Module- 5 (**Load Runner Up and Selenium IDE)**

Which components have you used in Load Runner

Load Runner is a software testing tool from [Micro Focus](https://en.wikipedia.org/wiki/Micro_Focus). It is used to test [applications](https://en.wikipedia.org/wiki/Application_software), measuring system behavior and performance under load. Load Runner can simulate thousands of users concurrently using application software, recording and later analyzing the performance of key components of the application.

Load Runner simulates user activity by generating messages between application components or by simulating interactions with the user interface such as key presses or mouse movements. The messages and interactions to be generated are stored in scripts. Load Runner can generate the scripts by recording them, such as logging [HTTP](https://en.wikipedia.org/wiki/HTTP) requests between a client web browser and an application's web server.[[1]](https://en.wikipedia.org/wiki/LoadRunner#cite_note-1)

[Hewlett Packard Enterprise](https://en.wikipedia.org/wiki/Hewlett_Packard_Enterprise) acquired Load Runner as part of its acquisition of [Mercury Interactive](https://en.wikipedia.org/wiki/Mercury_Interactive) in November 2006.[[2]](https://en.wikipedia.org/wiki/LoadRunner#cite_note-2)[[3]](https://en.wikipedia.org/wiki/LoadRunner#cite_note-3) In Sept 2016, Hewlett Packard Enterprise announced it is selling its software business, including Mercury products, to Micro Focus.[[4]](https://en.wikipedia.org/wiki/LoadRunner#cite_note-4) As of 01-Sept-2017, the acquisition was complete.[[5]](https://en.wikipedia.org/wiki/LoadRunner#cite_note-5)

On Dec 12, 2019, Micro Focus announced newer names for Load Runner package and started following Culver. [[6]](https://en.wikipedia.org/wiki/LoadRunner#cite_note-6)

Load Runner is now Load Runner Professional 2020

Performance Center is now Load Runner Enterprise 2020

Storm Runner Load is now Load Runner Cloud 2020

The key components of Load Runner are:

Load Generator generates the load against the application by following scripts

Vu Gen (Virtual User Generator) for generating and editing scripts

Controller controls, launches and sequences instances of Load Generator - specifying which script to use, for how long etc. During runs the Controller receives real-time monitoring data and displays status.

Agent process manages connection between Controller and Load Generator instances.

Analysis assembles logs from various load generators and formats reports for visualization of run result data and monitoring data.

Modules are available to enable Load Runner to capture, replay and script different application and networking technologies. These include support for:[[7]](https://en.wikipedia.org/wiki/LoadRunner" \l "cite_note-7)

applications using Microsoft [.NET](https://en.wikipedia.org/wiki/.NET_Framework) and [Java](https://en.wikipedia.org/wiki/Java_(programming_language))

database servers such as [Microsoft SQL Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server) and [Oracle](https://en.wikipedia.org/wiki/Oracle_Database)

internetworking protocols such as [DNS](https://en.wikipedia.org/wiki/Domain_Name_System), [FTP](https://en.wikipedia.org/wiki/File_Transfer_Protocol) and [LDAP](https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol)

e-mail protocols including [IMAP](https://en.wikipedia.org/wiki/Internet_Message_Access_Protocol), [MAPI](https://en.wikipedia.org/wiki/Messaging_Application_Programming_Interface), [POP3](https://en.wikipedia.org/wiki/Post_Office_Protocol) & [SMTP](https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol)

Remote client technologies such as [Citrix ICA](https://en.wikipedia.org/wiki/Independent_Computing_Architecture) and [RDP](https://en.wikipedia.org/wiki/Remote_Desktop_Protocol).

Load Runner can be run standalone or multiple instances can pooled for use by several people under the control of Load Runner Enterprise, formerly known as [HP Performance Center](https://en.wikipedia.org/wiki/HP_Application_Lifecycle_Management#HP_Performance_Center).

