International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 15 (2014), pp. 1507-1518
© International Research Publications House
http://www. irphouse.com

Comparative Analysis of Open Source Automated Software Testing Tools: Selenium, Sikuli and Watir

Inderjeet Singh and Bindia Tarika

Guru Nanak Dev Engineering College Ludhiana, inderjeetsinghit@gmail.com
Guru Nanak Dev Engineering College Ludhiana, bindiatarika11@gmail.com

Abstract

Testing is one of the important aspects of Software Engineering and there is wide availability of open source software testing tools in this category of which Selenium, Sikuli and Watir are usually used open source automated testing tools. This paper presents the comparative analysis of these different tools in terms of their recording capabilities, Data Driven Testing, Efficiency, Languages Supported, Test and Code Reusability that conclude the effectiveness of testing tool under these parameters

Keywords: Selenium RC, Selenium IDE, Selenium Web Driver, Sikuli, Watir, Eclipse

Introduction

Software testing is one the main component in developing software successfully and to ensure its correctness about the operation which it is expected to perform under different input cases. Various strategies or methods exist for performing software testing, major of which are Black Box and White Box Testing.

Where Black Box testing gives an abstract view of operations performed by entire software, White Box testing gives detailed view about how the process of the software are carried out [1]. Other Classification of the Software testing techniques can be done accordingly how the testing is carried out [2]. According to this classification testing can be either Manual Testing or Automated Testing

Manual Testing is carried out by preparing test cases manually and is more prone to human errors where as Automatic Testing is carried out by recording the various test cases on the basis of what actions had user performed. This saves a lot of time in writing test cases manually and improving the efficiency. But also along with these above classification there is one another on the basis of the license associated with testing tools which comprises of Propriety Tools and Free and Open source Tools.

Proprietary tools include those testing tools which are closed source in nature and license has to be purchased so as to harness their functionalities to full extent. Example under this category is Test Complete and Quick Test Pro [2]. Open Source automated software testing tools however may not require purchase of license and the code of the application is available to the user for further modifications to be performed [3].

Problem Formulation

Software testing is necessary to ensure the smooth execution of the application developed. It is also important because it covers a wide variety of test cases and the even worst situation that the application may have to face when put up in the real environment. The comparative analysis is done of three open source testing tools viz. Selenium, Sikuli and Watir for providing user an easy selection process and to save the extra time wasted in checking and installing every individual tool [2].

Open Source Software Testing Tools

Open Source Software testing tools like Selenium, Watir are quite efficient non-proprietary tools and are available under GPL license that is distributed freely.

But all these tools differ in how they perform their intended tasks and the goodness of the features that they are expected to deliver. So, the problem lies in terms of two major questions i.e.

- How to differentiate these tools on basis of some common feature?
- Which Open Source Testing tool in general can be optimum?

To answer these queries analysis of these major open source tools was required which is presented in this paper.

Background Analysis

Selenium

Selenium is a one of the efficient open-source automated testing tool which provide a nice testing framework for testing wide variety of applications exporting scripts in almost every language including java,.net,c#. The main feature of Selenium is multi-browser support for execution of test cases [5]. Various variants of Selenium exist and these are [4]:-

- Selenium IDE which is used only in firefox and used as plugin.
- Selenium RC which supports multiple browsers and scripting in different languages.
- Selenium Web Driver that supports all browsers for execution.

Sikuli

Sikuli is one tool that uses images to generate test cases and automate testing [7] for the application under consideration.

It can be used with an ease and similar to that of Selenium and can also be used for various Web based applications and also for PC applications.

