



DESIGN OF AUTOMATION SCRIPTS EXECUTION APPLICATION FOR SELENIUM WEBDRIVER AND TestNG FRAMEWORK

Rishab Jain C and Rajesh Kaluri

School of Information Technology and Engineering, VIT University, Vellore, Tamil Nadu, India

E-Mail: Rishabjainc@gmail.com

ABSTRACT

To develop and deliver software to the customer, validating its quality is highly important. Software testing can be performed manually or using automation tools to identify defects, assess the quality of the product and gain confidence in the software being developed. Automation tools helps in design and execution of test scripts saving time and cost involved in manual testing. This paper mainly focuses on the automation testing tools currently available to support design and execution activity, challenges faced by manual tester in executing automation scripts, approaches in executing of automation scripts using TestNG and its disadvantages and then overview of the proposed web application which overcomes the problems faced by manual testers, reduce the time spent on initial set-up activity to carryout test scripts execution and overcome disadvantages of execution using TestNG.

Keywords: Selenium WebDriver, TestNG, Quick Test Pro, Test Complete, Manual Testing, Automation Testing.

1. INTRODUCTION

Software testing is a process of validating the developed software against functional and non-functional requirements. It helps to determine the quality of a product based on the attributes like functionality, reliability, usability, efficiency, maintainability and portability as specified by ISO-9126-1 model [5, 6]. Automation tools are test management tools that help in carrying out testing activities like test design, execution, defects management and test coverage analysis. All the automation tools are not licensed tools. Some of the open-source tools includes selenium for functional and cross-browser testing, Jmeter for performance testing, Bugzilla for defects management and Testopia, TestLink for test management. The main objectives behind the use of these automation tools are speeding-up the testing process and meet the given timeline. There are various types of testing and use of these testing types depends on the need of the application being tested, time constraints, budget allocated to carryout testing activity.

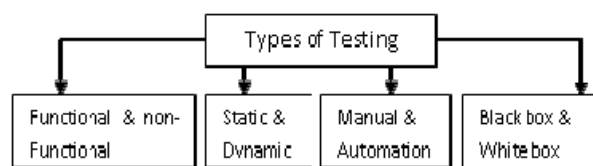


Figure-1. Various types of testing.

The objectives of testing are to ensure that the system meets the needs of the end-user and the client, identify defects in the system, to assess the quality of the software being delivered to the client and to gain confidence in the system being developed. There are various software development models like waterfall model, agile model, spiral model and iterative enhancement models. All these models have a testing phase in it. The steps involved in the testing process cycle are:-

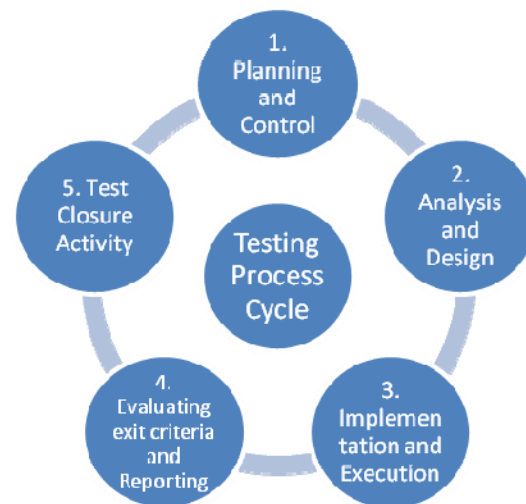


Figure-2. Various activities in testing process Cycle.

In the above testing activities, execution is one of the phases after analysis and design activity. The main focus is on execution of test cases by a manual test engineer or execution of test scripts by an automation test engineer. Manual tester can execute the automation test scripts, if he has sufficient knowledge of the tool being used and the framework designed by an automation tester, programming skills to debug the error in the scripts. Prior to start with execution of the available test scripts, the manual tester should create a set-up in the local system by installing necessary softwares, create a workspace and import the latest code from code repository. The above tasks should be performed by all the manual testers who wanted to execute the automation scripts.