How can you set the number of Vusers in Load Runner

Problem:

How to get the active VUser count at any point in time in Load Runner?

Explanation:

Load Runner executes the script in the controller or Performance Center and generates the required number of Virtual Users (VUsers). But all the VUsers are not active full time especially when login and logout part is in the Action(). If you want to know how many VUsers are active at a particular time then follow below steps.

Solution:

Prerequisite:

As a prerequisite, you need to first identify how many TCP/IP connections are established by one VUser during the test. For this purpose:

Comment out the “Logout” transaction of the VUGen script

Run the test for a few minutes. The test duration should be long enough to complete at least one iteration

Collate the result

Open the result file in Load Runner Analysis Tool

Open the “Connections” graph  
Graphs->Add New Item->Add New Graph->Web Resources->Connections

Note down the number of connections established during the one user test (No. of Connections = 6; Refer the below screenshot)

You can set the number of Vusers in the controller section while creating your scenarios. Many other advanced options like ramp-up, ramp-down of Vusers are also available in the Controller section.

Learn more about [Controller & Scenario creation](https://www.guru99.com/how-to-use-controller-in-loadrunner.html)

What is Correlation

Correlation is used to obtain data which is unique for each run of your test script (ex: session ids). While recording, these dynamic values are hard-coded in your script causing the script to fail during playback. Correlation is a technique where dynamic values are not hard-coded in your script but are extracted at run-time to avoid failure.

Correlation is done for the dynamic value or the value returned by server for any request.

Parameterization differs from correlation in a way that former takes care of user input data whereas later takes care of data returned by server.

Manual correlation and automated correlation follow the same steps.

In Manual Correlation, we have to identify the dynamic value and capture it from the response of previous request.  Replace dynamic value with parameter name manually everywhere in the script.

Automated Correlation works with existing rules.

WDiff is used to identify the dynamic value. With WDiff compare two scripts with identical steps and user input.

WDiff does line by line comparison.  Another tool available can also be used for word by word comparison.

Correlation function web\_reg\_save\_param is used for capturing the value for correlation. The other versions of correlation are web\_reg\_save\_param\_ex and web\_reg\_save\_param\_regexp.

The mandatory attributes of correlation function web\_reg\_save\_param are parameter name, left boundary (LB) and right boundary (RB).

Correlation is not only done for dynamic values which change every time but also for data returned by server for different users. To identify such data record, use two scripts with different users (login credentials) keeping user input and steps same. Compare these scripts either with WDiff or any text comparison tool.

What is the process for developing a Vuser Script

There are 5 steps for developing a vuser script.  
  
1-recording the vuser script .  
  
2-edit the vuser script.  
  
3-runtime setting .  
  
4-run the vuser script in stand-alone mode.  
  
5-incorporate the vuser script into a load runner scenario.

How Load Runner interacts with the application

Protocol is used in Load Runner to interact with the application.

What is Load Runner?

Load Runner is a Performance Testing tool which was pioneered by Mercury in 1999. Load Runner was later acquired by HPE in 2006. In 2016, Load Runner was acquired by Micro Focus.

Load Runner supports various development tools, technologies and communication protocols. In fact, this is the only tool in market which supports such a large number of protocols to conduct [Performance Testing](https://www.guru99.com/performance-testing.html). Performance Test Results produced by Load Runner software are used as a benchmark

How many VUsers are required for load testing

One of the most common questions we get is "how many virtual users do I need to simulate?" The obvious answer is "how many people do you expect to have accessing the your system simultaneously?". If you have this information, then you're done!

But a tester or project manager frequently has less specific information available, such as stats from a web log analyzer. The calculators on this page will help determine how many virtual users are required to accurately simulate the expected load on a website.

Performance testers, depend on the business teams to give the idea about the concurrent user load required for the application to be tested. If only, life would be that easy. In most scenarios, the information regarding the number of concurrent users (and the steady state time for those) is not available with the business teams. Translating the information of real time user load into simulated load is tricky. Also the performance test environments are scaled down as compared to that with the production. In such situations, the business expects the performance test team to come up with the information regarding the number of users to test and the duration of the test and here comes an article that can help you with that.