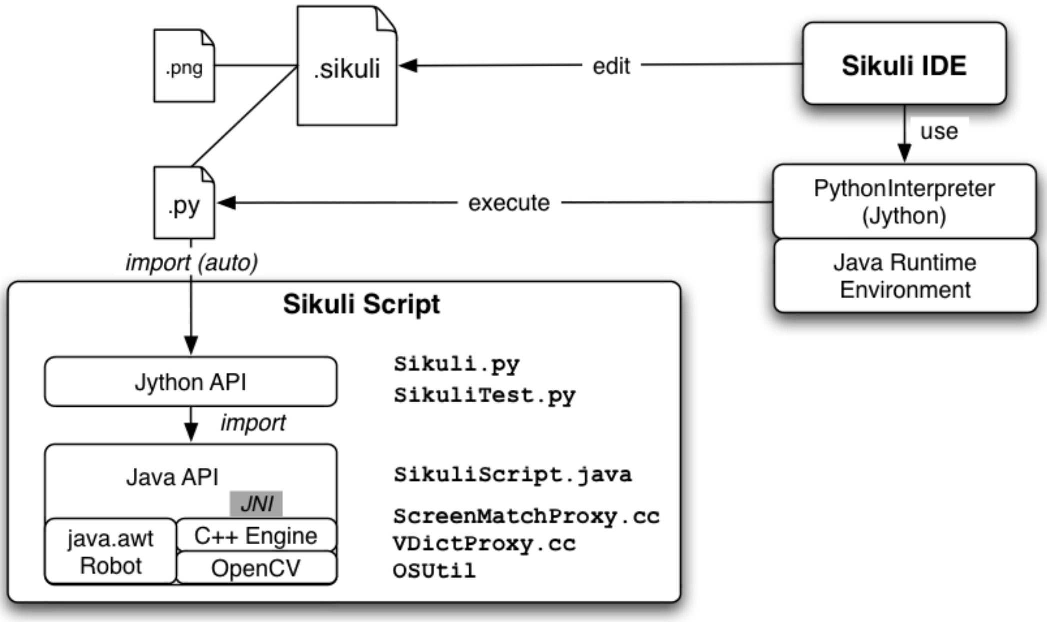


Figure 1 Sikuli Workflow [6]

Watir

Watir is a set of Open Source Ruby Libraries released under BSD license to test the various web based applications and pronounced as Water [8]. It is available as RubyGems and capable of driving variety of browser including the major like Internet Explorer, Firefox etc [9]. Bret Pettichord and Paul Rogers developed Watir. Watir project is composed of several other projects of which watir-classic, watirspec and watirwebdriver are important.

- Watir-Classic works on the fact that it has capabilities of Object Linking and Embedding [10].
- Watir-WebDriver is the modern implementation of testing functionality by watir working as an API which is selenium based and to which Jari Bakken had implemented using the HTML specification which is compatible with wide variety of W3C standards [11].
- Watir-Spec is (like RubySpec is for Ruby and) executable specification for the API of Watir [8].

Evaluation Study

There are number of factors on the basis of which comparison of these three open source software testing tools was done and they are as follow.

Recording Capabilities

Recording Capabilities define that how effectively the tool is able to record for the test cases. Here, among three stated testing tools Selenium provides an easy way in terms of Firefox plugin for recording and Watir provides IDE support but that is not native.

Execution Speed

It matters a lot to check for the execution speed when it comes to performance comparison of various tools. Here the Execution speed is determined in terms of the average test run time of number of transaction then evaluation is done so as to evaluate which one is best in terms of speed.

Scripts Generation

Test cases can be exported to equivalent codes in wide variety of languages. Comparison is done with respect to the ability of the tool to support large number of languages in which code can be exported.

Data Driven Testing

Data driven Testing includes testing in accordance with various data sets that can be import form the file e.g. from an excel file. Selenium provides data driven testing by using the wide range of sources viz. Excel files, XML files etc [12].

Watir provides few external sources access [13] and Sikuli provide rare access to the external sources for Data driven testing [14].

Ease of Learning

Selenium comes with different variants and hence provide wide perspective to explore the capabilities and utilize the entire functionality of Selenium Web Driver, RC and IDE [4].

Sikuli provides a very easy to learn graphically identifiable events based testing capable of defining its working with few constructs available. One can masters with the use of Sikuli in a very short span [7].

Watir hereby can be proven to be quite tough in understanding what the codes says as it is mandatory to learn Ruby to be able to get through the coding of the stuff [8].

Testing Reports/Output

Testing output if presented in well-format works like icing on the cake and adds much value to the testing tool. One if not able to understand what the testing tool wants to depict or is not able to produce the results in easy readable and understandable format is of less use to the tester.

Supplement Features

Cost is one of the important factor and plays an important when it comes to comparison of various vendors/same vendors with different products and can played as the sole criteria in some cases. Here this factor has equivalent impact in all of the three products as all of them are Open Source in Nature.