The reason behind manual tester executing automation scripts is saving of his manual effort and time involved in executing test cases manually. Our main focus is to help manual testers in execution of the test scripts without installing any automation tools or creating



workspaces in his machine and save cost and time spent in downloading (and installing) the necessary softwares.

2. METHODOLOGY

2.1 Manual testing and Automation testing

In manual testing, initially all the requirements are clearly analysed. After freezing of the requirements, test design starts which involves design of test artifacts like test strategy and test plan document, test scenarios and test cases manually by a manual tester. After design activity, all the test case are reviewed by the testlead or a subject matter expert (SME) and then once the build is deployed into the testing environment execution phase starts. In test execution phase, each test case are executed and it is associated with a status (pass/fail/blocked/not executed/no run). All the failed test case are to be logged as defects. Once the defects are fixed, that particular test case is to be re-tested. But, during fixing of a particular defect there are possibilities that other modules or other functionalities may be affected and leads to defects if other modules are re-tested. So, regression testing is carried out to ensure that the bug is fixed, all existing functionalities are working and no new bugs are introduced into the code. Regression testing involves re-executing of all the failed test case and other test case that are passed but, are dependent on the functionality of the failed test case. So, the passed test case are also re-executed. Regression testing is done by the manual testers many times before moving the code to production. The total number of test case in the regression testsuite may vary depending on the total number of requirements being covered in that release. For a major release, there will be many number of test

case and executing all these test case repeatedly due to bug fix or for regression testing consumes more time.

Some of the problems faced during manual testing are:

- Changes in the requirements.
- Documenting of the bugs that were logged, bugs fixed and those that are still open using an excel sheet.
- Maintaining of Requirements Traceability Matrix (RTM) without using any tools manually is very tedious job.
- Carrying out regression testing on a daily basis or bi-weekly or as needed by manual testers is not possible.

To overcome the above problems, testing team prefers use of test management tools and specific type of testing automation tools.

2.2 Automation tools for Software testing

Automation testing is a process of validating the functionality of the application by designing scripts using a programming language and executing it as and when needed. Automation tools are software developed to support various activities of testing called test management tools and few tools are developed which are specific to different needs of testing like functional testing, performance testing, security testing, web services testing, database testing, usability testing etc. The automation tools that are available in the market can also be classified as open source and paid tools [3]. Once the manual test case are automated, manual effort of executing the test case for regression testing by the manual test engineer can be eliminated by executing the scripts using test execution tools. Some of the automation tools that help in carrying out the test script design and execution are [4]:

Table-1. Comparison of automation testing tools.

Parameter	SELENIUM WebDriver2.0	QTP	Test complete
Programming Language	Scripts are commonly designed using Java, Ruby, Python, PHP, and Perl.	Support scripting using JavaScript and VBScript only.	Allows scripts to be designed in VBScript, JS Script, Delphi Script, C++ Script, C# Script.
Platform Supported	Windows, MAC, UNIX operating systems.	Supports only in windows.	Runs on windows Vista and 7.
Open-Source / Paid Tool	Completely Open-Source.	Paid tool based on each project	It is licensed tool and costs lesser than QTP.
Customer Support	Open source so no much support provided.	Online support provided to the customer and End-user.	Customer support is provided by the vendor.
Test Execution Report	Do not generate report by itself. There is a need for TestNG APIs and plug-in like TestNG to execute and generate reports in html format.	Test execution report is generated using QTP and determine the execution status.	Execution results are generated as a separate file and displayed to the user.

With the current trends and developments in the market with respect to the project budget allocated for each project, the cost of purchasing or renewal of the

licensed tool is very high and may exceed the allocated budget. So, the alternative solution to the use of commercial automation tools is to adapt or migrate from



commercial tools to the open-source tools. In the above mentioned automation tools, it is efficient to have open source tools like selenium webdriver, which has many advantages and supports multiple functionalities as compared with the licensed automation tools.

