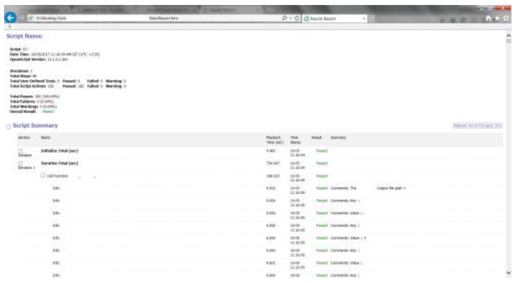


Figure 2 – Report generated by OpenScript

3.5 Reusability of code

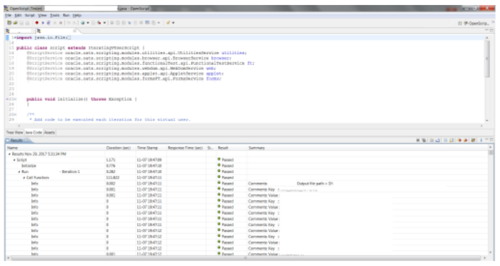
Reusing testing logic repeatedly is the ultimate goal of software test automation. OpenScript organizes all the references using Assets which forms the backbone of the automation in OpenScript. All the coding logic is in the form of user-defined JAVA scripts where scripts can be stored as function libraries. It is the folder in directory hierarchy where libraries as per product are saved and will be used frequently used by other scripts and contains common functions. The common scripts are reused easily on adding them as functional libraries. When application was modified by changing some properties of the objects, the same script can used on the new build. OpenScript has object repository where it recognizes and stores info such as object's properties i.e. XPATH. Selenium [9] can also work equally on reusing the code using methods i.e. defining the code logic within a block and then use it by calling it wherever required.

3.6 Execution Speed of scripts

Selenium is faster in executing scripts than OpenScript. Execution Speed do depends upon application or product under test that`s why their response time should be taken into account, as environment response will change the total execution time of script. On repetitive execution of scripts it was found Selenium is faster than OpenScript. Execution speed was calculated by taking total test run time of each script (i.e. start test run time + end test run time). Both Selenium and OpenScript result windows shows the start test run time and end time, and also total time taken in seconds.

In the case of Selenium,
Average time = $\Sigma T1 / n = 267/ 8= 33.38$ secs
where T1 is the time taken by each user screen in seconds and n is the total number of user screens.

In case of OpenScript,
Average time = $\Sigma T2 / n = 342/8= 42.75$ secs
where T2 is the time taken by each user screen in seconds and n is the total number of user screens.



```
import java.io.File;

public class script extends IteratingVUserScript {
    @ScriptService oracle.oats.scripting.modules.utilities.api.UtilitiesService utilities;
    @ScriptService oracle.oats.scripting.modules.browser.api.BrowserService browser;
    @ScriptService oracle.oats.scripting.modules.functionalTest.api.FunctionalTestService ft;
    @ScriptService oracle.oats.scripting.modules.webdom.api.WebDomService web;
    @ScriptService oracle.oats.scripting.modules.applet.api.AppletService applet;
    @ScriptService oracle.oats.scripting.modules.formsFT.api.FormsService forms;


    public void initialize() throws Exception {
    }

    /**
     * Add code to be executed each iteration for this virtual user.
```

Name	Duration (sec)	Time Stamp	Response Time (sec)	St...	Result	Summary
Results Nov 29, 2017 5:31:34 PM						
Script	1.171	11-07 18:47:09			Passed	
Initialize	0.776	11-07 18:47:10			Passed	
Run - Iteration 1	0.282	11-07 18:47:10			Passed	
Call Functions	111.822	11-07 18:47:11			Passed	
Info	0.002	11-07 18:47:11			Passed	Comments Output file path = D:\
Info	0.001	11-07 18:47:11			Passed	Comments Key : [illegible]
Info	0.001	11-07 18:47:11			Passed	Comments Value :
Info	0	11-07 18:47:11			Passed	Comments Key :
Info	0	11-07 18:47:11			Passed	Comments Value :
Info	0	11-07 18:47:11			Passed	Comments Key :
Info	0	11-07 18:47:11			Passed	Comments Value :
Info	0	11-07 18:47:11			Passed	Comments Key :
Info	0	11-07 18:47:12			Passed	Comments Value :
Info	0	11-07 18:47:12			Passed	Comments Key :
Info	0	11-07 18:47:12			Passed	Comments Value :
Info	0	11-07 18:47:12			Passed	Comments Key :
Info	0.001	11-07 18:47:12			Passed	Comments Value :

Figure 3 – Script Playback and Result`s Tab

3.7 Scripts Playback

When script is played [11], it replays the user actions that were performed by the user during recording. If object is not recognized during replay, it gives the error message as object is not found. In our study, both tools played the scripts efficiently.

3.8 IDE feature

OpenScript Integrated Development Environment (IDE) is having large feature sets which are similar to Eclipse IDEs. OpenScript [7] have two perspectives i.e. Tester Perspective and Developer Perspective. Selenium Suite has Selenium IDE which has limited features and WebDriver has extended features like that of OpenScript. OpenScript can interact with many browsers like Firefox, Internet Explorer and Chrome but Selenium WebDriver not only supports former mentioned browsers with support for Safari, Opera and Safari.

3.9 Licensing Cost

Selenium [6] is Open Source software testing tool and hence no license fee is involved. But, OpenScript is part of Oracle Application Testing Suite (OATS) which has licensing cost involved and OpenScript is bundled with it.

