**Automation Tools**

**Automation Testing:**

Automation testing is a technique uses an application to implement entire life cycle of the software in less time and provides efficiency and effectiveness to the testing software.

Automation testing is an Automatic technique where the tester writes scripts by own and uses suitable software to test the software. It is basically an automation process of a manual process. Like regression testing, Automation testing also used to test the application from load, performance and stress point of view.

In other word, Automation testing uses automation tools to write and execute test cases, no manual involvement is required while executing an automated test suite. Usually, testers write test scripts and test cases using the automation tool and then group into test suites.

The main goal of Automation testing is to increase the test efficiency and develop software value.

**When we should go for manual testing:**

* If UI of the system under test is changing more frequently, so after every time when the UI changes then the Automated script needs to be updated accordingly.
* If you have tight release deadlines and no enough time to automate the system then I prefer to go with manual testing instead of Automation testing.
* If Identify the test cases which are going to be executed initially once and do not automate such test case.
* To do automation testing you should have the skilled resources with having sufficient programming knowledge. If you do not have the skilled resources to automate the application under test or if you are not ready to invest time and money to educate peoples on automation testing to build good automation team then don’t go for Automation testing.

**When we should go for automation testing**

Consider a scenario where the defect is fixed in the build and similar feature was used in different working modules. So it is hard to check new bug is introduced in previous working functionality. While doing test pass you need to check regression testing around the defect fixes. This testing exercise needs to be executed each and every time you need to manually test the functionality around the impacted area. So considering resources, time and money you need to work effectively and smartly. In such scenarios you need to think of Automation testing.

Test automation is a process to check the software application after development and getting new build or release. The investment for test automation is time, money and resources. In requires initial efforts which will help you whenever you want to execute the regression cases.

**Why to use automation testing:**

1. **Automation testing Saves Time and Money:**

After each development of the Software product the test has to be repeated to ensure quality of the software. Due to every changes of source code the test has to be repeated. When the new software releases, we test that software is compatible to all operation system and also the hardware configurations. Manually testing same test suite repetitively is more costly and time consuming task. But, when we go for automation testing; only the initial cost is there after that it runs over and over again at no additional cost. One more advantage is it can be executed n number of times and they are much faster than manual tests. Automated software testing process reduces the time when test repetition starts, if manual testing is taking a day, automation will only take some hours.

1. **Testing Increases Correctness**

In case of manual testing, the expert tester can do mistake due to each and every time changes of the test methodology, but in case of automation testing the same steps of test repeats each and every time when the source code changes which maintain the accuracy of a software system.

1. **Automate the test due to version changes**

Is it possible to test the application manually whenever the application gets update? Yes it is possible, but will take longer time during test or some time it is not possible. Also, it won’t be effective in terms of company cost, resources, Time etc.

The best way to automate the test as many as the application version gets change and you get lots of regression work.

1. **Increase Test Coverage:**

Automated software testing more focus on the depth and scope of tests which increases the quality of software. Automated software testing process works on thousands of different complex test cases which is not possible with manual testing. If the software is huge and complex, manual testers are scare to test that software but testers who do automation testing can easily work on that particular software, automation testing also facilitates testers to test the software on multiple computers with different configurations. This testing process is capable to check the application inner database, data table, memory, and file containing the application to determine the application is performing as estimated.

1. **Increases Speed, Efficiency,  Quality and Decreases the Cost:**

When we start developing the software, our main goal is to release the software on time. Although, Automation testing process uses same module in different test scenario, run fast. Automated regression test provides the non-stop system steadiness and functionality after changes to the software were completed main to shorter development cycles joint with better quality software and thus the welfare’s of automated testing fast out gain the initial costs.

1. **Testers get Motivated which increases the efficiency:**

In case on manual testing, testers do not get any new technique and tools, they apply manual tricks to test the software, that’s why they don’t get motivation which affects the manual tester’s efficiency. But, in case of automation testing, testers always get different tools with testing software which makes them to work fast with increasing efficiency.

1. **Helpful in testing complex web application:**

Automation testing process is helpful for those web applications where millions of users interact together. If we go for manual testing process, creating those many users manually and simultaneously are difficult or impossible.

So, to test those web applications go for load automation testing and create virtual users to check load capacity of the web application.

Automation testing process can also be used on that software where GUI will always be same and functionalities gets changed always due to source code changes.

