SAKSOFT

Information Management Specialist

Automation using Selenium

Table of Contents

1. A view on Automation Testing

Every software development group tests its products, as the delivered product contains defects. Test Engineers strive to catch them before release of the product with the best manual testing processes but they end up in vain. Automated Testing is the best way to test the product with increase in efficiency. Automated tests are created and can be easily repeated to perform the tasks impossible with manual testing.

Automated testing is considered critical for big software organizations or too expensive for small companies to implement. Automated software testing improves the accuracy, test coverage within short interval of time at low cost.

Test automation has specific advantages for improving the long-term efficiency of a software team's testing processes at low cost and time.

2. Automation Testing Tools

In the competition world, companies/testers have multiple options to pick licensing or open source automation tools such as Quick Test Professional (QTP), selenium etc., based on their requirement.

2.1 Licensed Tools

License is available through single-seat licenses, as well as floating or concurrent licenses and based on the Vendor.

Various automation testing tools are available for performing Functional, Regression and Performance Testing such as QTP, Win Runner, Test complete, Ranorex, QA Wizard Pro, Silk Test, Silk Performer, QA Test, QA Load, and Rational Robot.

Functional Testing Tool -QTP has the feature for storing screenshot of each and every page navigated during the execution. So it can be used as a proof for completion of testing, and also we can refer the screenshots of previous executions if there is any need to refer them. Test report can be automatically written to a customized report page which will ensure accuracy of the report and also it can improve look & feel of the report.

2.1.1 Market Growth & Productivity

Test automation is a way of storing knowledge and increase productivity. QTP survived in the market for its:

- User friendliness and less effort for script creation
- Meets the needs of both technical and non-technical users
- Advanced solution for functional test and regression test automation
- Automation scripts are executed within specific interval of time using task scheduler/crone job.
- Many inbuilt functionalities by default and easy configuration
- Excellent Object Identification process/mechanism
- Easily integrated with test management tools like QC

2.1.2 Current Scenario

In recent economic crisis, software companies are planning for cost reduction and efficient productivity. QTP is losing its market because of the following limitations:

- Runs only in Windows environments
- Cannot test with all browser types and versions
- Limited to smaller organizations/ testing teams
- High licensing and add-inns costs
- Slow in execution when compared with open source tools

2.2 Open Source Tools

Open source tools are typically created as a collaborative effort in which programmers improve upon the code and share the changes within the community, and are usually available at no charge under a license defined by the open source initiative. The tool can be downloaded at free of cost.

Recent days, the open source tools are getting popular because of its integration and interaction, rapid debugging and development with flexibility at low cost by avoiding lock-in to one supplier.

There are lots of Open Source automation test tools available for doing Functional, Regression and Performance Testing like Selenium, SOAP UI, Open STA, Robotium, WebDriver, WebInject, Arbiter, Jmeter, Junit, Nunit, etc.

Among these tools **Selenium** Functional Testing Tool is considered as a portable software testing framework, and one of the best tools available in the current market for web applications at free of cost.

The tests can be written as HTML tables or coded in a number of popular programming languages and can be run directly in most modern web browsers. Selenium can be deployed on Windows, Linux, and Macintosh. Selenium is used for UAT (User Acceptance Test).

2.2.1 Selenium Components

There are four variants of Selenium, which can be used in isolation or in combination to create complete automation suite for your web applications.

- Selenium IDE – Recording/Creating and Enhancing scripts
- Selenium Remote Control (RC)– Executing scripts
- Selenium Web Driver (API) – Cross browser testing
- Selenium Grid – Cross platform testing

2.2.2 Market Trend

Selenium became a most powerful automation tool in market as most of the software developed is browser based and Agile adopted.

There has been a vast improvement in Selenium features that concentrates on simple-to-use, capable GUIs and can meet 90% or more of the needs of the most application testers.

Selenium ranks top in the software market with the following features:

- Simple and powerful document object model (DOM) level testing, can be used for continuous integration with Agile projects
- Has great extensibility and flexibility, along with its tight integration with the browser unmatched by available proprietary tools
- Supports multiple browsers such as Internet Explorer, Fire fox, Safari or Opera on Windows, Mac OS X and Linux
- Wires Object Oriented Programming languages like JAVA, .NET, Ruby, Perl, PHP, etc.

- Provides the option of using wide range of IDEs such as Eclipse, Netbeans, Visual Studio etc depending on the choice of development language.

2.2.3 Limitations

Even though Selenium has many advantages; it has its own disadvantages, which are as follows;

- Supports only browser based application, not the windows application
- Does not support file uploads from local machine
- Provides only partial support for dialog boxes
- Being an open source, Selenium has no official technical support

3. Selenium Vs QTP

Few of the comparison between Selenium and QTP is as follows

FEATURES	SELENIUM	QTP
Cost	Open source & Portable	Licensed and very expensive; Ten user license costs approx. 60L
Application support	Web Applications only	Client server applications only (like built in TCL/TK and PowerBuilder)
Support for web browsers	Supports IE, Firefox, Safari and Opera	Supports IE & Firefox only
Object Oriented Language support & Scalability	Supports Java, .Net, Perl, PHP, Python, and Ruby	Supports VB script only
Support for operating system/platforms	Supports Windows PC/MAC/UNIX Platforms	Supports Windows Platform only
Support for Test management tool integration	Not available. Need to track separately	TD/QC will be easily integrated
Test Development Environment	We can use wide range of IDEs like Eclipse, Netbeans, Visual Studio etc	Need Separate environment
UI object management & Storage	Managed using UI-Element user extension and properties A set of dynamically loaded libraries that is stored in the Java archive file.	Built-in object repository and easy handling
Support for Dialog Boxes	Supports partially	Supports all kinds of dialog boxes
Support for File upload (system)	Not available	Supports all kinds of File upload
Creation of Scripts	Not powerful. Because many actions are not recorded by the IDE and have to be manually entered.	Powerful

Table 1

4. Our experience in Selenium Testing

Saksoft has done many successful projects, using QTP as automation tool. We have also developed our own Framework using QTP and implemented in projects.

