ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 5, May 2016

Design of an efficient automation framework for the Web application

Akshay Lakhotiya
Dept. of Computer Engineering
College of Engineering, Pune
India

Prof. Dr. V.K. Pachghare
Dept. of Computer Engineering
College of Engineering, Pune
India

***Abstract*—In today's market, software is being released at unprecedented rates, which is putting a strain on the traditional software development processes. Getting such software released to the users is usually a painful, risky and time-consuming process. Most of the software applications are written in such a manner that it'll run in a web browser, such application is apprehended as a web application. Testing these web application is an integral, time-consuming, repetitive and expensive activity. Tester has to test the application inside the fundamental measure assigned to them before delivering it to the client. In some cases tester has to have put additional time and effort to fulfill the point in time. One solution is to have an automation framework which can be useful for the tester to test the web application within time. To save the time of repetitive testing work, it's highly recommended to automate testing efforts by having a good automation framework.**

In this paper, authors have designed an automation framework for testing the web applications. The proposed automation framework can be used with selenium to the test the web application. Selenium is an open source automated testing tool for web applications across completely different browsers and platforms. Selenium is the software testing tool one will use with the proposed framework to own powerful and efficient testing of the web applications. It provides two ways that to check the appliance, record/playback testing and domain-specific language to write the testing scripts in many languages. The main advantage of the proposed framework is re-usability of the testing code and scale back the time of testing the web application. At the start, writing the test scripts for a new area may take a long time, but an important quantity of time goes to cut back as tester explores the testing of the respected area.

Keywords—web application; selenium; object layer; action layer; test script layer;

I. INTRODUCTION

In today´s fast moving world, the world-wide-web has become the most important information source for most of us. As a result of this, the web applications are becoming more popular to impress and gain the potential marketplace. It's become a challenge for all the companies to continuously maintain and improve the quality and efficiency of the developed web-application. Many times testing is neglected by the companies or they're going to perform the testing with poor test strategy which ends within the client discontent. Thus, the need for testing of those applications has become more important. Testing the web application during the application development phase has become an integral part of the software Development Life Cycle and these tests are generally planned to check the functionality of the application.

It is also attainable to test the application by using the manual testing strategy. But once we tend to compare manual testing with the automation testing then it results that automation testing has several benefits over the manual testing. Manual testing is time-consuming and tedious Since test cases are executed by human resources thus it is very slow and tedious. Whereas automation testing is fast, since Automation runs test cases significantly faster than human resources. In manual testing investments need to be done for human resources which end into higher cost, but this is not the case with automation testing. Manual testing is less reliable as compared to automation testing, as tests may not be performed with precision each time due to human errors. Because of the above reasons automation testing gaining the more and more importance over the manual testing. Automation testing allows software to be developed to a high customary and easily packaged and deployed to test environments, resulting in the ability to rapidly, reliably and repeatedly push out enhancements and bug fixes to customers at low risk and with nominal manual overhead. The automation of tests is initially associated with increased effort, but the related benefits can quickly pay off.

The goal of this paper is to set a design an automation framework for testing the web application. The proposed framework is nicely developed, easy to perceive, and can be used with the selenium to have powerful and efficient testing of the web application. Selenium is developed by Jason Huggins and released in 2004. Selenium is a test tool that simulates web browsers and supports several programming languages. The languages supported by the selenium are C#, Java, Ruby, Groovy, Python, PHP, and Perl. Selenium is a set of powerful completely different software tools working with several browsers, programming languages, and testing frameworks each with a different approach to supporting automation test for testing web-based applications. Selenium has four components.

1. **Selenium Remote Control (RC) :** it's a test tool that enables the tester to write automated web application UI tests in many programming languages against any Http website using any mainstream JavaScript-enabled browser.

