|  |
| --- |
|  |
| Selenium Tutorial |
|  |
|  |
| **Vignesh Ayyyamani** |
|  |

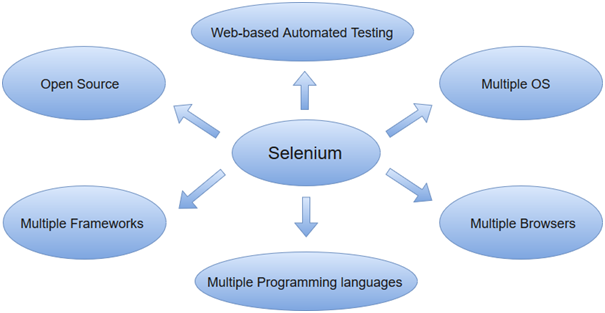
|  |
| --- |
|  |

What is Selenium

Selenium is one of the most widely used open source Web UI (User Interface) automation testing suite. It was originally developed by Jason Huggins in 2004 as an internal tool at Thought Works. Selenium supports automation across different browsers, platforms and programming languages.

Selenium can be easily deployed on platforms such as Windows, Linux, Solaris and Macintosh. Moreover, it supports OS (Operating System) for mobile applications like iOS, windows mobile and android.

Selenium supports a variety of programming languages through the use of drivers specific to each language.Languages supported by Selenium include C#, Java, Perl, PHP, Python and Ruby.Currently, Selenium Web driver is most popular with Java and C#. Selenium test scripts can be coded in any of the supported programming languages and can be run directly in most modern web browsers. Browsers supported by Selenium include Internet Explorer, Mozilla Firefox, Google Chrome and Safari.



Selenium can be used to automate functional tests and can be integrated with automation test tools such as **Maven**, **Jenkins**, **& Docker** to achieve continuous testing. It can also be integrated with tools such as **TestNG**, & **JUnit** for managing test cases and generating reports.

Automation Testing

Automation testing uses the specialized tools to automate the execution of manually designed test cases without any human intervention. Automation testing tools can access the test data, controls the execution of tests and compares the actual result against the expected result. Consequently, generating detailed test reports of the system under test.

Automation testing covers both functional and performance test on an application.

* Functional automation is used for automation of functional test cases. For example, regression tests, which are repetitive in nature, are automated.
* Performance automation is used for automation of non-functional performance test cases. For example, measuring the response time of the application under considerable (say 100 users) load.

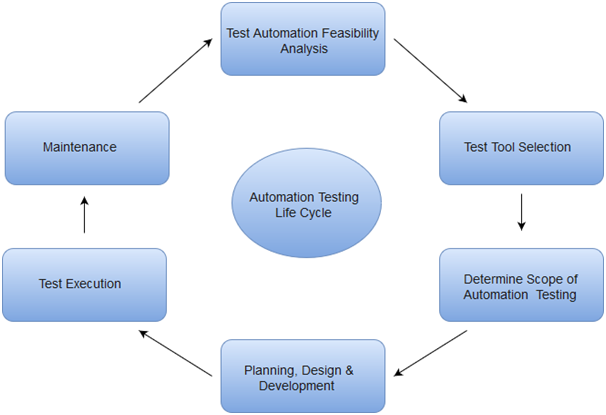
Automation Testing tools which are used for functional automation:

* Quick Test Professional, provided by HP.
* Rational Robot, provided by IBM.
* Coded UI, provided by Microsoft.
* Selenium, open source.
* Auto It, open Source.

Automation Testing tools which are used for non-functional automation:

* Load Runner, provided by HP.
* JMeter, provided by Apache.
* Burp Suite, provided by PortSwigger.
* Acunetix, provided by Acunetix.

Automation Testing Life Cycle

Why Automated Testing

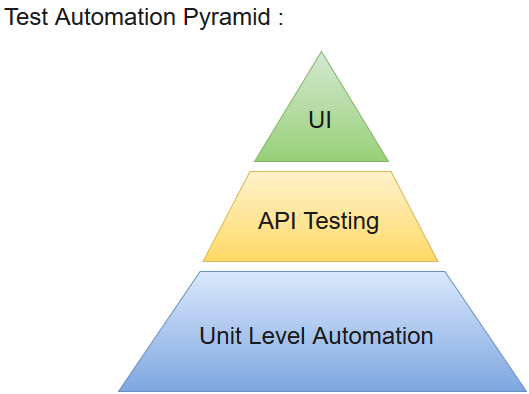
Automation testing has specific advantages for improving long-term efficiency of any software. The key benefits of test automation are:

* Automated testing has long been considered beneficial for big software organizations. Although, it is often thought to be too expensive or difficult for smaller companies to implement.
* Automated testing tools can be programmed to build and execute test scripts at a specific time without involving any human intervention.For instance, automated test can be automatically kicked off overnight, and the testers can analyse the results of the automated the next morning.
* Automated testing tools are able to playback pre-recorded and pre-defined actions.
* Automation testing supports frequent regression testing.
* It provides rapid feedback to developers.
* It provides unlimited iterations of test case execution.
* It provides disciplined documentation of test cases.
* Automated test generates customized defect reports.
* Less error prone as compared to manual testing.