2.3 Selenium WebDriver and TestNG

Selenium webdriver is the latest version of the selenium IDE and selenium Remote Control(RC).It is also named as selenium 2.0 . It allows the designed test scripts to communicate with the browser directly with the help of native methods. It supports testing of web applications on desktop as well as on mobile devices like Android and iOS devices. The cost of project is reduced with the help of this tool since it is an open-source tool. The time taken to execute a test scripts on webdriver is less when compared to selenium IDE and Selenium RC.It allows the test scripts to be designed for different browsers like Internet explorer, Firefox, Mac safari and Chrome. The test scripts can be developed using languages like Java, C#, Ruby, Perl, Python[4].

TestNG is a testing framework used for automation testing along with selenium 2.0[2].It support various levels of testing like unit, integration, system and user acceptance testing (UAT).It is commonly named as "Test New Generation".

Features of TestNG are:

- It supports use of annotations like @Test,@BeforeTest,@AfterTest,@BeforeClass, @AfterClass,@Parameters.
- Allows passing of parameters to each of the test methods.
- Grouping of test case based their priority is possible.
- Listeners helps to track number of test case passed, failed and not executed.
- Test methods names can be any user-defined names.
- Parallel execution of test case with TestNG increases the throughput of the test execution.

2.4 Need for TestNG with Selenium

- It helps in executing of the selenium test scripts and generating of test execution reports as selenium webdriver do not generate any kind of test summary report.
- TestNG listeners helps in tracking of all the test scripts that are passed,failed and skipped.
- Assertions can be made to compare expected result with the actual result during the execution of the test scripts.
- To execute a Selenium test script,TestNG generates an xml file which contains the details of all the testclasses and test methods to be executed.This file is generated everytime even if a single testmethod is executed which is stored on localfilesystem.

3. EXISTING SYSTEM

In current system, few applications are tested only using manual testing process and many other

applications do have both manual and automation testing [1]. Only manual testing is preferred for few applications testing because

- The application to be tested is small.
- Increase in project cost due to hiring automation resources or purchasing of tools if necessary.
- Unavailability of as many automation resources as needed before the project could start for a specific automation tool which is planned to use for automation testing.
- No Regression testing involved once the code is moved into production.

3.1 Approaches to start-up with open-source automation testing tools

To kick start with automation testing, initially there is a need to analyse the various types of testing that are to be performed which are mentioned in the test strategy and test plan document. It could be functionality testing, security testing, performance testing, usability testing, webservices testing etc. Each of these testing requires the test lead, test manager and business analyst(BA) together with the other stakeholders to carry out a case study on the current tools available in the market and compare them to choose the best one according to the needs and requirements of the client and the application to be tested. After deciding the tool, a small feasibility study is to be carried out to ensure that the tools that are selected meets all the requirements and to know if any additional training is needed to other team members on this tool.

3.2 Design of automation scripts using selenium2.0 and TestNG API

To startup with automation using open source tool like selenium webdriver, the framework is developed which will be used throughout the project. Once the framework is ready, the test scripts are designed using selenium webdriver API, testNG API along with one of the programming languages supported by selenium. A sample test script (Test1.java)

```
package Sample1;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.*;
import org.testng.Assert;
import org.testng.annotations.*;
import org.testng.asserts.*;
public class Test1 {
    @Test
    public void Sampletest(){
        WebDriver driver=new FirefoxDriver();
        driver.get("http://www.google.com");
        String title=driver.getTitle();
        Assert.assertEquals(title, "Google");
    }
}
```



After completion of scripts designing, execution starts. Executing of test scripts using java compiler and interpreter will not help in generation of test execution report after execution is completed. The alternative is to use TestNG as a plug-in for script execution which is again an open-source and can be easily downloaded.

