

Comparative Study of Automated Testing Tools

Aayush Nagpal

Computer Science Department

University of New Orleans

LA, USA

Email: anagpal@uno.edu

Abstract—Testing is a major component of software development process. During development cycles, tests need to be repeated many times to ensure the quality. As web applications become more complex, there is a growing concern about the quality. Considering end user usage of the application on various browsers, cross browser testing becomes an important aspect of testing. Automation testing addresses the challenges presented by manual testing thus reducing time and effort spent on testing exponentially. The primary objective of this study is to analyze and compare automated testing tools focusing especially on Acceptance Tests and UI testing of web based applications. Selenium, cucumber and helium are the tools of choice for this study. A comparison matrix is generated for the tools comparison and selected tools are rated based on several parameters such as performance, compatibility, ease of use etc.

Keywords—Selenium; WebDriver, Cucumber, Helium, TestNG, Automation.

I. INTRODUCTION

Manual testing is the ancient and most demanding way of testing the software which necessitates a tester to accomplish manual test processes to test the software. Repetitive manual testing is difficult to perform on large software applications or applications having very large dataset coverage. Test automation can automate some repetitive but necessary tasks in a formalized testing process. There are many advantages to test automation. Most are related to the repeatability of the tests and the speed at which the tests can be executed. There are several commercial and open source tools available for assisting with the development of test automation. Selenium WebDriver is possibly the most widely-used open source solution.

To automate or not to automate is an open debatable topic. However, it is not always beneficial to automate test cases. There are several scenarios where automation might not come across as a viable option such as, frequently changing user interface or lack of time to test. For short term goals, manual testing is an effective approach.

In the evolution of web systems and their test cases, the task of generating test cases to cover the existing and new functionality and the exploration of the web system's structure

are closely interwoven activities. The Web application programs testing, especially the regression testing is much more difficult than the traditional but on the other hand manual test case execution is not only a costly and time-consuming exercise but it is also prone to error. Automation testing addresses the challenges presented by manual testing. Automation testing helps find defects and errors which are easily overlooked by manual testing.

Identification of right automation tools is critical to ensure success of testing project. The time and effort spent on scientifically choosing a test automation framework is a valuable investment. Selecting a relevant tool would not only save time but would keep the cost of development process in check. Most of the time a tool is used without figuring out its advantages and disadvantages rather decision should be made in favor of a tool that is right for their specific purpose. The features that are provided by the tool are also very important in selection of right automation tool. The selection of tools depends on numerous factors such as:

- Testing Requirement
- The application and its stack needs to be tested
- Skill set available
- License cost.

Compatibility is an issue when developing a web application ready for various platforms. Browser compatibility is an important testing aspect of web application. It is important to carry out a research by comparing tools and framework that would help towards selecting right automation tool for browser based testing. This paper would be comparing the performance of various tools (Selenium WebDriver, Cucumber, Helium) and explains various features and limitation of Web based testing tools. The discussed tools are integrated with other testing framework to make the testing process more organized. Such as integrating Selenium and Cucumber and TestNG and JUnit respectively. To store execute and automate all process in an organized order build tools are used such as Ant, Maven etc. In this study, Ant and Maven would be used along with selenium and cucumber respectively.

The various sections of the paper are explained. The details of each tool are explained in Section II along with its various components. Research methodologies that are implemented to perform the comparative study are mentioned in Section III. Analysis and Results of the comparative study is showcased in Section IV.

II. BACKGROUND

A. Test Automation Framework

A “Test Automation Framework” is scaffolding that is laid to provide an execution environment for the automation test scripts. The framework provides the user with numerous benefits that helps them to develop, execute and report the automation test scripts efficiently. It includes the physical structures used for test creation. Testing framework has components such as:

- Well organized directory structure
- Set of standards and coding guidelines
- Location of test data
- Location of object repository(OR)
- Location of common functions
- Location of environment information
- Methods for running test scripts and location of results.

A well-defined test automation framework helps is reusability of code, provides maximum coverage along with low cost of maintenance and recovery scenario

B. Selenium WebDriver and TestNG

Selenium is composed of multiple software automation tools such as, Selenium IDE, Selenium RC (selenium 1.0) and Selenium webdriver(selenium 2.0). WebDriver supports multiple web browsers. WebDriver directly calls the browser when tests are performed in a native machine by using the browsers built-in JavaScript support for the automation. However, despite all advantages of selenium webdriver, it has some limitations when testing the web applications. Selenium webdriver does not have built in functionality to generate the screenshots for failure test cases.

TestNG is a testing framework inspired from JUnit and NUnit, but introducing some new functionalities that make it more powerful and easier to use. There are several motivations for using TestNG as choice of framework as it provides rich set of annotations and addresses the limitation of selenium by generating HTML reports, parallel testing and generating logs is also possible by using TestNG. Some example of annotations are given below:

- **@BeforeTest** – The annotated method will be run before any test method belonging to the classes inside the tag is run.
- **@AfterTest**: The annotated method will be run after all the test methods belonging to the classes inside the tag have run.
- **@BeforeMethod**: The annotated method will be run before each test method
- **@AfterMethod**: The annotated method will be run after each test method.
- **@Test**: The annotated method is a part of a test case.