# Simple Steps to follow in Automation testing:

There are lots of helpful tools to write automation scripts, before using those tools it’s better to identify the process which can be used to automate the testing,

* Identify areas within software to automate
* Choose the appropriate tool for test automation
* Write test scripts
* Develop test suits
* Execute test scripts
* Build result reports
* Find possible bugs or performance issue

# Different Software testing tools Available in Automation Testing:

|  |  |  |
| --- | --- | --- |
| Application | Tool | Open Source/licensed |
| Windows | UFT | Licensed |
| TestComplete | Licensed |
| Test Studio | Licensed |
| Silk Test | Licensed |
| Sikuli | Open source |
| AutoIT | Open source |
| Web | Selenium | Open source |
| UFT | Licensed |
| TestComplete | Licensed |
| Test Studio | Licensed |
| Katalon Studio | Open source |
| IBM Rational Functional Tester | Licensed |
| Ranorex | Licensed |
| Silk Test | Licensed |
| Watir(Ruby) | Open source |
| Sahi Pro | Licensed |
| IBM Rational Performance Tester | Licensed (performance) |
| Jmeter | Open source(load testing) |
| BlazeMeter | Open source(load & performance testing) |
| Cucumber | Open source |
| Mobile | TestComplete | Licensed |
| Test Studio | Licensed |
| Katalon Studio | Open source |
| Appium | Open source |
| Robotium(Android) | Open source |
| Silk Test | Licensed |

**Windows, Web & mobile Applications**

### **#1) Silk Test (Licensed)**

**Silk Test**is a licensed product of [Microfocus](https://www.microfocus.com/) aims at automated functional and regression testing. It has cross browser support and provides unified test automation for a variety of applications including desktop apps, mobile apps, web apps, rich-client applications and enterprise applications.

It enables efficient, speedy and high-quality automation testing.

### **#2) TestComplete** (**licensed)**

TestComplete is a licensed tool. TestComplete is **SmartBear’s automated testing** solution which allows all level of users to quickly create powerful, reusable and time-saving GUI automation tests for the web, mobile, and desktop applications.

It lets you combine the recorded scripts and tests into a single framework which reduces the training cost and testing time.

The best part of the tool is **TestComplete Visualizer,** which is a screenshot based feature allows you to modify previously recorded tests, quickly update your assertions and checkpoint and provide a visual recording of your test – all in one area.

* Support for multiple scripting languages
* The ability to record robust automated tests without scripting knowledge
* [Regression tests that don’t fail when the UI changes](https://smartbear.com/product/testcomplete/features/object-based-recording)
* [Data-driven testing](https://smartbear.com/product/testcomplete/features/data-driven-testing)

[Custom plugins and extensions](https://smartbear.com/product/testcomplete/features/plugins-and-extensions)

### **#3) Test Studio: (Licensed)**

Telerik Test Studio **is a licensed tool.** Telerik Test Studio is a**comprehensive test automation solution.** It is well suited for GUI, performance, load and API testing.

It allows you to test desktop, mobile and web applications.

Its main features include Point-and-click test recorder, support for real coding languages like C# and VB.NET, central object repository and continuous integration with source control.

**Windows & Web Applications**

### **#1) HPE UFT: (Licensed)**

It is a licensed tool. However**,** the good news is that its trial version (valid for 60 days) is available free of cost. This tool given by [**Hewlett-Packard Enterprise**](https://www.hpe.com/in/en/home.html) is one of the best automation testing software for functional testing. It was previously known as QuickTest Professional (QTP).

It brings developers & testers coming together under one umbrella and provides high-quality automation testing solutions. It makes functional testing less complex and cost-friendly.

Its top features include **Cross browser & multi-platform compatibility,** Optimized distributed testing, multiple testing solutions, image-based object recognition and canvas – visual test flows.

HP Unified Functional Testing uses VB Script as its scripting language. This is the only language that is fully supported by UFT’s IDE. VB Script supports Object Oriented Programming concepts but not polymorphism and inheritance.

### **Supported Browsers:**

Internet Explorer 6, 7, 8, 9, 10, 11, Edge

Firefox 3.0.X, 3.5, 3.6 to 46

Google Chrome till version 56

[Safari on Mac OS](https://www.learnqtp.com/uft-12-new-features-in-detail/) v6 – v9

**Supported OS:**

Windows 7 (SP1), Windows 8/8.1 and Windows 10 including Surface support for Windows 8.1 and Windows 10. UFT 12 onward supports [Safari on Mac OS](https://www.learnqtp.com/uft-12-new-features-in-detail/) (in Beta mode)

**Mobile & Web Applications**

### **#1) Katalon Studio: (Open source)**

**Katalon Studio is a powerful test automation solution for mobile, Web, and API testing. And it is completely FREE!** It provides a comprehensive set of features for test automation, including recording actions, creating test cases, generating test scripts, executing tests, reporting results, and integrating with many other tools in the software development lifecycle.