Results

5 point based system has opted for evaluating various parameters. i.e. 5, 4,3,2,1 as Excellent, Very Good, Fair, Bad, Very Bad respectively. Every Parameter corresponds to some important metric that helps to evaluate the effectiveness, usefulness of testing tool. Average is taken of point assigned to each sub criteria for each parameter that calculates complete score.

Execution Performance

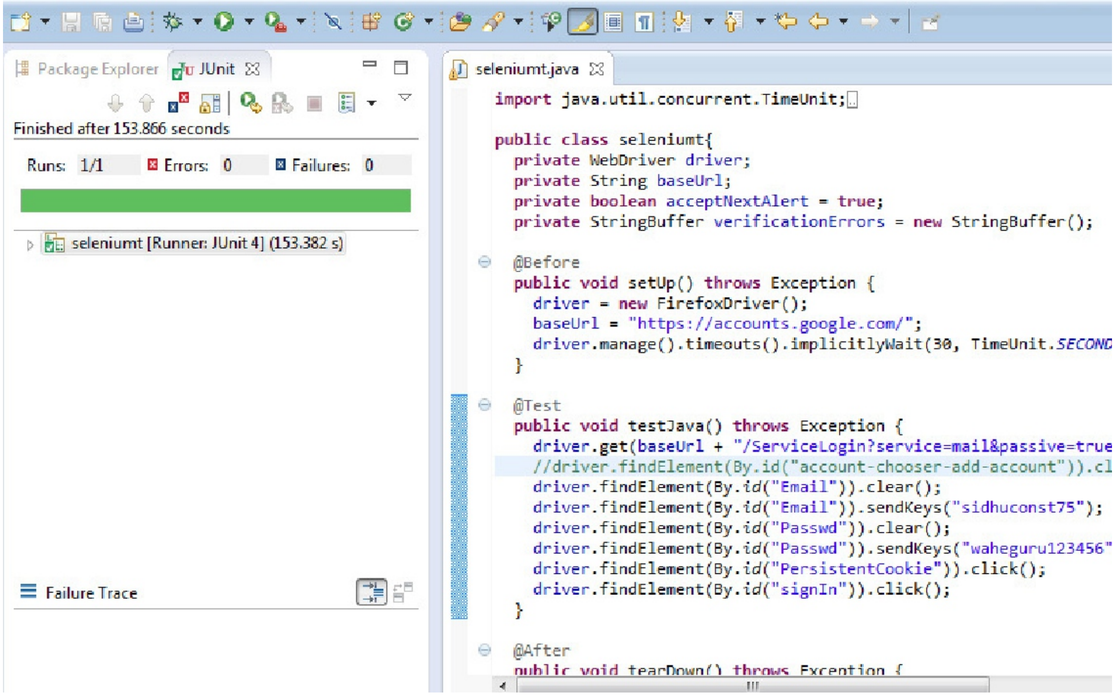
Execution Performance matters a lot when it comes to performance measures because it depicts the fortune of the scripts executed. For calculating the execution time a simple test case of login into the gmail account is considered for the scripts to be executed on all three different

Calculating Execution Time for Selenium

Selenium RC is used as server listening to calls that are made to run the test case on default browser. Test cases are written in Java and recorded using Selenium IDE plugin in firefox. Following is the code written for executing the test case [15].

```
public class seleniumt{
  private WebDriver driver;
public void setUp() throws Exception{
    driver = new FirefoxDriver();
    Url= "https://accounts.google.com"; }
  @Test
public void testJava() throws Exception{
    driver.get(Url                                                    +
"/ServiceLogin?service=mail&passive=true&rm=false&continue=http://mail.google.c
om/mail/
   &scc=1&ltmpl=default&ltmplcache=2&emr=1");
   driver.findElement(By.id("Email")).clear();
    driver.findElement(By.id("Email")).sendKeys("sidhuconst75");
    driver.findElement(By.id("Passwd")).clear();
    driver.findElement(By.id("Passwd")).sendKeys("waheguru123456");
    driver.findElement(By.id(" PersistentCookie")).click();
    driver.findElement(By.id("signIn")).click();}
```

Above is the output of the above code. After executing several times, the average of execution time is calculated and it computes to 153.866 sec.