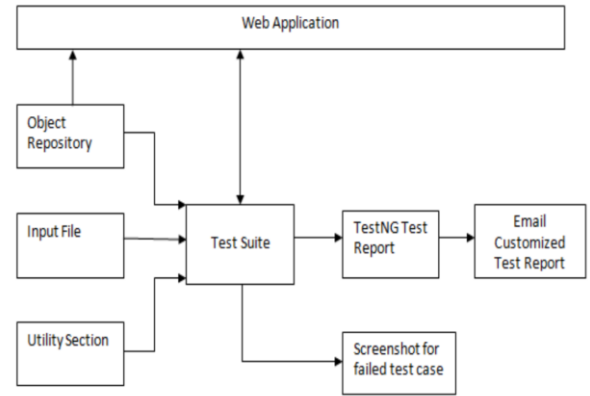


Fig. 1. Architecture of Proposed Framework

C. Cucumber

Cucumber is a testing framework that supports acceptance tests in Behavioral Driven Development. Cucumber requires features to be written in a DSL called Gherkin. With the help of Gherkin language cucumber helps to facilitate the discovery and use of ubiquitous language within the team which makes it easier for the understanding of management and stakeholders. Cucumber has three important sections feature file, step Definition and Runner Class. Gherkin requires each feature to be saved in a separate file Cucumber, tests are written in plain text in the format of Given, When and Then statements, where the

- **Given** statement specifies what all the pre-conditions are, the
- **When** statement specifies what all the conditions are, and the
- **Then** condition specifies what is the expected outcome

Step definitions are used to map the behavior described in scenarios to the implementation of that behavior in the system. In simple terms, this can be referred as main class in typical java program from where the execution starts. In Cucumber, the execution starts from the RunnerClass. An acceptance test is not complete without a test oracle that compares the output and state of the system to the expected output and state. The Then step defines the expectations its step definition is where expectations are compared to actual results.

In this study, Cucumber is used in integration with selenium and Junit. Below are the steps required in the setup:

- Create Maven project in eclipse
- Add Selenium jars and add cucumber jars
- Create feature file
- Create Step Definition
- Create Runner Class
- Generate Reports

D. Helium

Helium is a library that makes web automation as simple as giving instructions to a colleague, looking over his/her shoulder at a screen. There are several pros and cons of using Helium.

Helium removes the need of finding and locating html elements. There is no need for Xpath and other locators. Helium also reduces code complexity by nearly 50% as all WebDriver, WebElement references can be done in very less keywords. There is no need to call driver instances repeatedly to run test cases. Some other important aspects Helium tool is that it is paid tool and it requires including almost near to 14 jars in your project.

III. METHODOLOGY

To achieve the goal to compare several web automation tools, this research was performed in the following steps:

- Select target application to be tested.
- Research and select a set of tools to be evaluated.
- Develop a metric suite to be used to evaluate the tools.
- Perform a feature assessment of the selected tools.
- Test the target application using the selected tools.
- Evaluate and interpret results and the tools.
- Draw Inferences and make recommendation.

A. Application Under Test

The selected tools were evaluated using the proof of concept discussed in next section. For the purpose of evaluation, a large-scale university website is considered as application under test where many scenarios are executed in order to evaluate different aspect of the tools.

The AUT has many diverse set of features that can be tested in order to evaluate the various aspects of selected tools also AUT is under renovation which would be interesting to note as it would provide in depth evaluation of the selected tools for live projects.

B. Comparison Matrix

While selecting the right automation tools among the various others a comparison of key parameters helps select the right tool. Table I provides a comparison chart of tools based on most important parameters for automation projects. Information provided in the comparison matrix can be used to make recommendation for the right automation tool keeping the automation requirement in consideration.

Table 1 Comparison Matrix

Parameter	Criteria	Selenium	Cucumber	Helium
Ease of Adoption	License Cost	Open Source, Apache 2.0	Open Source, MIT	Open Source
	Ease of Support	High Community Support	Medium Community Support	Low Community Support
Ease of Scripting and Reporting Capabilities	Scripting Language	All Major Programming languages	All major programming languages	All major programming languages
	Object Recognition	FirePath, Firebug, Selenium IDE	FirePath, Firebug	No
	Script Creation Time	High	Low	Very Low
	Learning Curve	High	Medium	Low
	Script Execution Time	Low	Very Low	Low
	Framework	[TestNG, Unittest, Junit, Nunit, Rspec]	BDD	Standalone
	Test Result Generation	TestNG Reporting, Allure Framework	Maven Reporting	Eclipse Console
Tools Usage	Continuous Integration	Yes, Jenkins etc.	Yes, Jenkins	No
	Non-Browser based app	No	No	No
	Operating System	All major OS	All major OS	All major OS
	Browser support	All major browser	All major Browser	All major Browser
	Parallel Execution	Yes, Selenium grid	Yes	Yes
	Types of Testing	Browser Automation	Browser Automation	Browser Automation
	Support File Upload	Supports	Supports	Supports
	Install Required	No	No	No
	Cloud Based	No	No	No

C. Proof of concept

The last phase of tools evaluation is proof of concept (PoC). Evaluating the selected tool using PoC will provide more confidence in selecting the right tools. The following steps need to be considered during the PoC.