Katalon Studio runs on both **Windows** and **MacOS**, supporting automated testing of **iOS** and **Android apps, web applications** on all modern browsers, and API services. It can integrate with tools such as JIRA, qTest, Kobiton, Git, and Slack.

### **#2) HP LoadRunner: (Licensed)**

This is again an automated load and performance testing tool provided by [**Hewlett Packard**](http://www8.hp.com/in/en/home.html). It supports testing in various environments and over different types of applications.

Though it’s a licensed tool but it is quite affordable. It supports mobile and cloud testing as well. **HP LoadRunner** gives a clear picture of the system performance, allows you to do the RCA and fix the bugs before the application is released to live environment.

**Windows** **Applications**

### **#1) Sikuli: (Open source)**

**Sikuli is based on image recognition** and has the capability of automating anything that we see on the screen. Currently, it supports desktop apps only which run over windows, Mac or Unix/Linux. This tool is good at reproducing bugs quickly and its users have reported it to be very useful as compared other tools when you are going to automate an application which is not web based.

This tool is good at reproducing bugs quickly and its users have reported it to be very useful as compared other tools when you are going to automate an application which is not web based.

Sikuli is open source testing tool.

### **#2) AutoIT: (Open source)**

AutoIt v3 is a freeware BASIC-like scripting language designed for automating the Windows GUI and general scripting. It uses a combination of simulated keystrokes, mouse movement and window/control manipulation in order to automate tasks in a way not possible or reliable with other languages (e.g. VBScript and SendKeys). AutoIt is also very small, self-contained and will run on all versions of Windows out-of-the-box with no annoying “runtimes” required!

**Web** **Applications**

### **#1) Selenium: (Open source)**

**It is an open source tool.** It is the #1 automation testing tool among all web application testing tools. Selenium can be executed in **multiple browsers and Operating systems.** It is compatible with several programming languages and automation testing frameworks.

With selenium, you can come up with very powerful browser-centered automation test scripts which are scalable across different environments. You can also create scripts using Selenium that is of great help for prompt reproduction of bugs, regression testing, and exploratory testing.

Selenium is often used for automating web applications for testing purposes, but it does not include a testing framework. Some testing frameworks that can be used with Selenium are listed below.

* C# - Frameworks available: [NUnit](http://www.nunit.org/)
* Java - Frameworks available: [JUnit](https://github.com/junit-team/junit), [TestNG](http://testng.org/doc/index.html)
* Javascript - Frameworks available: [WebdriverJS](https://github.com/SeleniumHQ/selenium/wiki/WebDriverJs), [WebdriverIO](http://webdriver.io/), [NightwatchJS](http://nightwatchjs.org/)
* Objective-C
* Perl
* PHP - Frameworks available: [Behat + Mink](https://github.com/Behat/en-mink.behat.org/blob/master/index.rst)
* Phython - Frameworks available: unittest, [pyunit](http://pyunit.sourceforge.net/), [py.test](http://pytest.org/latest/), [robot framework](http://code.google.com/p/robotframework-seleniumlibrary)
* Ruby - Frameworks available: [RSpec](http://rspec.info/), Test::Unit

Selenium-WebDriver supports the following browsers along with the operating systems these browsers are compatible with.

* Google Chrome
* Internet Explorer 7, 8, 9, 10, and 11 on appropriate combinations of Vista, Windows 7, Windows 8, and Windows 8.1. As of April 15 2014, IE 6 is no longer supported. The driver supports running 32-bit and 64-bit versions of the browser where applicable
* Firefox: latest ESR, previous ESR, current release, one previous release
* Safari
* Opera
* HtmlUnit
* phantomjs
* Android (with Selendroid or appium)
* iOS (with ios-driver or appium)

### **#2) IBM Rational Functional Tester: (Licensed)**

This tool is primarily intended for **automated functional testing & regression testing**. It also allows you to perform data-driven and GUI testing. The automated testing in **RFT** is based upon script assure technology which highly improves the efficiency of testing and provides easy script maintenance.