3.3 Two Approaches for execution using TestNG and its disadvantages

One of the approaches is to manually design an xml file including all the java files (test classes and test methods) in it and then executing using the TestNG plug-in. A sample xml file with two testclasses Test1.java and Test2.java

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM
"http://testng.org/testng-1.0.dtd">
<suite name="Default suite">
  <test verbose="2" name="Default test">
    <classes>
      <class name="Sample1.Test1"/>
      <class name="Sample1.Test2"/>
      <class name="Sample1.Test3"/>
      <class name="Sample1.Test4"/>
    </classes>
  </test>
</suite>
```

The xml file specified above is simple xml file. There are other complex xml files to be designed to run automation tests and this approach may be a little difficult for a manual tester.

The other approach is to run the required java file one at a time. The disadvantage is that it consumes more time when there is a need to execute some 100 test files. So, it is preferred for testing only if we wanted to execute one or two test files and debug for errors in the scripts that are designed. To execute the test script, TestNG internally first creates a folder and the xml file in it (xml file is shown previously) into our local file system and then executes it. Also, next time when some other script file is executed, there is another folder newly created with the xml file in it. TestNG creates as many as xml files equal to the tests run. This consumes huge amount of storage space on the secondary storage device. The size of each folder varies depending on the size of xml file. But, the minimum size of each folder is 1KB.

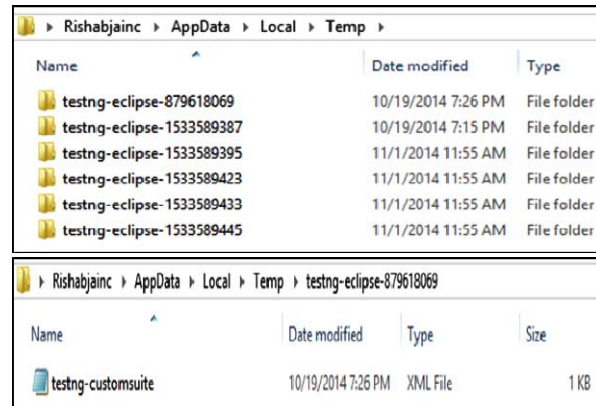


Figure-3. TestNG suite file generated on local file system.

3.4 Challenges in execution of automation scripts by manual test engineer

For a manual tester, execution of test case is very easy as it is been designed by himself or another manual test engineer. But, execution of automation scripts designed by the automation testing team becomes difficult because of lack of knowledge on the automation framework and the tools used for automation.

Following are the list of activities to be followed by each and every manual testers who would like to utilize the available automation test scripts:-

- Knowledge Transfer(KT) session on the End-to-End automation framework either through conference call or through direct interaction or self-study of the system understanding documents that are prepared by the automation testing team.
- Training on the automation tool being used for test script designing.
- Basic programming skills on the language being used for test scripts design which may be Java, Python, Ruby or C# so that during execution of the scripts debugging of the code and identifying the cause of failure in the scripts could be easy.
- Installing all the necessary softwares, creating workspace, importing the latest code from code repository, resolving the dependencies like setting up the classpath and adding of external jar files into the project code.

The above activities should be carried out by multiple manual testers to take the advantages of readily available automation test scripts instead of validating the same functionality manually. To carry out all the above activities by a single manual tester it takes time, which is between 8-10 hours or even more depending on the skill set that manual tester already posses and the complexity of the framework. To overcome the above problems faced by the manual test engineer and disadvantage of TestNG which utilizes huge space on the secondary storage for generation of xml file for each tests that are run, we



propose a web application that helps manual tester in executing of all the test scripts and reduce the time spent to just 1hour and also eliminate creation of separate folder and xml file for each test that is run.

4. PROPOSED SYSTEM

