

Test Automation Framework on Library Architecture

Chethan M
M.Tech in Computer Engineering
Department of Computer Science & Engineering
PES College of Engineering, Mandya
doncheetha@gmail.com

Padma M C
Head of the Department
Computer Science & Engineering
PES College of Engineering, Mandya
padmapes@gmail.com

Abstract - A Test Automation Framework provides an execution environment for the automated test scripts. In Test Automation Framework for Library Architecture we make use of a central Library where all methods written based on the workflow of applications, we make use of these methods further to write the test scripts.

Keywords - Automation testing, Interface, Library architecture, Application.

I. INTRODUCTION

When a product is developed, it is very important that the product is tested for the correct functionality it has been developed for, before it can be sold to customers or clients. Thus, the role of testing in the software development process is very important. It should not happen that, when a customer is using the software, there should not be defect or a bug.

Automation Testing is a method where we use testing tools and reduce the need of manual or human efforts. Automated testing is used by software engineer to save time and resources. Test automation is the process of writing a computer program to do testing otherwise which had to be done manually. Once tests have been automated, they can be run quickly and repeatedly [1]. This is often the most cost effective method for software products that have a long maintenance life, because even minor patches over the lifetime of the application can cause features to break which were working at an earlier point of time. Test automation offers a possibility to perform these types of testing effectively [2].

Test automation follows many approaches; some of them are [3]:

- **Graphical user interface testing:** In this a testing framework generates user interface events such as keystrokes and mouse clicks.
- **Code-driven testing:** Here the classes, modules or libraries are tested with a variety of input arguments to validate that the results that are obtained are correct.
- **API driven testing:** This testing framework uses a programming interface to the application to validate the behaviour under test.

II. NEED OF TEST AUTOMATION

Companies not only need to test software adequately, but also it is required to test quickly and thoroughly. To accomplish this goal, automated testing is required. Manual testing is a time consuming process and error prone, automation testing helps to overcome this draw back. Running the tests frequently increases confidence in the application. Executing the test cases will also help us understand what portion of the functionality is implemented. The set of the automated test suite can form a regression test suite. Automating also reduces the time in finding the problems at an earlier stage and solving them.

III. TEST AUTOMATION FRAMEWORK

A framework is a combination of set of protocols, rules, standards and guidelines that can be incorporated or followed as a whole.

A test automation framework sets the rules of automation of a specific product. This also integrates the function libraries, test data sources, object details etc. These components are the small building blocks which represent a business process. The framework is the basis of test automation and also simplifies the automation effort.

The main advantage of a framework is to provide automated software testing in a low cost maintenance. If there is change to any test case only the test script should be changed, if there is any change in the application test script need not be changed but the new test case for change should be taken care.

If suppose we have a group of testers and if each project implements a different strategies then the time taken for the tester become productive in the new environment which will

take long. To handle this we cannot change the automation environment for new applications, instead we develop a testing framework that is application independent and has capability to expand the requirement as the application expects. Testing framework also helps in avoiding duplication of test cases automated across the application. In brief Test frameworks help teams to organize their test suites which in turn help improve the efficiency of testing.

Choose of the right framework for scripting which helps to reduce the costs. The costs associated with the framework are due to development and maintenance efforts.

There are various framework techniques that are generally used [4]:

1. Linear approach: This is usually procedural code, possibly generated by tools.
2. Structured approach: This uses control structures - typically 'if-else', 'switch', 'for', 'while' conditions/statements
3. Data-driven approach: In this data is persisted outside of tests in a database, spread sheet, or other mechanism
4. Keyword-driven approach
5. Hybrid approach: Here two or more of the patterns or approaches are used.
6. Agile automation framework

There are many responsibilities for testing framework [5]

1. Defining the format in which the results are expected.
2. Creating a mechanism to drive the application which is under test.
3. Executing the test cases.
4. Reporting the results in the expected format.

Requirements for automated suite [6]:

1. Business Requirements
 - Maintainability: Ease of maintain and update of the scripts.
 - Selective Testing: It should be easy to select test scenarios at more granular level.
 - Recovery: Test Suite should be able to resume testing from one point of time to another.
 - Scalability: At the same time we should be able to execute tests from different machines.

- Usability: Anyone in the team should be able to execute automation suite.

2. Technical requirements

- Data abstraction: Test data must be completely isolated from the actual script code.
- Maintainability: Automated suite should be be easily maintainable for updates.

IV. TEST AUTOMATION INTERFACE

Test automation interface are the platforms which provide one workspace for many testing tools and also frameworks for System or Integration testing of application which is under test. The aim of Test Automation Interface is to ease the process of mapping tests to business criteria without coding. Test automation interface are used to improve the efficiency and flexibility of maintaining test scripts [7].

Test Automation Interface consists of the following modules [8]:

1. **Interface Engine:** Interface engine consists of a parser and a test runner, where parser is present to parse the object files which are coming from the object repository into the test specific scripting language and the test runner executes the test scripts using a test harness. Interface engines are on top of Interface Environment.
2. **Interface Environment:** Interface environment is which, that consists of Project Library and Framework Library. Framework Library contains modules which are related with the overall test suite while the Project Library contains of modules specific to the application under test.
3. **Object Repository:** Object repositories contain Application object data which are obtained by the testing tool while exploring the application under test.

