



Design and Implementation of Hybrid Test Automation Framework for Web Based Application

Sabastain Raju J, Vaidhehi V

Department of Computer Science, Christ University, Bangalore, India

Manuscript History

Number: IJIRAE/RS/Vol.04/Issue03/MRAE10097

Received: 07, March 2017

Final Correction: 20, March 2017

Final Accepted: 20, March 2017

Published: March 2017

Abstract -- There many ways to create Web application UI Test Automation Framework. This paper elaborates the creation of the Hybrid Test Automation Framework, and explains the architecture and design of the framework. This Framework is easy to use and incorporate the use of high level programming language allowing object oriented design, permits low level access to the UL components and recover quickly for UI changes in the application under test. The architecture of the framework as multi layer design which isolates the browser UI compounds and test scripts. This Framework Architecture is a merged combination of Library and Data Driven Test Automation Framework. This Framework contents multi packages with classes which isolates the reusable functions and involves the external database on fetching the data. In this framework we can execute the same test case repeatedly in a loop to handle different test input which is retrieved from the database.

Keywords— Test Framework, Data driven Architecture, Library Test Architecture.

1. INTRODUCTION

In this paper we discuss about creating and overview of the architecture, design and implementation of a browser automatable hybrid framework. The test framework is based on Library Architecture and Data Driven Architecture; both are simple and effective architectures which are used for automation testing. Library Architecture framework are based on higher implementation of Module Based Testing Framework, which is evolved after Module Based Testing Framework. Data Driven Testing Framework helps the user separation of the test script code and the test input data from each other. It lets the user store the test input data into a separate data file. Combination of both the framework would make an efficient and easy process to automate a Web Based application using Selenium as the automation supporting tool.

The automation testing performed with the help of testing tools or some kind of programming languages to control the testing class. The main idea behind automation testing is to automatically do the testing work without human interactions; it reduces the cost and also makes testing more reliable and effective. Many tools are developed for this testing requirement. Selenium is automation tool which supports the different way or approaches to test application. The entire suite of tools contains a rich set of testing functions specifically for the needs of testing of web applications of all types. These operations are highly flexible, having many options for locating UI elements and comparing expected test results against actual application behavior. One of Selenium's features to support for executing one's tests on multiple browser platforms.

1.1 NEED OF TEST AUTOMATION

Software companies are not only required to test the software adequately, but also it is required to test thoroughly and quickly. To perform this goal automation testing is needed. Manual testing is a time taken process and error prone, automation help to overcome this drawback by running the test frequently increases confidence in the application. Execution the test cases will also helps the user to understand the functionality. The set of automation tools can form a regression test suite. Automation also helps in finding the problem in earlier stages and fixing it.

1.2 RELATED WORK

This section discusses various methods that have been used on creating the Hybrid Architecture; this section will also state the related works which are done on different framework.

In [1], Chethan M. proposed a better framework approach than the Modular Based Testing Framework, He claimed using Library Architecture Testing Framework is better as the code can be reused when saved in the methods or functions, the author is also stated the advantages of using the Library architecture. In [2], Anbunathan, proposed Data Driven Architecture for Android Mobile. He mentioned it is challenging task to generate Data Driven architecture. It required an external data base provide input to the test scripts. In Data Driven the same test scripts are executed multiple times for different provided inputs. They have explained the effectiveness of this method based on experimental result. In [3], Zeng Wandan, Jiang Ningkan, Zhou Xubo, in this paper the author as introduce the new framework based on object feature set and dynamic searching policy, using the obtained result it shows that the framework is more convenient and effective with less resources and time cost by higher test coverage. [4] In the paper author Hari SankarChaini, have mentioned the process of test script design and execution as well as failure analysis of the scripts. Author as also mentioned about the different failures which comes under false positive.

II. LIBRARY ARCHITECTURE TESTING FRAMEWORK

Quality library architecture framework is very similar to the test script modularity structure and offers the same advantages, but it splits the application-under-test into methods and functions (or items and methods with regards to the setup language) rather than scripts. This kind of framework requires the creation of library files that represent modules, sections, and functions of the application-under-test. These library files are then called directly from test case script. Many like script modularization this framework also yields a high amount of modularization and adds to the overall maintainability of the testing. The basic fundamental lurking behind the framework is to determine the common steps and group them into functions under a selection and call those functions in the test intrigue whenever required [1].