The basic idea is to have a web project replacing stand-alone project. The proposed system is a web based application running on the apache tomcat server. It can be accessed by the entire QA team (Quality Assurance) connected within the network. This application uses TestNG API to generate xml code for scripts execution. The web application is designed using Eclipse as an IDE, languages used for implementation is DHTML(HTML, Java Script, CSS), JSP and Excel file(.xls) to store all the test classes which is a java file. The execution of test scripts is carried out by the webserver and the results are displayed on the JSP page to the user. Below is an overview of the proposed web application.

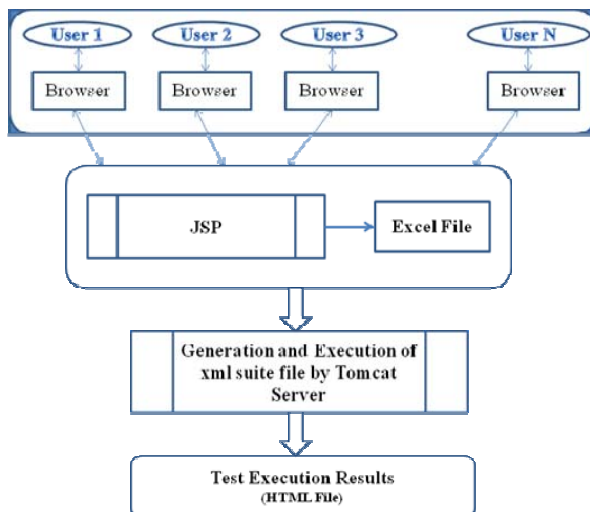


Figure-4. Overview of the Architectural design of the Web application.

4.1 Features of the proposed web application

- This Application provides an innovative, cost-effective and efficient way for execution of designed automation scripts as it does not generate a xml file on the local file system.
- It helps in invoking the required test scripts by just selecting the testclass from the user interface there by eliminating designing of the xml files by the manual testers.
- It keeps track of the number of testmethods under execution.
- After completion of test execution, the test summary report can be downloaded.
- Display of the test execution metrics to the users.

- If the test scripts execution takes a longer time then the user can request for sending a mail. The mail has attachment of the scripts execution summary report.
- To execute the automation scripts, manual and automation testers can access this application without having the project code in their local system.
- Each and every user within intranet requires a just a browser to access this application.

4.2 Input-output screens

Figure-5. UserInterface for selection of testclass for execution.

Scripts Execution Status	
Passed:	1
Failed:	0
Skipped:	0
Total:	1
Pass Percentage:	0

Download Execution Report

Figure-6. Test execution metrics.

5. CONCLUSIONS

One can use any type of testing tools that supports activities across testing life cycle. There are



various parameters to be taken into account like time, budget, size of the project and technical competency of the individual members in a team. In this paper, we have described the process of manual and automation testing, some of the widely used open-source and licensed automation tools that can help reduce manual effort and save time (and cost), approaches to test scripts execution using TestNG and its disadvantages, challenges faced by manual tester in automation scripts execution. Our proposed web application will help manual testers in automation scripts execution without installing automation tools. It also saves their time and effort which in turn is a saving to the project cost. Our web application, overcomes the disadvantage of generating xml files on the disk by TestNG for scripts execution thereby saving disk space. This application can be easily integrated with existing selenium and java projects that uses TestNG frameworks for automation testing.

REFERENCES

- [1] J. Lee, S. Kang and D. Lee. 2012. Survey on software testing practices. *IET Softw.* 6(3): 275-282, doi: 10.1049/iet-sen.2011.0066.
- [2] Deepti Gaur, Rajendra Singh Chillar. 2012. Implementation of Selenium with Junit and TestNG. *International Journal of Computer Science and Management Studies.* 12(03), ISSN (Online): 2231-5268
- [3] Monika Sharma, Rigzin Angmo. 2014. Web based Automation Testing and Tools. (IJCSIT) *International Journal of Computer Science and Information Technologies.* 5(1): 908-912.
- [4] Harpreet kaur. 2013. Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Test Complete. *Journal of Engineering Research and Applications,* ISSN: 2248-9622, 3(5): 1739-1743.
- [5] Thamer A. Alrawashdeh, Mohammad Muhairat and Ahmad Althunibat. 2013. Evaluating the Quality of Software in ERP Systems Using the ISO 9126 Model. *International Journal of Ambient Systems and Applications (IJASA).* 1(1).
- [6] Deepshikha Jamwal. 2010. Analysis of Software Quality Models for Organizations. *International Journal of Latest Trends in Computing (E-ISSN: 2045-5364),* 1(2).