Finished after 153.866 seconds

Runs: 1/1 ☒ Errors: 0 ☒ Failures: 0

seleniumt [Runner: JUnit 4] (153.382 s)

```
import java.util.concurrent.TimeUnit;

public class seleniumt{
  private WebDriver driver;
  private String baseUrl;
  private boolean acceptNextAlert = true;
  private StringBuffer verificationErrors = new StringBuffer();

  @Before
  public void setUp() throws Exception {
    driver = new FirefoxDriver();
    baseUrl = "https://accounts.google.com/";
    driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECOND
  }

  @Test
  public void testJava() throws Exception {
    driver.get(baseUrl + "/ServiceLogin?service=mail&passive=true
    //driver.findElement(By.id("account-chooser-add-account")).cl
    driver.findElement(By.id("Email")).clear();
    driver.findElement(By.id("Email")).sendKeys("sidhuconst75");
    driver.findElement(By.id("Passwd")).clear();
    driver.findElement(By.id("Passwd")).sendKeys("waheguru123456"
    driver.findElement(By.id("PersistentCookie")).click();
    driver.findElement(By.id("signIn")).click();
  }

  @After
  public void tearDown() throws Exception {
```

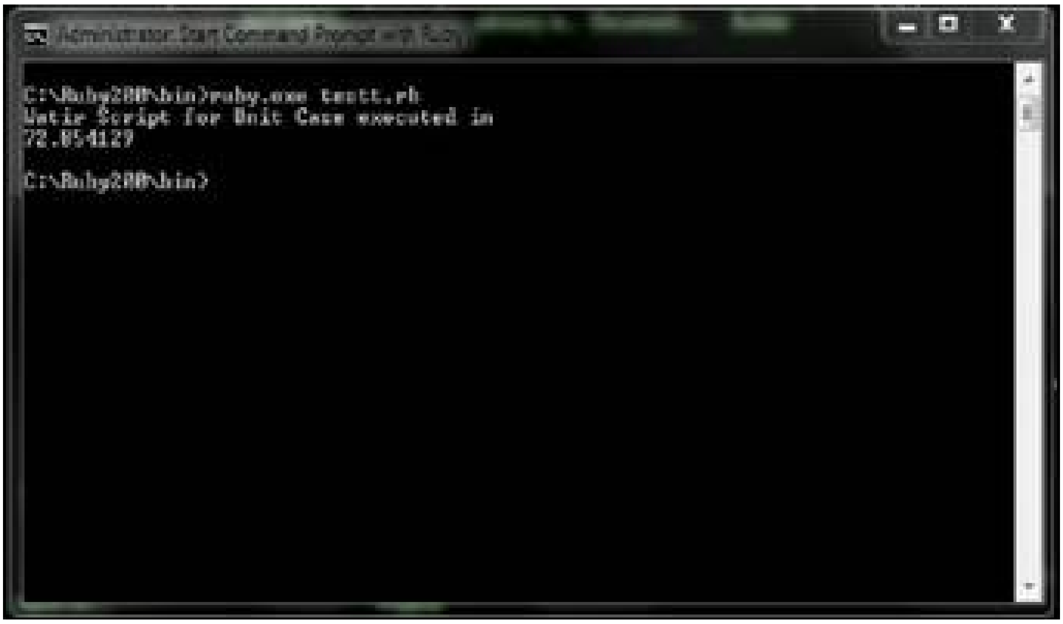
Figure 2 Selenium Test Case Execution

Calculating Execution Time in Watir

Watir is ruby based testing tool and thus require the ruby environment to run the scripts. Watir webdriver is used for executing the testing scripts written in ruby files recorded with Testwise Recorder Sidebar (plugin for firefox). Code snippet used to examine the execution speed is as follow [16].

```
browser = Watir::Browser.new
browser.goto(’http://mail.google.com’)
browser.checkbox(:id, "PersistentCookie").clear
browser.text_field(:id, "Email").set("sidhuconst75")
browser.text_field(:id, "Passwd").set("waheguru123456")
browser.button(:value,"Sign in").click
puts end_time - start_time
```

Above is the output of the above code. After executing several times, the average of execution time is calculated and it computes to 72.85 sec.



```
C:\Ruby200\bin>ruby.exe testt.rb
Watir Script for Unit Case executed in
72.854129

C:\Ruby200\bin>
```