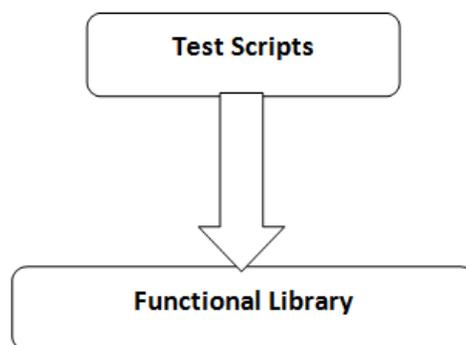


Fig 1: Library Based Automation Framework

ADVANTAGES

- Higher level of code reuse is achieved in Structured Scripting
- It is less costly to develop as the higher code reusability achieved
- Easier Script Maintenance

DISADVANTAGES

- Technical expertise is required to write Scripts using Test Library Framework.
- Time consuming to plan and prepare test scripts.
- Test Data is hard coded within the scripts

III. DATA DRIVEN ARCHITECTURE FRAMEWORK

On automating or testing any application, sometimes it may be required to test the same functionality repeatedly multiple times with the different set of input values. Thus, in such cases, we can't let the test data entered in the test script. Hence it is advised to retain test data into some external data base or data files outside the test scripts. A data-driven framework is where test input/output values are read from data files and are loaded into variables or manually coded scripts. In this framework, variables are used for both input values and as well as output verification values. Navigation through the program, reading and retrieving data from the data files, and loading of test status and information are all coded in the test script. This is similar to table-driven testing in which the test case is containing in the data file and not in the script; the script is just a "driver," or delivery mechanism, for the data. In data-driven testing, only test data is contained in the data files [2].

ADVANTAGES

- Less coding required as it covers all the possible combination of test scenario
- Changes done on the test data will not affect the test scripts
- Increases flexibility and maintainability
- A single test scenario can be re executed with multiple test data

DISADVANTAGES

- The process is complex and required additional efforts to create data sources and read mechanisms

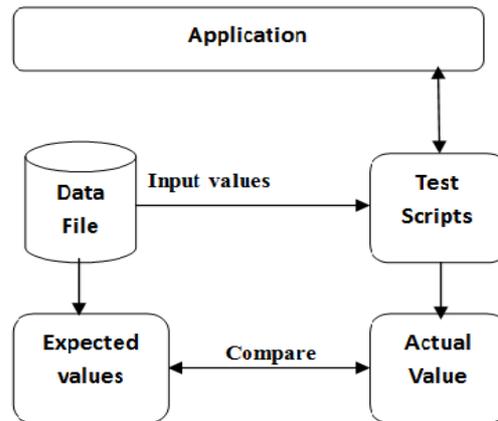


Fig 2: DDA Based Test Automation Framework

IV. PROPOSED FRAMEWORK ON HYBRID ARCHITECTURE TESTING FRAMEWORK

In this section, architecture and design constraints of proposed Hybrid architecture based test automation framework is discussed.

OVERVIEW OF THE HYBRID ARCHITECTURE TESTING FRAMEWORK

This Framework is being combination of Library Architecture and Data driven Architecture. This allows data driven scripts to take advantage of the powerful libraries and utilities that usually accompany library architecture. The framework utilities can make the data driven scripts more compact and less prone to failure than they otherwise would have been. The utilities can also facilitate the gradual and manageable conversion of existing scripts. Here we have the function for each common step under a common library and call it in the test scripts when ever required, it also contain the data file which provides the data from the outside storage medium, like files, SQL or any data storage medium from which we can retrieve the data needed to test the software.

ARCHITECTURE OF THE HYBRID FRAMEWORK

Hybrid framework is the existences of multiple frameworks forming one single framework. Here the library architecture is merged with data driven architecture to form a hybrid architecture of the both architectures are very efficient in their own ways.