- Choose a few Scenarios that cover different parts of AUT
- Select tools based on comparative study
- Automate the chosen scenario.
- Generate and analyze the results
- Analyze the Integration of the tools with other tools.

IV. RESULTS AND ANALYSIS

The selection of a testing tool depends upon the tester's requirements. There is no single testing tool that performs everything. Certain tools are best suited for certain purposes the comparative study shows that Selenium is best suited for people with some experience and coding skill as it involves integrating various frameworks in order to implement high-level functionalities. Selenium tool is certainly an asset for anyone who is looking to add a powerful web testing tool to their toolkit.

Cucumber feature file hides the actual implementation of the selenium code, thereby making it easier for any anyone from a non-technical background also it bridges the gap between technology team and business team. Helium is basically a wrapper around selenium. Even though it provides abstraction and relatively less coding to achieve more automation, it is difficult to handle more complicated edge cases using helium. Helium also has relatively smaller community support. One last difference Helium has from other two selected tools is that helium is a paid tool whereas Selenium and cucumber are open source tools with large community base.

Screenshots of results are provided below for reference. Selenium reports are generated with the help of Surefire dependency (figure2) , Cucumber reports are generated with the help of maven dependencies (figure 3,4) and finally helium reports are provided in the eclipse console only.

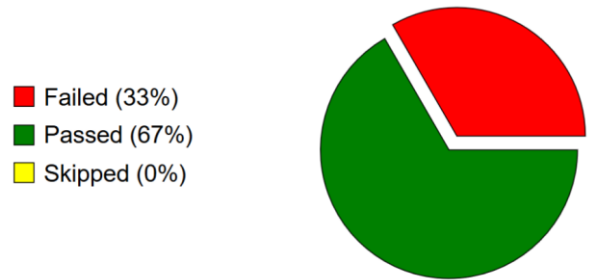


Figure 1 Selenium Test Suite Overview

Features Statistics

The following graphs show passing and failing statistics for features



Figure 2 Cucumber Feature Statistics

Steps Statistics

The following graph shows step statistics for this build. Below list is based on results, step does not provide information about result then is not listed below. Additionally @Before and @After are not counted because they are part of the scenarios, not steps.

Implementation	Occurrences	Duration	Average	Ratio
AllStepMethods.browserquit()	5	2s 952ms	590ms	100.00%
AllStepMethods.click(String)	4	3s 349ms	837ms	100.00%
AllStepMethods.getToNo(String)	5	31s 807ms	6s 361ms	100.00%
AllStepMethods.hoveron(String)	1	4s 871ms	4s 871ms	100.00%
AllStepMethods.i_verifyFooterLink(String)	5	1s 617ms	323ms	100.00%
AllStepMethods.i_verifyOptionsUnderTab(String)	7	3s 471ms	495ms	100.00%
AllStepMethods.i_verifyThePageDrifts(String)	2	4s 263ms	2s 131ms	50.00%
AllStepMethods.switchtochildwindow(String)	1	367ms	367ms	100.00%
8	30	52s 700ms	1s 756ms	Totals

Figure 3 Cucumber Steps Statistics

REFERENCES

- [1] Harpreet Kaur, Dr.Gagan Gupta, “Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete
- [2] Nidhika Uppal and Vinay Chopra “Design and Implementation in Selenium IDE with WebDriver” in International Journal of Computer Applications May 2012
- [3] Kauar H, Gupta G. Comparative study of automated testing tools: Selenium, quick test professional and testcomplete. International Journal of Engineering Research and Applications. 2013; 3(5):1739–43.
- [4] Bordelon N, Simmonds D, Reinicke B, Patterson L, Noble J. A comparison of automated software testing tools. Annals of the Master of Science in Computer Science and Information Systems at UNC Wilmington. 2012; 6(1):1–77
- [5] Mohamed M, El-mahdy MM. Evaluation of automated web testing tools. International Journal of Computer Applications Technology and Research. 2015; 4(5):405–8.
- [6] MaurizioLeotta,DiegoClerissi,FilippoRicca,CristianoSpadoaro. Comparing the Maintainability of Selenium WebDriver Test Suites Employing Different Locators; ACM,2013.
- [7] Selenium Documentation.[Online].(<http://www.seleniumhq.org>). (Accessed 01 MAY. 2017).
- [8] TestNG Documentation.[Online].(<http://www.testng.org>). (Accessed 01 MAY. 2017).