Figure 3 Watir Test Case Execution

Calculating Execution Time in Sikuli

Sikuli being a graphic based testing tool requires the correct screen and region selection so as to perform testing. Java is used to write the Sikuli test case and is compiled with the help of sikuli-java.jar base package. Code snippet used to examine the execution speed is as follow [17].

```
public static void main(String[] args){
try{s.find("img/firefox_ico.png");
s.doubleClick("img/firfox.png");
s.wait("img/address_bar.png");
s.click("img/address_bar.png");
s.type("http://www.gmail.com");
s.click("img/go.png");
long diff= end - start;
System.out.println("Time taken toexecute Sikuli Script is "+ timeSec+" sec");}
catch(FindFailed e){e.printStackTrace();}}}
```

Following is the output of the above code. After executing several times, the average of execution time is calculated and it computes to 23 sec.

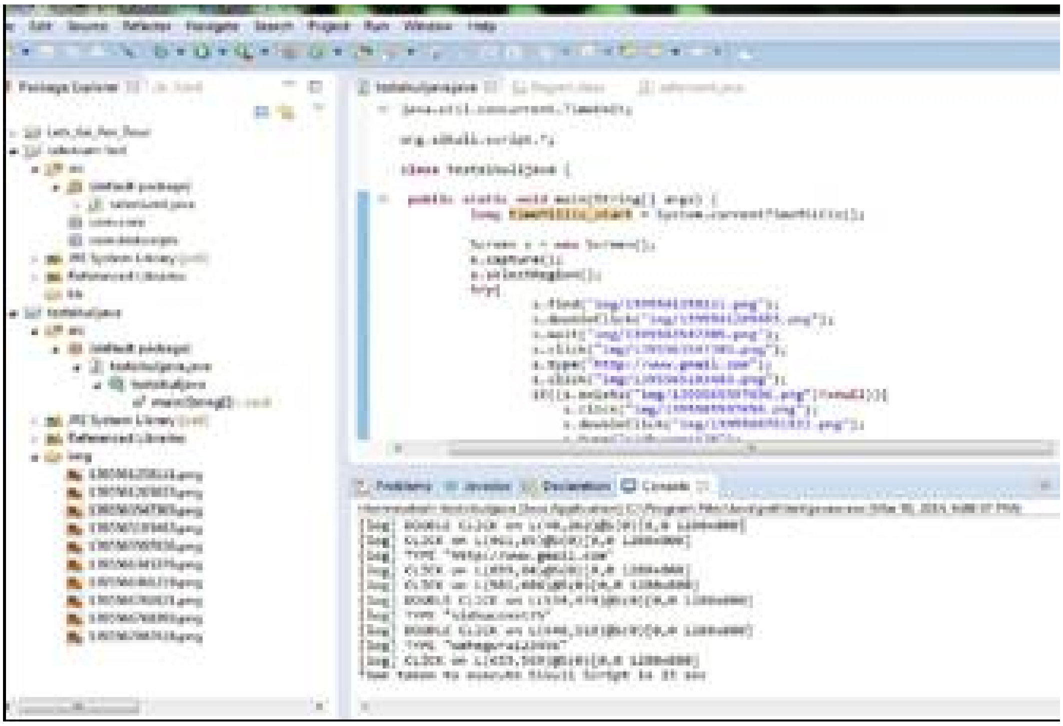


Figure 4 Sikuli Test Case Execution

Recording Capabilities

Recording Capabilities provide an analysis about what wide variety of options are available for Recording Test cases/suites.

Table 1 Recording Capabilities of Selenium, Sikuli and Watir

Metrics	Selenium	Sikuli	Watir	Remarks
Command Insertion	1	1	1	No tool allow this operation to be performed while running
Pause Option	5	1	4	Natively Selenium supports it whereas Watir supports it with 3rd Party Plugin
Ease of Recording	5	2	3	Selenium scores full as it natively support option
Access to Controls	4	2	2	Selenium scores more, whereas Sikuli and Watir scores less due to non-native support

For Selenium the value of parameter calculates = (1+5+5+4)/4= 3.75
For Sikuli the value of parameter calculates = (1+1+2+2)/4= 1.5
For Watir the value of parameter calculates = (1+4+3+2)/4= 2.5

Scripts Generation

In this section capability of the testing tools is checked against the languages and accuracy with which test case/suites are exported.