3.10 Learning Ease

Selenium is used for automating the web browser based applications and does not cater other desktop level application to a large extent and hence it can be learnt easily in small span of time but OpenScript has larger set of applications that can be automated and hence takes time to understand the tool capabilities and also it is in-house tool of Oracle and hence enhancements and support is available through organization portals only and new feature set will be released as per organization cycle. OpenScript has more features, and is more complicated than Selenium.

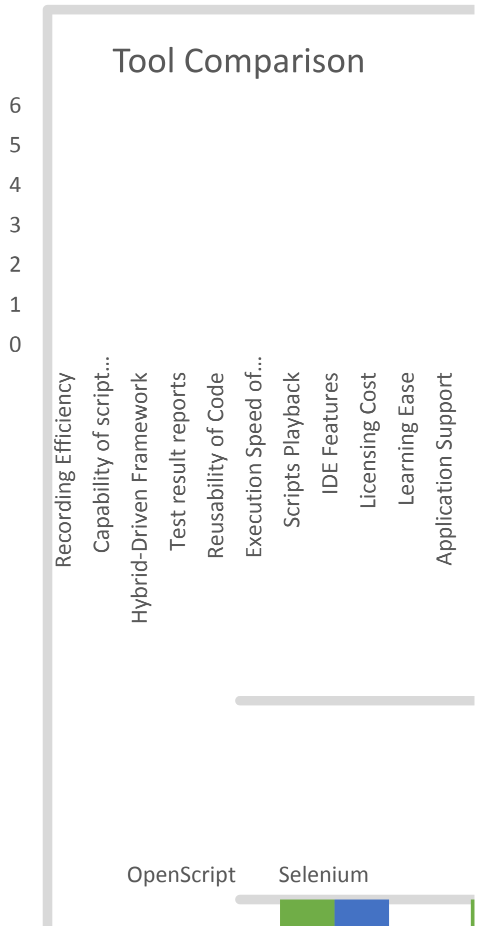
3.11 Application Support

Selenium can automate browser based applications and has open community that supports it, due to support dependency on community future upgrades and patches will be limited and would be delayed. Browsers supported are Mozilla Firefox, Internet Explorer, Safari, Opera and Chrome. Operating Systems supported are Microsoft Windows, Apple OS X and Linux. Programming Language and Frameworks are C#, Java, JavaScript, PHP, Python, R and Ruby.

OpenScript supports application of both Desktop and Browser based applications. Its capability to work with Java Applications is highly stable and with Oracle Forms. But, with Application Development Frameworks (ADF) it is still in evolving state and is currently improving. Browsers supported are Internet Explorer, Firefox, Chrome (version 33 or higher), Safari and Microsoft Edge. On Linux platform only Firefox browser is supported. Microsoft Windows is the supported operating system. Programming Language supported is Java version 7 or higher.

IV. CONCLUSION

Automated software testing is beneficial for large organizations. Since, in long run functionality of software products scales and hence their complexity increases. Thus, both time and money saving are essential for fast delivery of product/application across patches. Selenium and OpenScript both are efficient tools for software testing automation with their own features. Selenium has easy to use UI and efficient playback and OpenScript do have rich UI but it do have larger applications that can be automated. Deciding which tool is better after taking into account various aspects i.e. technology stack used in Application Development. Selenium can be used for applications which are Browser based applications but OpenScript is best for automating Java based frameworks primarily with browser application as well.



REFERENCES

[1]. Craig, Rick David; Stefan P. Jaskiel (2002). *Systematic Software Testing*. Artech House.

[2]. Dustin, Elfriede, Jeff Rashka, and John Paul. *Automated software testing: introduction, management, and performance*. Addison-Wesley Professional, 1999.

[3]. *IEEE Standard for Software and System Test Documentation," in IEEE Std 829-2008 , vol., no., pp.1-150*, July 18 2008

[4]. Torkar, Richard, and Stefan Mankefors. "*A survey on testing and reuse." In Software: Science, Technology and Engineering, 2003.* SwSTE'03. Proceedings. IEEE International Conference on, pp. 164-173. IEEE, 2003.

[5]. R. M. Poston and M. P. Sexton, "*Evaluating and selecting testing tools*," in IEEE Software, *vol. 9*, no. 3, pp. 33-42, May 1992

[6]. *Selenium Testing Tools Cookbook*, by Unmesh Gundecha, Packt Publishing, November 2012

[7]. Oracle® Functional Testing, OpenScript Programmer's Reference Release 13.2.0.1, October 2017

[8]. M. Lutz et al., "*Testing tools (software)*," in IEEE Software, *vol. 7, no. 3, pp. 53-57,* May 1990.

[9]. Uppal, Nidhika, and Vinay Chopra. "*Design and implementation in selenium ide with web driver*." International Journal of Computer Application 46 (2012): 8-11.

[10]. Khaled Mustafa, Rafa E. Al-Qutaish, Mohammad I. Muhairat, "*Classification of Software testing Tools Based on the Software Testing Methods*", 2009 second International Conference on Computer and Electrical Engineering, 978-0-7695-3925-6, 2009

[11]. Silverstein, M. I. C. H. A. E. L. "*Logical capture/replay*." STQE Magazine 5, no. 6 (2003): 36-42.

[12]. H. Kaur and G. Gupta, "*Comparative Study of Automated Testing Tools : Selenium , Quick Test Professional and Testcomplete*," vol. 3, no. 5, pp. 1739–1743, 2013.

Vaibhav Jain " Comparative Study of Software Automation Testing Tools: OpenScript and Selenium "International Journal of Engineering Research and Applications (IJERA), vol. 8, no. 2, 2018, pp. 29-33.