Let’s assume some basic items while simplifying this task –

An application can have multiple modules (or functionalities) and each of those can be realized using Agile Load scripts.

While testing the app the virtual users perform all the transactions as are available in the script to complete all the steps required for the module.

Not all virtual users will execute each of the functionalities

Scripts are recorded in such a fashion that while replaying, they will send a request to the server and response received would be read by the controller.

Think time is added into the scripts to emulate the pauses (or time that he thinks for) that a user has when he moves from one page to another. Change in think time can have a direct impact on the number of transactions load (and thus the load) on the application.

Post test execution counters like hits / sec, number of users / unit time or transactions / sec can give an idea about the load subjected on the application.

One request made by a user can be considered as one Agile Load transaction.

Let’s look at the formula that we will use to calculate the number of users –

Number of virtual users = Length of user scenario (in sec) \* Number of requests per second required

To further understand, let’s consider an example. -> Peak load on an application is 10 users per hour and each user on an average spends 10 mines on the website and goes through 10 web pages. This means the implemented user scenario will contains 10 page-requests and each virtual user will run for 10 minutes.

The above formula in such cases gets modified to:

Number of virtual users = Number of users per hour \* Number of requests per user \* 3600 / [Length of user scenario (in sec)]

Here:

Number of users per hour = 10

Number of requests per user = 10 (number of web pages he accesses)

Length of user scenario = 10 min \* 60 = 600 sec

Hence using above formula:

Number of virtual users = 10 \* 10 \* 3600 / 600 = 600

Hence the above formula gives a user count of 600 users. However a noticeable point is in an hour a user runs for only 10 min s and is inactive for most of the time. Moreover tool licensing are planned in such a way that more the number of concurrent users you require, more is the cost of license. Thus we need a better way of calculating the required load in such a way that the production load is met without increasing the cost of the tool (and thus the test). Instead of keeping the users idle for 50 mins out of 1 hour of the test, if same user is made to perform the task of multiple users and thus create load without impacting the load, the concurrent vuser load required can be substantially reduced.

Eg. In above scenario, if after 10 mins, a virtual user is made to make multiple iterations and thus make 6 iterations in an hour the number of concurrent users required (for the performance test) can be substantially reduced to 100 vusers.

This user load can be further reduced by reducing the think time and thus increasing the transaction load (per unit time). However it is not the best approach since it doesn’t help to create the accurate production like scenario and is coupled with multiple issues. Here is why this reducing think time approach should not be followed -(See our article  [Number of Virtual Users – An unavoidable parameter while designing performance tests](http://www.agileload.com/agileload/blog/2012/12/14/number-of-virtual-users-an-unavoidable-parameter-while-designing-performance-tests) )

What is the relationship between Response Time and Throughput

Response time and throughput are related. The response time for an average transaction tends to decrease as you increase overall throughput.

However, you can decrease the response time for a specific query, at the expense of overall throughput, by allocating a disproportionate amount of resources to that query. Conversely, you can maintain overall throughput by restricting the resources that the database allocates to a large query.

The trade-off between throughput and response time becomes evident when you try to balance the ongoing need for high transaction throughput with an immediate need to perform a large decision-support query. The more resources that you apply to the query, the fewer you have available to process transactions, and the larger the impact your query can have on transaction throughput. Conversely, the fewer resources you allow the query, the longer the query takes.

The Throughput graph shows the amount of data in bytes that the Vusers received from the server in a second. When we compare this with the transaction response time, we will notice that as throughput decreased, the response time also decreased. Similarly, the peak throughput and highest response time would occur approximately at the same time.

What is the difference between hits/second and requests/second

Hits per second represents the number of requests sent to the server in one second (the load which the server is being hit).

Hits per second is the total load set by the concurrent virtual users on the server, no matter if they are executed successfully or not on the server side.