Below is the description of the architecture flow.

FRAMEWORK FOLDER STRUCTURE

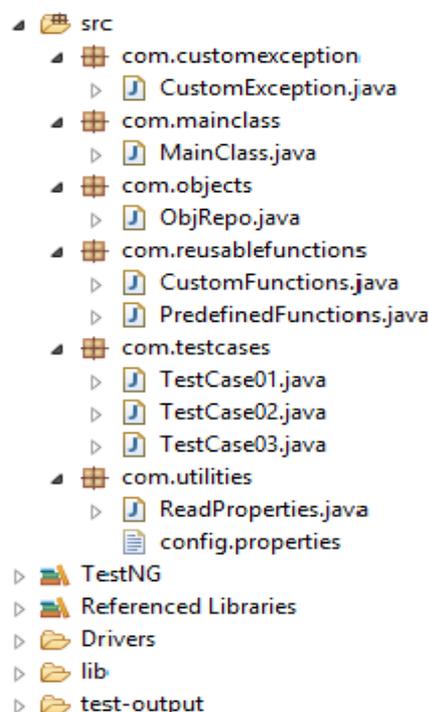


Fig 3: Folder Structure of Hybrid Framework

A. CUSTOM EXCEPTION FOLDER

This is a folder which contains the user defined exception class which extends the Exception class, which is mainly used to handle the exception occurred at the time of execution. They are used to provide a simplified message for the users about the errors occurred while executing. They are basically helps the users to understand the failure and for debugging effectively.

B. MAIN FOLDER

This folder contains the class which would call the XML file for evocation of the test scripts. They are TestNG xml file. This can be used for setting the priority for test cases execution flow. They are the first class which would be executed.

C. OBJECT FOLDER

In this folder we have captured the various web elements or web element identifiers in the form of variables which can be accessed in other places of the Framework.

D. REUSABLE FUNCTION FOLDER

The folder is constituted of the classes which contain functions and methods that can be shared and used amongst the multiple classes. Very often, we are suppose to perform certain operation prior and before to the actual test execution like login to the application, setting up environments, activities related to rolls, data manipulations, sending results, methods those perform pre/post-conditions to other methods. Since we tend to perform these activities for all or most of the test script. Thus it is always recommended to create a separate class for such activities instead of coding them repeatedly inside each of the test script.

E. TEST CASES FOLDER

The "test" folder is constituted of majorly test suite and the folders representing the many modules of the software under test. Thus, each of these folders is made up of the test scripts specific to the module that it is associated. Test suite is a logical blend greater than one test scripts. Thus, the end user can mark an admittance of any of the test script within the test suite that he/she wants to execute in the subsequent runs.

F. UTILITIES FOLDER

In this folder they are two classes one for the Read Properties –which are having the code to read the properties file and send the data to the test cases and as an input for the test steps, other is Properties file the static variables referencing to the paths and other environmental details. These details can be Application URL, URL to the Databases, Credentials for Databases, and URL to any third party tool being used. Properties class has the string data which would be accessed by the read properties method.

G. REPORT FOLDER

[5] Reporting is the most important part of any test execution, as it helps the user understand the result of the test execution, point of failure, and the reasons for failure. Logging, on the other hand, is important to know the execution flow or for debugging in case of any failures. TestNG, by default, generates a different type of report for its test execution. This includes an HTML and an XML report. TestNG also allows its users to write their own reporter and use it with TestNG. There are also options to write your own loggers, which are notified at runtime by TestNG.

H. LIBRARY FOLDER

In the Library folder we have stored all the supporting jar files which are required by the project. These jar files are used by the Java classes. [6] Jar files are in ZIP file format and is used for aggregating many files into one Jar. JARSare used as a general archiving tools, the primary motivation for its development was so that Java applets and their components (.class files, images and sounds) can be downloaded to a browser in a single HTTP instances, rather than opening a new connection for each time. In our project we have used Selenium, TestNG Jars for accessing the functionality of these tools.