V. AUTOMATION FRAMEWORK TARGETED BENEFITS

A framework is built on top of a complex internal architecture of the automation tool. This enforces the required standards for implementation. It gives structured design and organization of automation code into components. Test Automation framework must target the following factors [9]:

1. Faster test scripts generation
2. Design that enhances the outcome of code scripted.
3. Longer automation code life.
4. Re-usability of test code.
5. Suitable coupling of the Test Automation Framework and the application under test.
6. Software migration support.
7. Ease of maintenance.
8. Extended reporting capability.
9. Data driven test.

VI. MODULE BASED TESTING FRAMEWORK

Module based Testing Framework is based on Object Oriented Programming concept that is Abstraction. The framework divides the entire Application under Test into number of logical and isolated modules. In this for each module, we create a separate test script.

These modules are separated by an abstraction layer that the changes made in the sections of the application don't affect on this module [10].

Hence, these test scripts are taken together to build a larger test script representing more than one module.

Advantages:

1. The framework is scalable
2. This framework gives high level of modularization which helps in ease of maintenance and also it is cost effective.
3. If there is any change made in one part of the application, it only requires the change in test script representing that part, the other part of the test script remains the same.

Disadvantage:

While writing the test scripts for each module separately, we embed the test data into the test scripts. So, whenever there is requirement to test with a different set of test data, it requires the manipulations to be made in the database or any other repository.

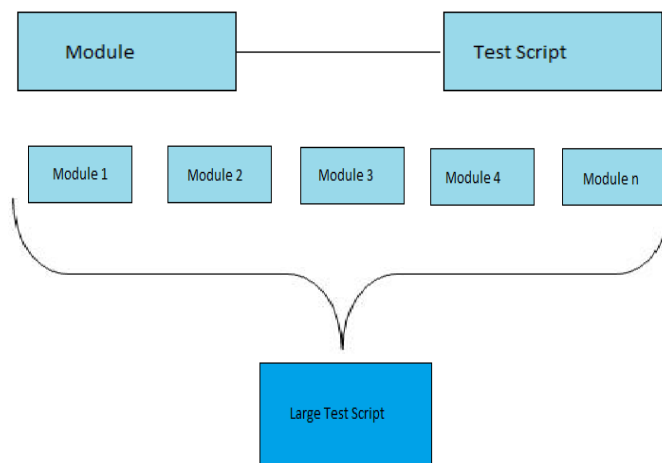


Fig 1: Module Based Testing Framework

VII. PROPOSED FRAMEWORK ON LIBRARY ARCHITECTURE TESTING FRAMEWORK

The Library Architecture Testing Framework is built on Module Based Testing Framework with added advantages. Here Instead of dividing the application under test into several test scripts, we segregate the application into common functions that can be used by the other parts of the application also.

The basic concept behind this framework is to determine the common steps and then group them into functions under a common library and call those functions in the test scripts whenever required.

Hence we create a common library constituting of common functions for the application under test. These libraries are called within the test scripts whenever or wherever required.

Here the login steps can be combined into a function and kept into a common library, all the test scripts those require to login the application can call that function.

Advantages:

1. The common functions which we create can be efficiently used by the various test scripts across the Framework. Hence, the framework introduces a great degree of re-usability.

Like Module Based Framework, Library architecture also introduces high level of modularization which results in easier and cost efficient maintenance and also scalability

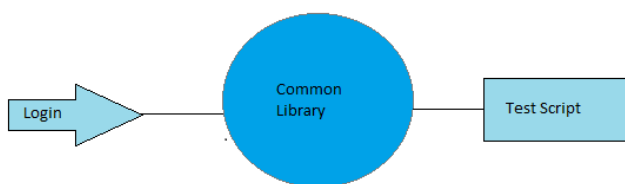


Fig 2: Library Based Automation framework

CONCLUSION

The main criterion of test framework is it should meet certain standard requirements they are:

- Framework should be application-independent.
- The test framework must be easy to expand, maintain, and perpetuate.

Library Automation Testing Framework is the quickest and easiest to implement. This framework reduces the automation cost, since the common library contains the functions which can be reused the risk of maintenance is also reduced. Technical automation experts implement and maintain a reusable automation framework independent of any application that will be tested by it.

REFERENCES

- [1] Test automation Framework
http://en.wikipedia.org/wiki/Test_automation
- [2] Software Automation Testing
<http://www.hcubesys.com/software-testing/>
- [3] What is automation testing?
<https://maheshm23.wordpress.com/2012/01/27/what-is-automation-testing/>
- [4] Manual Testing and automation testing questions
<http://www.qtptestinginterviewquestions.blogspot.com/2010/07/what-is-automation-framework.html>
- [5] How to test the data warehouse
<http://myqaengineering.blogspot.in/>
- [6] Automated Regression Testing Framework using Keyword Driven Approach by Rajeev Randhawa Lokesh Pandey.
http://edge.sapient.com/assets/imagedownloader/149/sapient_automatedregression.pdf
- [7] Automated testing: Group 4 – Evan Feller, Lawrence Green, Alexander Eirmann, David Duffrin, Ryan Shohoney
http://www.uwplatt.edu/~shiy/courses/se373/final-presentation/Group4_Final_Presentation.pptx
- [8] Test automation framework and guidelines
<http://wenku.baidu.com/view/60bac511f18583d049645988.html>
- [9] Functional Test Automation Leverage Automation Frameworks for Efficiencies in Software QA
http://geometricglobal.com/wpcontent/uploads/2013/03/Geometric_Functional_Test_Automation_White_Paper_March_20093.pdf
- [10] Module based Testing:
<http://www.pcids.co.in/common-selenium-interview-questions-and-answers.php>