The number of requests executed successfully by the server per unit of time is called throughput. Learn about [what is throughput in performance testing](https://loadfocus.com/blog/2013/07/what-is-throughput-in-performance-testing/).

To find how the hits per second increase while executing your performance test, you can check the charts in the [Load Testing Service](https://loadfocus.com/load-testing) by Load Focus.

First let's see what Hits per second & Throughput means.  
  
Hits per second - Number of request hits the web server per second.  
  
Throughput - Amount of data in bytes (Response) that the Vusers received from the server in a second. Measured in bytes/second.  
  
In Analyzer we have an option to merge the graphs, using that we can merge the Hits per Second & Throughput graph to understand how the application is performing.  
  
Below are the possible scenarios.  
  
1. When Throughput graph is plotted higher than the Hits per second graph then  the application is performing great, which means the servers are processing the requests at a rate greater than the requests are reaching the server.  
  
2. When both the graphs are parallel or overlapping with each other then the application is performing good, which means the servers are processing the requests at a same rate the requests are reaching the server. (See the below Image).  
  
3. When Throughput graph is plotted lower than the Hits per second graph then the application is having issues, which means the servers are processing the requests at a rate lower than the requests are reaching the server.

What is Automation Testing

Automation testing is the process of testing software and other tech products to ensure it meets strict requirements. Essentially, it’s a test to double-check that the equipment or software does exactly what it was designed to do. It tests for bugs, defects, and any other issues that can arise with product development.

Although some types of testing, such as regression or functional testing can be done manually, there are greater benefits of doing it automatically. Automation testing can be run at any time of the day. It uses scripted sequences to examine the software. It then reports on what’s been found, and this information can be compared with earlier test runs. Automation developers generally write in the following programming languages: C#, JavaScript, and Ruby.

Many software businesses will have an appointed [QA (quality assurance) automation tester](https://www.globalapptesting.com/how-we-help/qa-teams). They design and write the test scripts in the beginning stages. The QA automation tester will work with automation test engineers and product developers to actually test the software and products. They will form a team and control the test automation initiatives, and use different types of test automation frameworks to establish the best one for successful test automation.

Which Are The Browsers Supported By Selenium Ide

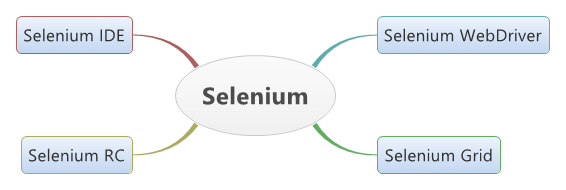
Selenium is not a single tool, instead it is a set of following components/tools:

> Selenium IDE

> Selenium RC

> Selenium WebDriver

> Selenium Grid



As Selenium RC is deprecated and Selenium Grid is just a set of configurations, I will explain the different browsers supported for the below Selenium components only:

> Selenium IDE

> Selenium WebDriver

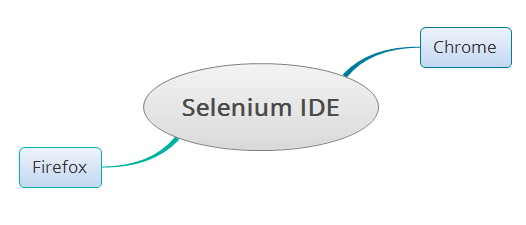
Different Browsers Supported by Selenium IDE:

Latest version of Selenium IDE got released into the market by Selenium guys in the Aug 2018 and it supports the following two browsers:

> Firefox Browser

> Chrome Browser

So, latest version of Selenium IDE can be installed on both Firefox Browser and Chrome Browser



Where as the older version of Selenium IDE used to support only the below browser:

> Firefox Browser

Different Browsers Supported by Selenium WebDriver:

Selenium WebDriver supports all the below mentioned famous browsers in the market:

> Firefox Browser

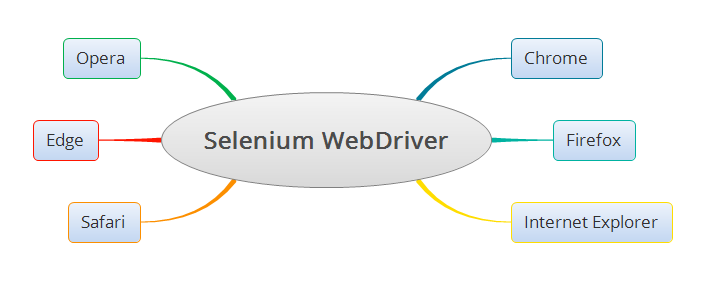
> Chrome Browser

> Internet Explorer Browser

> Safari Browser

> Edge Browser

> Opera Browser



Watch this video on “What are the different browsers supported by Selenium?” for more clarity:

What are the benefits of Automation Testing

Automation testing uses an automation tool to execute the test case instead of a human manually executing the suite by following step-by-step instructions. So the main goal of [automation testing services](https://www.testrigtechnologies.com/service/automation-testing/) is to reduce the number of test cases that has to run manually instead of eliminating manual testing altogether.

Automation testing is pursued not only in the hope of better ROI but also because it increases the area of test coverage and rules out human intervention and human errors. According to a recent report from The Daily Research chronicle, it is being contemplated that automated testing will be on a 100 percent rise from 2019 -2025 amongst the major IT MNCs.

What are the Benefits of Automation Testing?

Let us have a look at the top 10 benefits of Automation testing

1. Enhanced Results

Since automation testing saves plenty of time even when complex and enormous systems are taken into consideration. This allows testing to be carried out repeatedly, delivering better and faster results with significantly lesser efforts and reduced time.

2.  Swifter Feedback system

Automation testing is extremely crucial during the validation phase of any software project. It significantly enhances communication among the developers, designers, and product merchants, and provides space for the potential glitches to be rectified immediately thus enhancing the efficiency of the development team.

  3.  Brand Enhancement

 The effectiveness of testing is always dependent on the quality of test data that is being used. Testing is often performed on the copies of live databases as creating relevant and quality test data takes copious amounts of time. Automation solutions allow you to re-use your data time and again. This saves a lot of costs from project handling and project maintenance perspective.

The best aspect of automated testing is that it adds value to all the attached stakeholders. Automated testing systems not only enhance the system’s capability but also pave the way towards [digital innovation and revolution](https://www.itproportal.com/features/how-automation-can-provide-a-foundation-for-digital-transformation/). It not only improves the brand name but also increases brand recall value, thus ensuring far greater customer retention. Due to automation testing, there are permanent fixes generated to issues long pertained as unsolvable.

4. Cost-effective

Even though the initial investment needed for automation testing is on the higher end, it saves a lot of money for the company in the longer run. It is predominantly due to the reduction in the amount of time required to run the tests. It also contributes a much higher quality of work as there are no chances of neglect or human error. This decreases the necessity of fixing glitches in the post-release phase, thus saving huge amounts of project costs.

5. Efficiency Testing

Testing is one of the most pivotal parts of the entire application development cycle. The most attractive part of the automation testing is that it can be left virtually unattended. This leaves a lot of room for the results to be monitored towards the end of the process. This allows for increasing the overall efficiency of the application.

6.  Increase in Coverage Area

Through the use of automation testing, more tests can be allotted pertaining to any application. This leads to higher testing coverage and a reduction in software anomalies. It also allows room for testing more features and complex applications. However, in order to the same thing in a manual testing scenario would require a massive team along with heavy time constraints.

7.  Detailed Testing

All testers tend to have different testing approaches with different focus areas as per their exposure and level of expertise. With the help of automation, there is an equal focus on all areas of testing, thus assuring the best possible quality of the end product with greater emphasis on each aspect of the product. Automation testing is known for its atom level approach of testing due to which it is considered error-free.