[**IBM RFT**](http://www.softwaretestinghelp.com/ibm-rqm-integration-rft/) supports a variety of web-based and terminal emulator based applications.

### **#3) Ranorex: (Licensed)**

It is a licensed tool. **Ranorex** is flexible, all in one **GUI testing** tool where you can execute automated tests flawlessly throughout all environments and devices. What makes it superior to other GUI testing tools is its super smart object recognition feature that automatically detects any change in the UI and keeps the test going.

Other significant features of **Ranorex** include reusable code modules, early bug finding, seamless integration with other tools, simple test recording and easy to use the editor.

### **#4) Watir: (Open source)**

**Watir (pronounced as water) is an abbreviation for Web Application Testing in Ruby**. It is a very light-weight open source tool for automating web application testing. The best part of the tool is that it supports your web application regardless of considering over which technology your app is designed.

With water, you can come up with simple, flexible, readable and easily maintainable automated tests. There are many big companies that use Watir including **SAP, Oracle, Facebook,** etc.

### **#5) Apache JMeter: (Open source)**

**Apache JMeter** is an open source java desktop application designed for load testing. It mainly focuses on web applications. This tool can also be employed for unit testing and limited functional testing.

Its architecture is centered over plugins with the help of which **JMeter** provides a lot of out of box features. It supports many types of applications, servers and protocols like Web, SOAP, FTP, TCP, LDAP, SOAP, MOM, Mail Protocols, shell scripts, java objects, database. Other features include powerful Test IDE, dynamic reporting, command line mode, portability, multithreading, caching of test results and highly extensible core.

It supports many types of applications, servers and protocols like Web, SOAP, FTP, TCP, LDAP, SOAP, MOM, Mail Protocols, shell scripts, java objects, database. Other features include powerful Test IDE, dynamic reporting, command line mode, portability, multithreading, caching of test results and highly extensible core.

Other features include powerful **Test IDE,** dynamic reporting, command line mode, portability, multithreading, caching of test results and highly extensible core.

### **#6) BlazeMeter: (Licensed)**

**With BlazeMeter,** you can easily create load and performance tests. It is truly compatible with **JMeter** tool described above. Any Jmeter test works well on BlazeMeter as well.

Having BlazeMeter, you can easily setup API tests, do user interactive website testing, perform scalable load testing user virtual user traffic and do a lot more. This tool supports both native and mobile web apps.

It is a licensed tool. But its free testing trial is also available which allows 50 concurrent users, 10 tests, and 1 shared load generator. So, you can actually try doing load and performance testing for free by using this tool.

### **#7) Cucumber: (Open source)**

It is an open-source tool that is designed over the concept of **BDD (Behavior-driven development)**. It is used to perform the automated acceptance testing by running the examples that best describe the behavior of the application. It gets you a single up-to-date living document that is having both specification and test documentation.

Cucumber is scripted in **Ruby**. However, it now supports few other languages as well such as **Java**and.**NET.** It also has cross-platform OS support.

### **#8) IBM Rational Performance Tester: (Licensed)**

This tool is designed for doing**automated performance testing** over web and server based apps. It has **RCA** capabilities to remove performance bottleneck. It provides real-time reporting and test data customizations. It also offers load and scalability testing.

It is a licensed tool. However, **IBM** provides its free trial.

**Mobile** **Applications**

### **#1) Appium: (Open source)**

**Appium** test automation framework is mainly intended for mobile applications. The good news is that it is an open source tool.

It supports automation of native, hybrid and mobile web applications built for iOS and Android. **Appium** uses vendor-provided automation frameworks and is based on client/server architecture.

Appium is easy to install and use. it has gained huge popularity and stability over last few years as one of the best mobile automation testing tools.

### **#2) Robotium: (Open source)**

**Robotium** is an **open-source test automation framework** primarily meant for **Android UI testing**. It supports both native and hybrid applications.

**Using Robotium,** you can write time-saving, readable and easy to use automated gray box UI tests intended for android apps. You can also perform system testing, functional testing, and user acceptance testing over Android-based apps with the help of **Robotium**.

### **#3) Sauce Labs: (Licensed)**

It is a **selenium cloud-based solution** that offers automated testing over cross-browsers and multiple platforms. It has support for both mobile and desktop apps. It is known for significantly accelerating test cycles.

Various well-known companies including **Yahoo, Zillow, and OpenDNS** have testified that they have reduced their testing time by a huge extent with the help of Sauce labs.

This tool is a licensed. However, it also provides free testing for open source projects.