Automating Web Application using Hybrid Framework

Hanamanth Pujari, 4th Sem M.Tech , Software Engineering
Rabindranath S, Associate Professor, Department of Computer Science
AMC Engineering College, Bengaluru

***Abstract:-* Software testing is an integral part the software development cycle. Software must pass through testing process before releasing to the market.**

Software testing is the process of detecting and correcting the errors in the product by inspecting the expected and actual outputs of the product under test by providing possible inputs. Each and every product cannot be released to the market unless it is tested. Therefore testing will help to gain customer interest towards the product by reducing the defects in the product.

Software development lifecycle can have testing at each stage to ensure error free propagation of development process.

Automation testing is gaining popularity now a days because of disadvantages associated with the manual testing. Complexity involved in software development process is one of the reasons to go for test automation.

Automation testing helps to achieve increased efficiency and performance. By adopting test automation technique, 50% of test execution time can be reduced. Cost of test execution process can also be reduced and the test execution can be done without human interaction, this will helps in avoiding the mistakes usually made by test engineers.

BACKGROUND

Software testing is a costly and time consuming process of the software development cycle. Testing and software quality assurance will play major role in developing any application. Software test automation involves automating the activities of software development process and testing.

Over past few years the tools that help the programmer to build an application quickly with a graphical user interface also boosted the pressure on testers to test the product with less time. As new versions of a software by adding several new features are released, a set of test cases are designed and each test case is executed accordingly. Applications can have many test cases, executing thousands of test cases manually will take more time.

In order to reduce the amount of time taken for manual test case execution, test automation came into picture. Many tools are available which help to automate GUI, to reduce the test execution time and the cost.

Many tools provide record and play-back feature through which user can record the testing steps and play back many times. Such tools need the framework to be developed which provides a structural view for the automation practice.

FRAMEWORKS

In general, a framework is a set of guidelines, standards, concepts and reporting mechanisms. Frameworks provide the skeletal structure for using which user can develop the project.

An automation framework is a skeletal structure which provides an execution environment for automated test scripts. It provides the user with various benefits which helps them to develop, execute and report the automation test scripts in an effective and efficient manner.

Advantages of a test automation framework

- Reusability of code
- Maximum test coverage
- Recovery scenario
- Low cost maintenance
- Minimal annual investment
- Easy reporting

TYPES OF AUTOMATION FRAMEWORK

Various automation frameworks are available; these frameworks differ from each other based on their support to different key factors such as reusability of the code, ease of reporting and maintenance.

MODULE BASED FRAMEWORK

Module based automation framework divides the entire Application Under Test (AUT) into number of isolated and logically related modules. Separate and independent test scripts are developed for each of the modules.

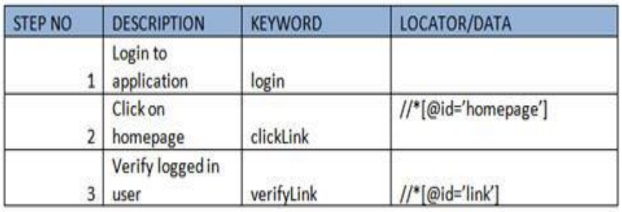
Modules are separated from each other in such a way that the changes made in one module will not affect the other modules.

DATA DRIVEN FRAMEWORK

Data driven automation framework separates the test script logic and the input test data from each other. It allows the user to store the input test data into separate external database.

KEYWORD DRIVEN FRAMEWORK

The keyword driven automation framework is an extension to the data driven automation framework, it not only separates the data but also the part of code and stores in the external database.

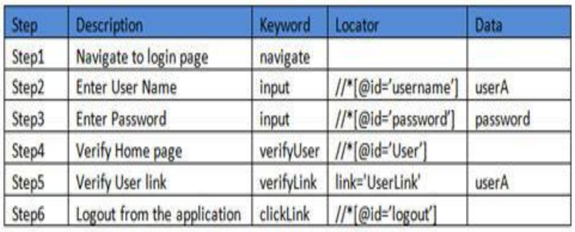


STEP NO	DESCRIPTION	KEYWORD	LOCATOR/DATA
1	Login to application	login	
2	Click on homepage	clickLink	//*[@id='homepage']
3	Verify logged in user	verifyLink	//*[@id='link']

Table: External database containing keywords.