8. Reusability

Test Automation is repetitive in nature due to the nature of its test automation cases. In addition to an easy setup configuration, it gives the software developers an opportunity to assess the program’s reaction. Automated test cases are totally reusable and hence can be utilized for testing any aspect of the code as per significance and through a plethora of different approaches.

9. Earlier Detection of Defects

 Automation testing documents the software defects and hence making it considerably easier for the testing teams. This also makes it relatively easier for the development and support team to together contemplate the defects and give a faster output. The overall development speed of the project is increased while ensuring correct functionality across relevant areas. The earlier any defect is identified, the better and cost-efficient it is to solve and deploy it.

10. Time to market

Test automation helps significantly in reducing the time-to-market launch of an application. Automation testing allows constant and regular execution of test cases. Post automation the test library execution is extremely swift and runs longer.

End Thoughts

Through the above-mentioned factors, it can be clearly established that automated testing results in significant cost and time reduction along with putting the manual efforts into the right use. It paves way for swift feedback thereby contributing to maximized profits. Automation testing can be considered a difference-maker for the Small and medium enterprises for their constant combat to gain a sustainable competitive edge and advantage. Automation testing is something that can be considered a change wave of functionality in the IT domain.

What are the advantages of Selenium

In Web Application Testing it is essential to address issues like website functionality, security, user interface, compatibility, and performance. Test Automation provides a framework to run tests with a combination of various forms of data to enhance test coverage.

Selenium is one of the many tools used for automated testing. But, along with the characteristic advantages of automated testing, Selenium has certain unique benefits that give it a competitive edge over others.

Open Source: One of the most important benefits of Selenium is its Open Source Accessibility. As it is an open-source tool, anyone can download the source code and use it. It can also be refactored based on project requirements. This improves the functionality of predefined functions and classes. Selenium has turned into the most dependable web automation tool on account of the ease of developing test scripts to validate the functionality.

Multi-Language Support: This must be the absolute and most imperative factor for any software skilled individual to get into the Automation Testing domain. The most common languages testers normally use for writing Selenium code are, C#, Java, PHP, Perl, Python, and Ruby. In most cases, enterprises may prefer to stick with the language already in use by testers and thereby eliminate the need for learning a new language. But in several instances, it is convenient to opt for a new language. Selenium is invaluable in such situations.  For instance, if the team is new, or if the software is being revamped and the legacy language is now obsolete, Selenium allows the option to opt for a new and easier language for testing. [Learn more QTP to Selenium migration](https://www.jadeglobal.com/qa-and-testing/quality-assurance/qtp-selenium-migration)

Multi-Browser Support: Chrome, Firefox, Safari, Internet Explorer, Opera, and Edge are the most commonly used browsers worldwide and are compatible with Selenium scripts. One can create Test scripts and execute them in these browsers without any changes in the script. That means, it is not required to rewrite the scripts for every browser, just one script for all browsers, with the help of browser drivers.

Platform Support: Yes, one more advantage we get with Selenium! We need not bother about the system configuration. Once we create test scripts on any operating system, they can be executed on any other operating system. For Example, if we create Selenium test scripts on the Windows platform, the same scripts can be executed on other platforms like Mac or Linux. This empowers QA Engineers/Testers to effectively write the scripts without paying much attention on the platform.

Framework Availability: Frameworks are like templates. They provide a structure to your scripts and help in making code maintenance easy. Frameworks provide increased code re-usage, higher portability, reduced script maintenance cost, and higher code readability. There are different frameworks like Data Driven Framework, Keyword Driven Framework, Hybrid Framework, etc.

Remote Control/Parallel Text Execution: Automated testing aims to save time and effort. Multiple test scripts should be executed in parallel, to reduce the test execution time. With the help of Selenium Grid, multiple scripts can be executed on remote machines. This is one of the most essential advantages of Selenium. Additionally, we can use Online Selenium Grids, for example, Lambda Test, which helps access to more than 2000 browser environments, over which we can run out tests and automate cross-browser testing. With the growing demand for testing deliverables, it is highly important to execute the test scripts quickly and efficiently.