2. **Selenium Integrated Development Environment (IDE) :** It is a Mozilla Firefox add-on which is capable of recording the user actions and also playback. A tester can also edit and debug the recorded tests.
3. **WebDriver :** It is used to verify that the web application works as expected. It provides friendly API that's simple to explore and perceive.
4. **Selenium Grid :** It supports distributed test execution which means that it allows for running test scripts in a distributed test execution environment. A tester can test the written test scripts on different machines against different browsers in parallel.

II. PROPOSED FRAMEWORK

The Framework working is based mostly on the layered architecture. The proposed framework has three layers Object layer, Action layer, and test script layer. Object layer represents controls present on the window of the application under test. The Action Layer contains methods for a specific task, mostly that methodology is of common operations performed throughout automation. Actions are methods that manipulate objects or controls present on the web application. Object Layer is the one which is responsible for the interface with the application under test. The test script Layer contains the code that verifies the test points present in the test case description. Every test script does a setup, drive to a test point, verify the function, and finally does the clean up action.

III. IMPLEMENTATION APPROACH AND DESIGN

The working of the framework is straightforward, it works as mentioned below. Object layer is the one which interacts with the web application under test, it contains methods that once called returns the respected control. Action layer interact with object layer, it calls the methods of the object layer to get the control and performs the particular task. Test script layer contains the one test script for each test case. Test script interact with the action layer, it calls the method that drives to a test point and verifies the test case is passed or failing on the basis of the value come back by the called method of the Action layer.

The design of the proposed framework is as shown in the diagram. The detail description about the working of each layer is as follows:-

1]Object layer :-

Object layer where you can store data regarding web application's user interface elements. It is also wherever you may write your Getter strategies, which returns objects enabling the Caller to question and manipulate these user interface components. Typically, these methods are referred by the Action layer. Object layer separates the object maps from the test cases. Object maps are placed in scripts whose sole purpose is to return the objects to the caller.

Each control of the web application under test is represented by one object. The files within the object layer contain the methods that return the controls present on the window of the browser. It is always a good practice to write down related objects within the same file, Unrelated objects in different files. Example: All the controls on the login page window(two text boxes and two buttons) should be written into the same file. The purpose of object layer is to carry scripts that return the user interface objects contained in the web application. Each script in this layer includes a private object map and several other methods that merely returns the object which will be used by action layer scripts.

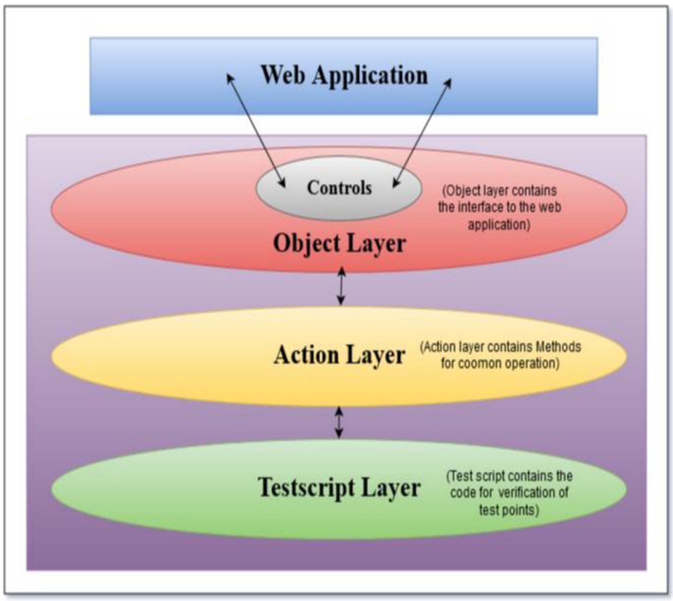
Web Application
Controls
(Object layer contains the interface to the web application)
Object Layer
Action Layer
(Action layer contains Methods for coomon operation)
Testscript Layer
(Test script contains the code for verification of test points)

Fig. Design of the automation Framework

2]Action layer :-

The Action Layer contains methods for common operations such as login into the web application, changing the login password or register a new user. Actions are methods that manipulate objects or controls present on the window of the web application. Action-layer methods invoke object layer methods in order to gain access to user interface elements of the web application. In turn, methods outlined within the action layer are invoked by test scripts. The main advantage of the action layer is that it promotes code reuse, as methods defined in it can be used by many test scripts. Test scripts simply call the methods outlined in the action layer and gets the resulted value. It means that Action layer hides the low-level implementation details from the test scripts.

some of the keys point to the methods written within the action layer.

