COMPARISON OF AUTOMATION TESTING TOOLS USING SELENIUM & WATIN

Samudrala Sahithi, P. L. S. Tejaswini, Paladugu Yasaswini, Dr. Sangeetha Yalamanchili

III/IV B.Tech students, IT Department, V. R. Siddhartha Engineering College, Vijayawada,
III/IV B.Tech students, IT Department, V. R. Siddhartha Engineering College, Vijayawada,
III/IV B.Tech students, IT Department, V. R. Siddhartha Engineering College, Vijayawada,
Associate Professor, IT Department, V. R. Siddhartha Engineering College, Vijayawada

ABSTRACT

Manual Testing requires more time or more resources sometimes both Performance testing is impractical in manual testing. It is Less accurate and Executing same tests again and again time taking process as well as Tedious GUI Objects Size difference and Color combinations etc.. are not easy to find in Manual Testing. It is Not Suitable for Large scale projects and time bounded projects Batch Testing is not possible, for each and every test execution Human user interaction is mandatory. Manual Test Case scope is very limited, if it is Automated test, scope is unlimited. Comparing large amount of data is impractical Checking relevance search of operation is difficult Processing change requests during software maintenance takes more time. Automation Testing tools enables developers and testers to easily automate the entire process of testing in software development Automated Software Testing Tools When we start or research for the right automated software testing tool, the major part is the selection of tools from a pool of various categories of tools. The aim of this research paper is to to conduct a comparative study of automated tools automated software testing tools to determine their usability and effectiveness. Software testing tools has major features likes: web testing, window application etc.

Keywords-- SDLC, STLC, automated testing, Selenium tool, WatiN etc...
[1] INTRODUCTION:

Selenium is an open source, robust set of tools that supports rapid development of test automation for web-based applications. Selenium tests can be executed on multiple browser platforms[1,2,3]. Allows scripting in several languages like Java, C#, PHP and python. Selenium can be deployed on Windows, Linux, and Macintosh. Selenium was created by Jason Huggins in 2004 as an internal tool when he was working at Thought works[4,5].

[2] SELENIUM

[2.1] Components:

1. Selenium IDE:
2. Selenium Web driver
3. Selenium Remote Control (RC)
4. Selenium Grid

Selenium IDE: It is a Firefox add-on that you can only use in creating relatively simple test cases and test suites.

Selenium Remote Control: It is also known as Selenium 1, which is the first Selenium tool that allowed users to use programming languages in creating complex tests.

Web Driver: the newer breakthrough that allows your test scripts to communicate directly to the browser, thereby controlling it from the OS level.

Selenium Grid: It is also a tool that is used with Selenium RC to execute parallel tests across different browsers and operating systems.

[2.2] Merit:

1. Selenium is highly cost-effective just because it is an open source tool with pro-grade features.
2. Selenium can be configured easily.
3. Selenium supports a range of browsers like Safari, Chrome, Firefox, and IE and almost every Operating System.
4. One main feature of Selenium is that the language used for building the program is independent of the language that the web application is using. This implies that the test script can be developed in any of the languages that Selenium supports.
5. Selenium is highly flexible and scalable tool for different types and volumes of requirements.

[2.3] Demerits:

1. It supports Web based applications only.
2. Limited support for Image Testing.
3. New features may not work properly.

[2.4] Features:

Combination of tool and DSL – Selenium is an absolute combination of tools and DSL (Domain Specific Language) in order to carry out various types of tests. It allows you to record the tests carried out through the browser. It supports multiple web browsers like Internet Explorer, Safari, Firefox, Chrome, etc.

Uses a rich language for tests – Selenium uses DSL in order to test the web applications.
This language includes 200 commands and is an easy programming language to learn.

A **flexible language** – Once the test cases are prepared, they can be executed on any operating system like Linux, Macintosh, etc.

**Reduce test execution time** – Selenium supports parallel test execution that reduce the time taken in executing parallel tests.

**Lesser resources required** – Selenium requires lesser resources when compared to its competitors like UFT, RFT, etc.

[3] **SELENIUM IDE INSTALLATION STEPS:**

1. Using Firefox, first, download the IDE from the SeleniumHQ downloads page. When downloading from Firefox, you’ll be presented with the Window.
2. Select Install Now. The Firefox Add-ons window pops up, first showing a progress bar, and when the download is complete, displays the Window.
3. Restart Firefox. After Firefox reboots you will find the Selenium-IDE listed under the Firefox Tools menu.

4. To run the Selenium-IDE, simply select it from the Firefox Tools menu. It opens as follows with an empty script-editing window and a menu for loading, or creating new test cases.