Flexibility: There are a wide variety of frameworks that supports Selenium when compared to other test automation tools in the same class. For example, Test management and Test controller activities (dependencies, grouping, etc.) are performed by other frameworks when we integrate (ex: Testing, JUnit) with Selenium. Hence, it is more flexible and adaptable in terms of automation development and maintenance.

Reusability: As we know, Selenium test scripts can be executed in multiple browsers and operating systems. Moreover, tests written in Selenium reduce a lot of manual efforts required in running them repeat ably.

Integrations: Selenium can be integrated with Testing and Junit for managing test scripts and generating reports. For Continuous Integration testing, it can be integrated with CI/CD tools like Jenkins, Maven, Docker, etc. To perform image-based testing, the Sikuli tool is used to integrate with Selenium. Likewise, we can integrate many other tools with Selenium for better use.

Regular Updates: Selenium is supported by an active community. The Selenium community regularly releases constant updates and upgrades. The finest part about having a community is that these updates and upgrades are ready to use and easy to understand. Selenium has got considerable support and recognition from various communities and vendors. This makes Selenium an Industry Standard Tool rather than being considered as only an open-source tool.

Because Selenium is a complete package of various automated testing solutions, businesses all over the world have started using it for testing the simplest as well as the most complex software applications. With its numerous advantages and easy-to-use features, Selenium proves to be one of the best automated testing tools for the IT industry.

Why testers should opt for Selenium and not QTP

If you are in the field of Software Development and testing, then the web testing tools are not new to you. Probably you have been looking for the best web testing software and came across several options like Selenium, QTP, Win runner, and several others. However, these are not all the same, and it is important to know the difference so as to determine the best to use depending on your environment and the resources that you have.

[Tweet “Here are 5 biggest reasons why Selenium is better than QTP.”]

In this regard, let’s spend a few minutes reviewing the most common question that software testers have been trying to answer for a few years now. You have heard about QTP VS Selenium, and the question is usually why Selenium the best and most preferred software testing tool? Here are some of the facts we found out about these two.

Reason #1: Selenium is an open source testing tool whereas QTP is paid for to use.

Why would people go for a paid too, whereas they can make use of the open source software? Therefore, most software testers will prefer using a solution that is free for use and can be downloaded multiple times. With the open source software, you can achieve the desired correct results and therefore it makes sense to go for the readily available tool with unlimited downloads. Secondly, the fact that selenium is free means more people use it. Hence you are likely to find more knowledge base to leverage on when you use a solution that is commonly used.

Reason #2: Selenium supports multiple languages whereas QTP is mostly a VB scripting environment.

With QTP, it supports VB scripting language whereas Selenium offers more power and flexibility. Therefore, those using Java, .net, Python and Ruby can use Selenium since it accommodates all these scripting languages. VB scripting is not as common compared to these other scripting languages, and this makes a major factor as to why web-based software testing is done using Selenium.

Reason #3: Selenium allows you to write one script for any browser.

The biggest advantage of selenium is that you develop your test script for one browser and you can use the same script for all desktops and mobile browsers with very little modification needed. Therefore, this will save more time since you don’t have to write different scripts for a different browser which isn’t the case with QTP testing tool.

Reason #4: Selenium offers software testers more scalability.

With Selenium grid, you have the ability to run as many tests as you want. Further, you can achieve this when you need and in parallel. You will just need a browser so that you can run the test. With QTP testing tool, there is no resource like selenium grid and hence selenium takes preference here.

Reason #5: QTP doesn’t support several browsers.

Selenium supports major browsers, and also enjoys support from Mozilla, Google, and Microsoft too. These browsers will provide APIs for Selenium, and hence this offers an upper hand to Selenium. However, QTP doesn’t enjoy similar support from the borrowers