- Actions are methods that manipulate multiple controls in combination to achieve complex goals. Action layer Methods contains the basic behavior of a single control. Example: A task method would be one that enables the Search toolbar, puts the focus on it, types a string into the Search text field, then clicks the Search button, checks for error dialogs and waits for the resulting Search Results view to appear.

- Action methods are meant to manipulate the web application so as to achieve a point wherever a verification test can be performed in an exceeding test script. Example: The above-described Search task method would be called with a specific string as the search target. After the method returned with success, the test script would examine the results to verify that the expected matched the actual results.
- Actions are meant to be generic enough to be reused by a lot of than one test script in more than one functional area. Non-generalized functions which are for one area or small set of test case will be set under the same test script. Example: In the above-described search task, the data passed to the method would be stored within the test script, but the manipulation would take place within the actions. Many totally different test cases might then call this method, and verify that the expected results came up in the specific test case.

3]Test script layer :-

The test script provides the most general view or point to point view of a specific test case.The test script layer contains the code that verifies the test points. Each test script can do a setup, verifies a test point, verify the function, and finally does the clean up action. Test scripts should call actions and object layer methods to manipulate the web application, but verification of the test points should be done in test script code. Every test script invokes methods of the action layer and will pass data as the parameters if necessary. The called method returns the expected value which gets compare with the actual value provided by the tester within the test script, this is the verify point within the test script. If both the value expected and actual get matched then tester can log results of the test case as passed otherwise failed. Test scripts should contain only the simple logic(call the appropriate method of action layer and compare result) and flow of control(points described as in the description of the test case); all else is reserved for the methods that were outlined inside the action layer.

Methods of action layer should be written as generic as possible so that it'll get utilized by plenty of test scripts which results in the re-usability of the code and save time and automation efforts. For this sufficient amount of time and efforts should be devoted to the development of the action layer. One more advantage of doing this will be that it makes the job of inexperienced testers too easy so that he/she can quickly get accustomed to the framework and might come back up with new test scripts ASAP.

IV. Case Study

Here is a simple case study based on the above framework. Let us assume that the web application under test contains a login page at the start of it. The login page has two text fields to get 'userName' and 'password' from the user, also has two buttons 'login' button and 'cancel' button. to access the web application, one needs to give correct userID and password. The user with correct credentials will have the access to web application otherwise, access has denied.

Tester has to check this functionality of a web application that whether or not a user with correct credentials is able to login into the web application. So the functionality to be checked is "Login page window". We will describe this test case as, "User with the correct credentials(userName and password) should be able to login into the web application." Tester has to write a test script during which he/she can perform some straightforward step to complete the above test case. The steps are as mention follow.

step1] open the web browser and load the login page URL.

step2] provide the userName and password to the login window.

step3] user with correct credentials is permitted to access the web application - positive testing. A user with invalid credentials should not get access to the web application - negative testing.

Step4] Logout in the case of successful login and close the web browser.

➢ For the above test case take a look at, what will go in the object layer of the above framework?

Object layer will have all the controls which are present on the login page window, those controls are text field for userName, text field for password, 'login' button and 'Cancel' button. Let's assume that the language used to develop the framework is java. So the Object layer can have a class file with some name say 'loginPageObj' and four controls defined in it. Each control is defined by one method which when called will return the respected control.

➢ what will go into the Action layer?