Test Automation for Web Applications

If we take a look at the type of software applications prevailing in current market scenario, most of the software applications are written as web-based applications to be run in an internet browser. The testing strategy for web-based applications varies widely among companies and organizations.In an era of highly interactive and responsive software processes where many organizations are using some form of agile methodology, test automation is frequently becoming a requirement for software projects.

The most effective manner to carry out test automation for web application is to adopt a pyramid testing strategy.This pyramid testing strategy includes automation tests at three different levels. Unit testing represents the base and biggest percentage of this test automation pyramid. Next comes, service layer, or API testing. And finally, GUI tests sit at the top. The pyramid looks something like this:



|  |
| --- |
| Comparison between manual and Automation Testing |

## automation testing

|  |  |
| --- | --- |
| Time Consuming and Tedious | Comparatively Fast Execution |
| Massive Investment in Human resources | Less Investment in Human Resources |
| Not Accurate all the time | Accurate all the time |
| Not ideal for regression testing | Ideal for regression testing |
| Once or twice execution | Frequent execution |
| Hard to manage results | Easy to manage results |
| No Programming Language | Some programming language is a must |

### Why is Automation used?

* To find defects which are missed in manual testing.
* To ease and reduce the time of execution of regression testing.
* To find quick bugs
* To support the Agile or extreme development process.
* To ease customized reporting.

## Advantages of Automation Testing

To summarise, whatever we discussed above, let us see the main benefits of Automation:

* Cost Reduction
* Reusability
* Faster Execution
* Unattended Execution
* Reliability
* Quality software.

## Disadvantages of Automation Testing

Let us now see the disadvantages of using Automation:

* Automation is not easy; you need to be good at any programming language.
* If something goes wrong in your automation scripts, debugging is not an easy task.
* Complexity in automating complex scenarios.
* Not everything can be automated.

Selenium Features

* 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 TestNG 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.

Limitations

* Selenium does not support automation testing for desktop applications.
* Selenium requires high skill sets in order to automate tests more effectively.
* Since Selenium is open source software, you have to rely on community forums to get your technical issues resolved.
* We cant perform automation tests on web services like SOAP or REST using Selenium.
* We should know at least one of the supported programming languages to create tests scripts in Selenium WebDriver.
* It does not have built-in Object Repository like UTF/QTP to maintain objects/elements in centralized location. However, we can overcome this limitation using Page Object Model.
* Selenium does not have any inbuilt reporting capability; you have to rely on plug-ins like **JUnit** and **TestNG** for test reports.
* It is not possible to perform testing on images. We need to integrate Selenium with **Sikuli** for image based testing.
* Creating test environment in Selenium takes more time as compared to vendor tools like UFT, RFT, Silk test, etc.
* No one is responsible for new features usage; they may or may not work properly.
* Selenium does not provide any test tool integration for Test Management.

**A Comparison between Selenium and QTP**

**Advantages of Selenium over QTP**

|  |  |
| --- | --- |
| **Selenium** | **QTP/UFT** |
| **Open source**, **free to use**, and **free of charge.** | **Commercial**. |
| **Highly extensible** | Limited add-ons |
| Can run tests across **different browsers** | Can only run tests in **Firefox** , **Internet Explorer** and **Chrome** |
| Supports **various operating systems** | Can only be used in **Windows** |
| Supports **mobile devices** | Supports mobile devise using 3rd party software |
| Can execute tests **while** the **browser is minimized** | Needs to have the application under test to be visible on the desktop |
| Can execute tests **in parallel**. | Can only execute in parallel but using Quality Center which is again a paid product. |

**Advantages of QTP over Selenium**

|  |  |
| --- | --- |
| **QTP** | **Selenium** |
| Can test **both web and desktop applications** | Can only test web applications |
| Comes with a **built-in object repository** | Has no built-in object repository |
| **Automates faster than Selenium**because it is a fully featured IDE. | Automates at a slower rate because it does not have a native IDE and only third party IDE can be used for development |
| Data-driven testing is easier to perform because **it has built-in global and local data tables**. | Data-driven testing is more cumbersome since you have to rely on the programming language’s capabilities for setting values for your test data |
| **Can access controls within the browser**(such as the Favorites bar, Address bar, Back and Forward buttons, etc.) | Cannot access elements outside of the web application under test |
| Provides professional **customer support** | No official user support is being offered. |
| Has native capability to **export test data** into external formats | Has no native capability to export runtime data onto external formats |
| Parameterization Support is in built | Parameterization can be done via programming but is difficult to implement. |
| Test Reports are generated automatically | No native support to generate test /bug reports. |

Though clearly, QTP has more advanced capabilities, Selenium outweighs QTP in three main areas:

* **Cost**(because Selenium is completely free)
* **Flexibility**(because of a number of programming languages, browsers, and platforms it can support)
* **Parallel testing**(something that QTP is capable of but only with use of Quality Centre)