Table 2 Scripts Generation by Selenium, Sikuli and Watir

Metrics	Selenium	Sikuli	Watir	Remarks
Supported Languages	5	2	1	Selenium scores most as a reason of capability to export code into multiple languages
Accuracy of Code Exported	4	3	3	Selenium code export feature is very good whereas Watir and Sikuli requires extra import lines

For Selenium the value of parameter calculates= (5+4)/2= 4.5
For Sikuli the value of parameter calculates= (2+3)/2= 2.5
For Watir the value of parameter calculates= (1+3)/2= 2

Data Driven Testing

In this section tendency to access of data from external sources and code reusability is checked. Selenium provides access to wide variety of external source [12]. Sikuli provides access to excel, xml and database to fetch data for driven testing but with fewer less methods than selenium [14]. Similarly Watir scores 4 for its access to wide range of external sources [13].

Table 3 Data Driven Testing

Metrics	Selenium	Sikuli	Watir	Remarks
Access of Data from External Sources	5	4	4	Selenium provides wide range of sources for data driven Testing. Watir and Sikuli both provide access to few sources.
Code Reusability on account of Data change	5	5	5	All three testing tools do support code reusability.

For Selenium the value of parameter calculates = (5+5)/2= 5
For Sikuli the value of parameter calculates= (4+5)/2= 4.5
For Watir the value of parameter calculates= (4+5)/2= 4.5

Ease of Learning

Under this section ease of understanding the functioning of testing software is checked for on the basis of three parameters viz. Accessibility, User Interface Features and Manual/Help Provided.

Table 4 Understandability

Metrics	Selenium	Sikuli	Watir	Remarks
Accessibility	5	4	3	Selenium provides greater accessibility. While sikuli and watir scores less
User Interface Feature	4	5	3	Sikuli exhibits best User Interface than selenium and watir
Manual/Help Provided	2	5	2	Best possibly organized help menu is provided in Sikuli

For Selenium the value of parameter calculates= (5+4+2)/3= 3.6
For Sikuli the value of parameter calculates= (4+5+5)/3= 4.6
For Watir the value of parameter calculates= (3+3+2)/3= 2.6

Testing Reports

Testing software is also measured for their productivity in terms of the output produced by them.

This section checks for the same on the basis of three parameters namely Ease of Readability and Understandability of Testing Reports produced, Information about previous runs and Graphical/Command Line nature.

Table 5 Testing Output Produced

Metrics	Selenium	Sikuli	Watir	Remarks
Ease of Readability and Understandability	4	4	3	Selenium and sikuli reports are relatively easy to understand
Information about Previous Runs	4	4	2	Selenium and sikuli provides history on previous runs
Graphical or Command Line	5	5	3	Selenium and sikuli provide output in both views

For Selenium the value of parameter calculates= (4+4+5)/3 = 4.3
For Sikuli the value of parameter calculates= (4+4+5)/3= 4.3
For Watir the value of parameter calculates= (3+2+3)/3= 2.6

Supplement Features

The features that enhance the basic functioning of testing software are evaluated in this section. Two main contributing parameters cost incurred and Nature of 3rd party Add-on/Extension are discussed here.

Table 6 Cost Incurred

Metrics	Selenium	Sikuli	Watir	Remarks
Cost Incurred	5	5	5	All are freely available
Nature of 3rd Party Add-On/Extension	5	2	3	Selenium provides large number of add-ons, watir provides restricted number and sikuli not at all.

Conclusion

Automated Software Testing is one of the useful and important aspect when it comes to testing the applications. Open source technologies are getting popular day by day due to the community support to these open source software. Here in this paper after evaluating the three most popular Open source Softwares viz. Selenium, Sikuli and Watir.

Selenium scores best due to the enhanced recording features, data driven testing, and ease of learning, improved support for 3rd party application integration.

Watir scores second and only lacks of the native support of recording the test cases which is removed by 3rd party applications, less support of languages to which test cases can exported whereas execution time is less than selenium.

Sikuli score least as only graphical nature is not sufficient for the effectiveness of a testing tool. It the support of range of languages to which test cases can be exported, least recording capabilities but it has faster execution speed than selenium and watir. Above is the performance graph which depicts the performance of all of three testing tools on different parameters in congregate manner thus compiling the comparative analysis being done in this paper.