We have also implemented Selenium in four projects for its client. We automated the functional test cases in the first phase, integrated the test cases into Selenium and JUnit Framework and executed and maintained test repositories.

We adopted Agile testing techniques and achieved test automation using Selenium. We planned to automate repeated tests and replaced important tests that are difficult to execute manually. We used Firebug tool to identify the XPath of elements displayed on screen to minimize changes to test suite as application changes and for those pages where the ID dynamically changes.

Some user extension files (.js) were added to overcome specific problems e.g. selenium flash.js for flash components. We adopted driver approach for test suites, and Selenium-java-client-driver was used for Java language and multiple Selenium RC servers on different ports to run multiple scripts simultaneously.

We have a dedicated team with expert Selenium specialists/testers and enriched training will be provided to team members with lab sessions.

4.1. Implementation Practice

Following activities were performed during this phase

- Gather the functional flows from client
- Plan the Selenium framework and Test plan preparation
- Allocate skilled resources and provide adequate training new resources
- Software Installation and Test environment set up
- Create and enhance test scripts using Selenium IDE using XPath technique
- Convert scripts to JUnit using Eclipse
- Execute all tests in Selenium RC with Eclipse as an IDE by Batch Scheduling
- Run the scripts in multiple browsers using WebDriver
- Run all tests for every build and multiple builds per day
- Reuse test scripts with less modifications. (i.e.) Updating scripts for every build
- Generate HTML reports and publish to entire team for quick actions

4.2. Areas concentrated while testing

We have extensively concentrated the below areas while test execution:

- Executed Selenium tests as part of each build
- Executed all acceptance tests in the regression test suite based on build release
- Found early bugs and addressed risk of catching bugs late in the cycle
- Tested the existence of Static UI element contents
- Focused on all the navigation links to broken links or missing pages
- Tested the functionality of form based on input page that contains various input fields, buttons and check boxes. Some of its types are User Profile, Account Settings, etc.
- Tested dynamic elements with varying HTML IDs
- Executed tests in browsers such as Firefox, IE, Chrome, Safari etc
- Tested Ajax application that supports dynamically changing UI elements such as animation, RSS feeds, etc. without reloading the web page.

4.3. Challenges faced while execution

Some of the challenges faced while executing the test scripts are enumerated below:

- Application does not respond sometimes due to timeout error in script.
- If application failed/stopped before the verification code and if we run the script again then the current application does not accept the new verification code and script fails.
- Frequently design of the application changes and due to this element of the verification code changes.
- In the Admin Console section under the Application queue table, the element-ID of the submitted application of data changes regularly.
- Script fails some times, because the application does not move from one stage to another
- Script failures due to Element not found error
- Script failures because of Popup Windows and Frames
- Version compatibility issues on Selenium and Java components
- Generic and wage report generated in eclipse

Though there were many challenges faced, with the expertise of the team we resolved them successfully.

4.4. Saksoft Solutions

We have implemented the following solutions to overcome the challenges mentioned above in different projects automated using Selenium.

- **Changing IDE commands manually**

Test scripts can be automatically recorded and edited manually using IDE Command with the combination of Command, Target and Value
Replacing **Click** by **ClickandWait** and increasing the load time to open a new page.

- **Locating Elements by XPath Using Firebug Tool**

Converting all objects/elements into XPath (Benefit- Scripts will never fail on different machines/browsers and even if elements location is changed).
Example : //input[@name="mname"] - if value is assigned to any attribute, //a[text() = “Maps“] - if value is not assigned to any attribute

- **Handling “Element not found on page”**

The potential cause of “**Element not found on page**” is that Selenium does not capture some critical elements on the web page. This error has been rectified by adding **Wait command** or adding element label in script using XPath.

- **Handling of Popup Windows**

To handle Popup/Alert Windows in Selenium IDE, observe the popup Window ID and add the command- storeAlert and value ($variableName) & storeAlertPresent and value ($variableName) before AssertAlert messages

- **Handling Ajax effect/control**

To handle Ajax effect/control, we implemented the following commands across the test suites.

- clickAndWait

- waitForValue() - When a field value changes in the current page, we used this command to Selenium wait until the expected value appears.
- waitForCondition() used to allows to specify a Boolean expression in Javascript which causes Selenium to wait until the expression evaluates to true.

5. Conclusion

Using Selenium as the Functional Test Automation Tool, we reduced the cost incurred for licensing using QTP. We reaped all the benefits of Automation testing. We supported the client to deliver customized versions faster from its end customers. We automated more than 80% of test cases within short interval of time.

In our experience we found that the efforts involved in scripting for Selenium increased by about 15 % than QTP in the initial stages. Once all the solutions were in place the efforts of automation using Selenium and QTP were almost the same. The licensing cost benefits were huge and clients were very satisfied with using Selenium as their standard automation tool for regression.