V. HYBRID ARCHITECTURE FRAMEWORK

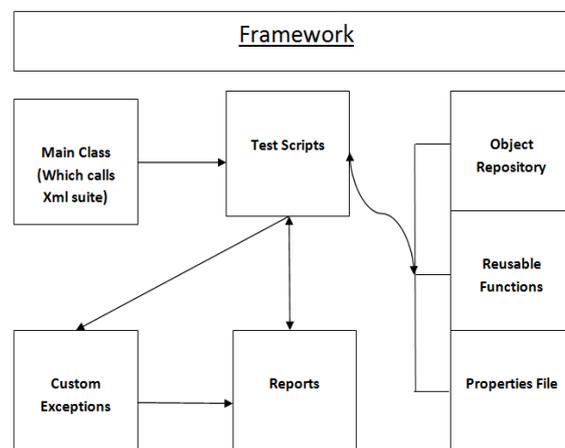


Fig 4: Process flow for Hybrid Framework

Using Framework is it easy to structure our code and make maintenance quite easy. We would have placed out data and code together which would rather be re usable nr readable without framework. Hence using the framework as provide various benefits like code re-usage, higher portability, reduced script maintenance cost etc. In our Hybrid Framework, Firstly the Main.class would be executed on calling the Main.class which calls the TestNG XML suite file, in which we have arranged the order of testcase execution. When our first test case is been executed it would have the Predefined functions which are retrieved from the Reusable function folder they are code which would be used in many places can be click event, enter the text box value, select value from drop down box and many more. These function are been called in the Custom function, which as the reusable functionality like login page, this class would contain the function which can be used for reusable functionalities. They take the values from Object. class and Properties file. Object.class would have the variables of the web element and where as Properties file would have the details of values which needed to enter like, application URL, Credentials and etc,

On executing the test case if they are some error of failure occur in the execution flow they are indeed captured in the CustomException.class. They are extending Exception.class and they are used to handle the exception which occurs in the execution, they provide a meaning full message on every error which appears. On the completion of the execution finally the report would be generated by the TestNG reporting feature [7]. TestNG, by default, generates multiple reports as part of its test execution. These reports mainly include TestNG HTML report, TestNG email able report, TestNG report XML, and JUnit report XML files. These files can be found under the output report folder

VI. CONCLUSION

Hybrid Framework is rugged, easy to implement, easy to use, easy to expand and easy to maintain. It is Technology and Platform independent and it is also separate from the Test Design. Once it is created it improves the speed and maintainability of automation Test Cases which uses the reusable libraries and reduces cost. In this paper we have explored the implementation of Hybrid Test Automation Framework and the different advantages of independent architectures like Library Architecture and Data Driven Architecture and how it can be effectively used during automation. We have also discussed about the folder structure and the execution flow of the framework. The framework is easy to extended and maintain. In order to support more complex tested cases, more kinds of applications will be researched in future work and it will also be checked for mobile implementation. At the same time, the exception handling in testing will also be further researched and implemented in more effective way.

REFERENCES

- [1] Chethan M and Padma MC, "Test Automation Framework on Library Architecture" Department of Computer science & Engineering, PES College of Engineering, Mandya
- [2] Anbunathan R and AnirbanBasu, "Data Driven Architecture Based Automated Test Generation for Android Mobile" Test Manager and Research Scholar, Bharathiar University Coimbatore, India
- [3] Zeng Wandan, Jiang Ning kang, Zhou Xubo, "Design and Implementation of a Web Application Automation Testing Framework", Software institute of East China Normal University, Shanghai, China
- [4] Hari SankarChaini and Dr. Sateesh Kumar Pradhan, "Test Script Execution and Effective Result Analysis in Hybrid Test Automation Framework", Department of Computer Science and Application Utkal University, Vani Vihar Bhubaneswar, Odisha, INDIA
- [5] CédricBeust, a senior software engineer at Google, about TestNG available at: <http://testng.org/doc/documentation-main.html>
- [6] ORACLE, Java Documentation about Jar files is <https://docs.oracle.com/javase/8/docs/technotes/guides/jar/jarGuide.html>
- [7] TestNGtutorials for learning the usage of the tool<http://www.tutorialspoint.com/testng/>