

Name: Umair Khan

Paper: Software verification and validation

ID: 6949

Selenium

Introduction to selenium:

Selenium is a suite of open source software testing automation tools that's become the product in the quality assurance world. With a list of several programming languages, all main operating systems and browsers supported, Selenium is currently used in production in such companies as Netflix, Google, Hub Spot, Fitbit, and more. The whole suite provides a range of solutions for different testing problems and needs. Further in the article, we'll talk about these testing tools, and the reasons Selenium stayed relevant after ten years since its creation. But for now, let's examine what it is about Selenium that made it such a big deal.

Background:

Jason Huggins' 2004 creation was a JavaScript framework aimed at freeing its creator from repetitive manual testing. The product that was first predictably named JavaScriptTestRunner could execute tests directly in a browser, drive interactions on the page, and rerun them without manual input. This [JavaScript tool](#) took off after Huggins realized its potential, made it open source and renamed it Selenium Remote Control. The innovative part of it was that no other tool before it allowed testers to talk to the browser in their programming language of choice.

However, it soon became apparent that browsers applied security limitations on JavaScript, making it impossible to use the tool's full capabilities. At that time Google was an avid Selenium user, but engineers struggled around the limitations. One of them, Simon Stewart, started to work on a product that would speak to browsers natively, which he called WebDriver. Selenium combined forces with WebDriver and changed the way we've been testing software for over a decade.

Selenium Infrastructure:

Selenium 1:

The first product in the Selenium lineup was Selenium Remote Control (now known as Selenium 1). Due to its aforementioned limitations and the following merge with WebDriver, it was soon deprecated and is no longer supported.

Selenium WebDriver (Selenium 2):

Most of the time, when people talk about Selenium, they mean Selenium WebDriver. The largest chunk of the product's development is focused on this product.

Selenium IDE:

This Firefox add-on is dismissed by half of the testers and celebrated by the other half. Not meant for production, it's easy to learn and perfect for prototyping tasks. You don't need programming skills to successfully operate in Selenium IDE as it basically records your actions in the browser and repeats them.

Selenium Grid:

Selenium Grid allows you to run parallel tests on multiple machines and browsers at the same time. The main function of this tool is to save time.

Part a)

Pros and Cons of selenium:

Selenium Grid

PROS	CONS
Supports simultaneous of test cases in multiple environments and browsers.	The code is executed only on the local machines where test cases are launched.
Captures the screenshots of browsers at different stages of test case execution.	The remote machines only receive the browser control commands.
Provides tools needed to diagnose the failures and rebuild a similar environment for new test execution.	Large quantities of node servers result in performance depreciation.
Selenium Grid saves time extremely as it uses Hub-Node design.	Considerable efforts and time are required for initial operation of parallel testing.

Selenium IDE

PROS	CONS
Its simple, easy install and use.	Only available for Firefox.
No prior programming knowledge & needed, though basic knowledge of DOM and HTML is required.	Designed to create test prototypes.
Test cases can be exported to usable formats in Selenium WebDriver and RC.	Conditional operations and iterations are not supported.
Extensions are supported.	Execution of test cases is slow compared to WebDriver and RC.
Built-in test results reporting and help modules.	Does not test dynamic web applications.
Supports appropriate scripting, reporting and debugging besides recording and replaying.	Data-driven testing is not supported.
Can export recorded tests in different programming languages like Java, Python, Ruby etc.	Programming logic is not used in scripting test cases.

Selenium RC

PROS	CONS
Supports cross-browser testing.	Complicated installation than IDE.
Conditional operations and iterations are supported.	Prior programming knowledge is needed.
Supports data-driven testing.	Selenium RC must be needed to run test cases.
Complete and matured API.	API contains confusing and redundant commands.
New browsers are readily supported.	Browser interaction is less realistic.
Execution speed is more than IDE.	Slower execution speed than WebDriver.
Supports user preferred languages.	Uses Java Script and results are inconsistent.

Selenium WebDriver

PROS	CONS
Easy installation compared to Selenium RC.	Complicated installation than Selenium IDE.
Direct browser communication.	Prior programming knowledge is required.
More realistic browser interaction.	No immediate support for new browsers.
No separate component such as RC server is needed.	No mechanism to track runtime messages.
Execution time is faster than RC and IDE.	Image testing is not possible.
Supports testing on different platforms like Mac, Linux, Windows, iOS, and Android.	No detailed test results and reports.