**Example:** Testing for [www.google.com](http://www.google.com), and [www.gmail.com](http://www.gmail.com)
COMPARISON OF AUTOMATION TESTING TOOLS USING SELENIUM & WATIN


1) Install the chrome driver

```java
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
public class Openchrome {

    @Test
    public void test12() throws Exception{

        // Initialize browser
        WebDriver driver=new ChromeDriver();

        // Open Google driver.get("http://www.google.com");

        // Close browser driver.close();
    }
}
```

Output

```
java.lang.NullPointerException
            The path to the driver executable must be set by the webdriver.chrome.driver system property; for more information, see http://code.google.com/p/selenium/wiki/O
            at com.google.common.base.Preconditions.checkState(Preconditions.java:197)
            at org.openqa.selenium.remote.service.DriverService.findExecutable(DriverService.java:195)
            at org.openqa.selenium.chrome.ChromeDriverService.createDefaultService(ChromeDriverService.java:99)
            at org.openqa.selenium.chrome.ChromeDriver.<init>(ChromeDriver.java:117)
            at org.openqa.selenium.chrome.ChromeDriverService$1.run(RemoteChromeDriverService.java:169)
            at java.security.AccessController.doPrivileged(AccessController.java:959)
            at org.openqa.selenium.chrome.ChromeDriver.<init>(ChromeDriver.java:111)
            at com.google.webdriver.remote.service.RemoteDriverService.<init>(RemoteDriverService.java:109)
            at com.google.webdriver.ChromeDriver.<init>(ChromeDriver.java:100)
            at com.google.webdriver.ChromeRemoteService.initChromeDriver(ChromeRemoteService.java:54)
            at org.openqa.selenium.remote.RemoteWebDriver.<init>(RemoteWebDriver.java:138)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:52)
            at org.openqa.selenium.remote.RemoteWebDriver.<init>(RemoteWebDriver.java:130)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:18)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:14)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:10)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:4)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:2)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1)
            at com.google.webdriver.ChromeRemoteService.<init>(ChromeRemoteService.java:1

```

We need to set variable.

Samudrala Sahithi, P. L. S. Tejaswini, Paladugu Yasaswini, Dr. Sangeetha Yalamanchili
[4.1] Gmail:
package login;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Login1 {
    public static void main(String[] args) {
        // Create a new instance of the Firefox driver
        WebDriver driver = new FirefoxDriver();
        // Wait For Page To Load
        // Put an Implicit wait, this means that any search for elements on the page could take the time the implicit
        wait is set for before throwing exception driver.manage().timeouts().implicitlyWait(10,
        TimeUnit.SECONDS);
        // Navigate to URL
        driver.get("https://mail.google.com/");
        // Maximize the window. driver.manage().window().maximize();
        // Enter UserName
        driver.findElement(By.id("Email")).sendKeys(" YOUR USER NAME");
        // Enter Password driver.findElement(By.id("Passwd")).sendKeys("YOUR PASSWORD");
        // Wait For Page To Load driver.manage().timeouts().implicitlyWait(60, TimeUnit.SECONDS);
        // Click on 'Sign In' button driver.findElement(By.id("signIn")).click();
        // Click on Compose Mail. driver.findElement(By.xpath("//div[@class='z0']/div")).click();
        // Click on the image icon present in the top right navigational Bar
        driver.findElement(By.xpath("//div[@class='gb_1 gb_3a gb_nc gb_e']/div/a")).click();
        // Click on 'Logout' Button driver.findElement(By.xpath("//*[@id='gb_71']")).click();
        // Close the browser. driver.close();
    }
}

[5] WATIN:

WatiN stands for Web Application Testing In .NET, and it's a framework that enables web application
testing through Internet Explorer. WatiN is inspired on WatiR, a web application testing framework for
Ruby. The way that WatiN works is very "easy", knowing that there's a lot of work behind a framework
like this. WatiN lets you open Internet Explorer instances, and through interop, interact with the
elements in a form. With WatiN, you can get and set values from the elements in a form, and you can
fire events of any of the elements in the document too.
[6] INSTALLATION OF WATIN:

**Step 1:** Install the visual studio either 2010 or 2015 model.

![Visual Studio Installation](image1)

**Step 2:** Open visual studio and start the new project as shown below.

![Visual Studio Project](image2)

**Step 3:** In order to perform the WatiN Testing download and add the Library files of WatiN Shown as below

![WatiN Library Addition](image3)

**Step 4:** Now You are successfully can work with the WatiN automation Tool.

[7] WATIN USAGE:

Now that we have a brief idea of what WatiN is and what we can do with it, let's write some code to
see the API's usage. The below code example opens an Internet Explorer instance and points it to www.google.com, then, when the page is loaded, it searches a textbox with name "q" and fills it with the text "WatiN". Then, we do the same we did for the textbox but, this time, looking for a button whose value is "Google Search". In this case, we perform the "Click()" event of the button.

**CODE:**

going System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using WatiN.core;
using WatiN.Core.Constraints;
namespace ClassLibrary2
{
    public class Class1
    {
        [STAThread]
        static void Main(string[] args)
        {
            Class1 ie = new Class1();
            ie.GoTo("http://www.google.com");
            ie.TextField(FindByName("q")).TypeText("WatiN");
            ie.Button(FindByValue("Google Search")).Click();
        }
    private object Button(Constraint constraint)
    {
        throw new NotImplementedException();
    }
    private object TextField(object p)
    {
        throw new NotImplementedException();
    }
    private void GoTo(string v)
    {
        throw new NotImplementedException();
    }
}
[8] RESULT:

[9] CONCLUSION:-

The Selenium tool is the automation tool of testing for the web applications. It requires less time as compared to the manual testing, manual testing takes large time to write the testcases manually and is so time taking. It is compatible with a wide variety of browsers and offers great flexibility to write text scripts as well. Furthermore, Selenium provides robust support to a variety of programming languages, including C#, Java, Python and Ruby. Even just a basic knowledge of any of this language is enough to work with Selenium. Therefore, Selenium tool is more efficient than WatiN testing tool.
REFERENCES