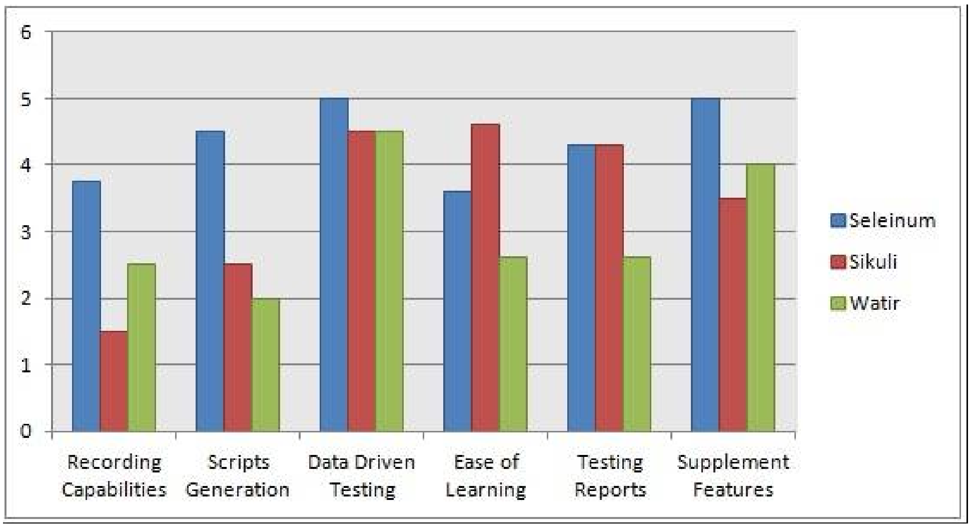


Figure 5 Congregate Peformance Graph of Selenium, Sikuli and Watir on various parameters

References

[1] M. Jovanovi, "Software testing methods and techniques," *IPSI BgD Journals*, vol. 5, pp. 30-41, 2009.

[2] K. M and K. R, "Comparative study of automated testing tools: Testcomplete and quicktest pro," *International Journal of Computer Application*, vol. 24, pp. 1-3, 2011.

[3] "Software Testing Tools List," [Online]. Available: http://www.softwaretestingclass.com/software-testing-tools-list/. [Accessed June 2012].

[4] N. Uppal and V. Chopra, "Design and implementation in selenium ide with web driver," *International Journal of Computer Application*, vol. 46, pp. 8-11, 2012.

[5] "Selenium Documentation," [Online]. Available: http://docs.seleniumhq.org/docs/.

[6] "How Sikuli Works," [Online]. Available: http://doc.sikuli.org/devs/system-design.html.

[7] "Sikuli website," [Online]. Available: http://sikuli.org/.

[8] "Watir Automated testing that doesn't hurt," [Online]. Available: http://watir.com//.

[9] B. Marick, Everyday Scripting with Ruby: For Teams, Testers, and You, The Pragmatic Programmers, 2007, 2007.

[10] B. Pattrichod, "watir-classic 3.6.0.," [Online]. Available: https://rubygems.org/gems/watir-classic.

[11] J. Bakken, "watir-webdriver 0.6.2.," [Online]. Available: https://rubygems.org/gems/watir-webdriver.

[12] M. A. May-Pumphrey, "How to do data driven testing in seleniumide," [Online]. Available: https://groups.google.com/forum/!topic/selenium-users/ZiOQNi31vTE..

[13] Nexle Corporation, "Watir web automated tests," 2012. [Online]. Available: http://www.slideshare.net/nexlesoft/watir-web-automated-tests.

[14] ruoyelj, "Sikuli vs qtp," [Online]. Available: http://tech.liuj.in/?p=62.

[15] Michelangelo van Dam, "UA Testing with Selenium and PHPUnit," May 2013. [Online]. Available: http://www.dragonbe.com/2013/05/ua-testing-with-selenium-and-phpunit.html.

[16] "Examples," [Online]. Available: http://watir.com/examples/.

[17] "How to use Sikuli Script in your JAVA programs," [Online]. Available: http://doc.sikuli.org/faq/030-java-dev.html.

[18] T. Yeh, T. H. Chang and R. C. Miller, "Sikuli: Using gui screenshots for search and automation," [Online]. Available: http://groups.csail.mit.edu/projects/sikuli/.