Part b)

Selenium Functionality:

- Selenium is an open source and portable Web testing Framework.
- Selenium IDE provides a playback and record feature for authoring tests without the need to learn a test scripting language.
- It can be considered as the leading cloud-based testing platform which helps testers to record their actions and export them as a reusable script with a simple-to-understand and easy-to-use interface.
- Selenium supports various operating systems, browsers and programming languages. Following is the list:
 - Programming Languages: C#, Java, Python, PHP, Ruby, Perl, and JavaScript
 - Operating Systems: Android, iOS, Windows, Linux, Mac, Solaris.
 - Browsers: Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari, etc.
- It also supports parallel test execution which reduces time and increases the efficiency of tests.
- Selenium can be integrated with frameworks like Ant and Maven for source code compilation.
- Selenium can also be integrated with testing frameworks like Test NG for application testing and generating reports.
- Selenium requires fewer resources as compared to other automation test tools.
- WebDriver API has been indulged in selenium which is one of the most important modifications done to selenium.
- Selenium web driver does not require server installation, test scripts interact directly with the browser.
- Selenium commands are categorized in terms of different classes which make it easier to understand and implement.
- Selenium Remote Control (RC) in conjunction with WebDriver API is known as Selenium 2.0. This version was built to support the vibrant web pages and Ajax.

Part C)

Supported Languages:

The flexibility that Selenium provides is almost unmatched in the test automation world. First, we have ten supported languages, among which are Java, Ruby, C#, PHP, JavaScript, Perl, and R – all among today's [most used programming languages](#). It's also the only mainstream tool that covers Linux testing.

SELENIUM COMPATIBILITY		
Languages	Platforms	Browsers
Java C# PHP Ruby Perl Python JavaScript Objective-C Haskell R	Windows Linux Mac Android (with Selendroid, Appium, or Robotium) iOS (with ios-driver or Appium)	Google Chrome Internet Explorer 7-11 Firefox Safari Opera HtmlUnit phantomjs

Part D)

Supporting tests:

With Selenium, you can also test [native, hybrid, or web mobile apps](#), though you'll need additional software. There are two main options – [Appium](#) and [Selendroid](#). They are both based on Selenium so that developers already versed in it can apply the same principles when testing mobile apps.

Both tools are open source and have great community support. The main difference is that Appium supports iOS, Android, and Windows devices, while Selendroid focused exclusively on Android. Selendroid also bundles with Appium so that when you test for Android versions 2.3 through 4.3, the program will automatically switch to Selendroid.

It may seem that using Appium is a more logical choice as it doesn't limit you to one OS. However, it makes more sense to just use Selendroid if you're planning to test Android exclusively. It also has a few nifty features such as "hot plugging" – you can plug in devices and unplug them without interrupting the tests, increasing the number of emulators or hardware devices that can be tested at the same time.

There are two more niche tools you might want to know about as well:

- [Robotium](#) – a black-box testing framework for Android
- [iOS-driver](#) – a Selenium WebDriver API for iOS testing integrated with Selenium Grid

Part E)

Code:

```
Import org.openqa.selenium.WebDriver;
Import org.openqa.selenium.firefox.FirefoxDriver
//import org.openqa.selenium.chrome.ChromeDriver;
Public class check {

    Public static void main(String [] args) {
        // declaration and instantiation of objects/variables
        System.setProperty ("webdriver.gecko.driver","C:\\\\geckodriver.exe");
        WebDriver driver = new FirefoxDriver ();

        String baseUrl = "http://demo.guru99.com/test/newtours/";
        String expectedTitle = "Welcome: Mercury Tours";
        String actualTitle = "";

        // launch Fire fox and direct it to the Base URL
        driver.get(baseUrl);

        // get the actual value of the title
        actualTitle = driver.getTitle ();

        If (actualTitle.contentEquals(expectedTitle)){
            System.out.println("Test Passed!");
        } else {
            System.out.println ("Test Failed");
        }

        //close Fire fox
        Driver.Close ();

    }
}
```