Each time user desires to access the web application must perform the login page action, these type of action is that the one which is going to be reused by the tester over and over. It is better to outline such action in the action layer of the framework as they are getting to use repeatedly for future automation. A method with some name says 'loginPageAction' are going to be outlined within the action layer which is able to contain the complete code from loading the URL in the browser to clicking on the 'OK' button after proving the userID and password. This method will be defined as Boolean method that can have two parameters userID and password. On the successful login, the method will return the value 'true' otherwise, it'll return 'false'.

➢ what we can have within the test script?

A tester can divide the test script into three parts setup part, main part, and cleanup part. The setup needed for the test case to run are going to be the primary a part of the test script. Open the application under test, can be the setup part for the test script. Similarly, the test script can also have clean up part say close the application under test.

The main part of the test script is to invoke the methods outlined within the action layer, passing data if necessary(userID and password), verify conditions, and log results. Each test script contains only the simplest logic and flow of control; all else is reserved for the Action layer. The main part of the test script is to ascertain that the login into the web application is successful or failing for the provided userID and password. To verify this, simply call the method 'loginPageAction' outlined within the action layer additionally give the required two parameters for the method. Tester has to provide the expected value(for the above test script it's Boolean value 'true' or 'false'), the called method will return a boolean value can be the actual value for the test script. If both expected and actual value get matched then the test case is passed otherwise failed.

The method 'loginPageAction' are going to be utilized in several test cases, therefore, the above framework leads to the re-usability of the code and conjointly leads to the time-saving effort for the longer term automation get reduced.

V. Conclusion

software testers can be faced with tougher challenges to fully test the web application within the time given to them. The proposed automation framework will be helpful for a tester to automate the web application efficiently, and it also frees engineer from tedious and redundant manual testing operations. The above framework leads to speed up a web application testing process and to reduce software testing cost and time throughout a software life cycle. The framework also helps to increase the quality and effectiveness of a software test process by achieving pre-defined adequate test criteria during a restricted schedule. As tester can use the framework for the generation of reusable test Scripts, it leads to reduce manual testing activities and redundant test operations using a systematic solution to achieve a far better testing coverage. The framework is designed in such a way that the overhead for maintaining the test script is extremely low, as the maintenance of automation scripts is simple and systematic.

Acknowledgment

I would like to thank Prasad Purandare and Sunanda Patil for their invaluable suggestions and guidance.

References

[1] Software Testing Automation Framework(STAF) Website http://staf.sourceforge.net

[2] Weider D. Yu Giri Patil, "A Workflow-Based Test Automation Framework for Web-Based Systems", Computer Engineering Department, San Jose State University, San Jose (Silicon Valley), California, USA.

[3] Alex Cervantes, "Exploring the Use of a Test Automation Framework", Jet Propulsion Laboratory, California Institute of Technology

[4] J.D. McCaffrey, "Low-Level Web Application UI Test Automation", Microsoft Developer Network Magazine, October 2005, vol. 20, no. 10, pp. 37-42.

[5] Xin Bai and Marco Sadeghi, "IT Module Based Test Automation Framework", The Microsoft Corporation, 2011 Eighth International Conference on Information Technology: New Generations.

[6] Fei Wang and, Wencai Du, "A Test Automation Framework Based on WEB", 2012 IEEE/ACIS 11th International Conference on Computer and Information Science

[7] The Robot FrameWork website. "A generic test automation framework" [Online], http://code.google.com/p/robotframework

[8] Sandeep Sivanandan and Yogeesha C. B, "Agile Development Cycle: Approach to Design an Effective Model-Based Testing with Behaviour Driven Automation Framework", Cisco Systems (India) Private Limited, Bangalore.

[9] The Software Testing Automation Framework, by C. Rankin.

[10] Ying Wu, Automation Testing Framework for web base on web base on selenium. Infomation Techonlogy[J],2011,9:187-188