HYBRID FRAMEWORK

The hybrid framework is the combination of module based, data driven and keyword driven frameworks. The main advantage of hybrid framework is that it combines the benefits of the other frameworks.



Step	Description	Keyword	Locator	Data
Step1	Navigate to login page	navigate		
Step2	Enter User Name	input	//*[@id='username']	userA
Step3	Enter Password	input	//*[@id='password']	password
Step4	Verify Home page	verifyUser	//*[@id='User']	
Step5	Verify User link	verifyLink	link='UserLink'	userA
Step6	Logout from the application	clickLink	//*[@id='logout']	

Table: External database consisting of both keywords and data.

PROPOSED SYSTEM

In this paper we are automating a web based application which helps to create, monitor and manage all the platforms and partitions on the server in a virtualised environment

A platform is a collection of partions and each partition can be created by considering required hardware and software resources available on the server. So far our concept supports 12 platforms and 32 partitions.in each platform.

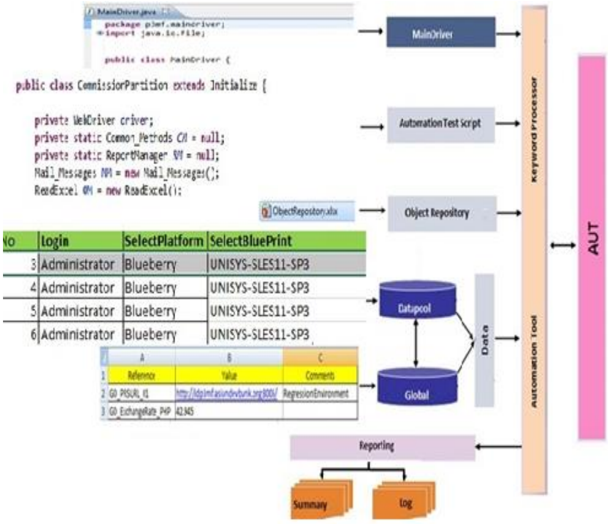
To create a partion, we need to select the Operating System image, number of cores, memory, ports, and server chasis. Operating Systems supported are flavors of Windows and Linux.

Each chasis allows dedicated number of cores and memory to be selected for creating a partition.

Unavailability of CPU, memory and ports are handled by the web application by throwing appropriate error messages.

Web application consists of several web elements, each operations like identify, click, enter text, check radio button and accessing excel sheets are taken care by the script.

FRAMEWORK MODEL



```
MainDriver.java
package p3mf.maindriver;
import java.io.File;

public class MainDriver {
```

```
public class CommissionPartition extends Initialize {

    private WebDriver driver;
    private static Common_Methods CM = null;
    private static ReportManager RM = null;
    Mail_Messages MM = new Mail_Messages();
    ReadExcel RM = new ReadExcel();
```

No	Login	SelectPlatform	SelectBluePrint
3	Administrator	Blueberry	UNISYS-SLES11-SP3
4	Administrator	Blueberry	UNISYS-SLES11-SP3
5	Administrator	Blueberry	UNISYS-SLES11-SP3
6	Administrator	Blueberry	UNISYS-SLES11-SP3

	A	B	C
1	Reference	Value	Comments
2	GO_PISURL_t1	http://ldpimfasiondevbunk.org:800/	RegressionEnvironment
3	GO_ExchangeRate_PHP	42.345	

Figure: Framework Components and their interaction.

FRAMEWORK COMPONENTS

There are many components in the hybrid automaton framework such as, Main Driver, Fabric Functions, Library Functions, Common Functions, Object Repository, and Reporting manager.

MAIN DRIVER:

The main driver is the one of the hybrid framework components which initiates the process of script execution. Main Driver itself is a script of few lines which invokes the process of synchronizing the keywords with the framework and the object repository.

This driver script calls the methods from the library, which reads the keywords, objects and parameters and performs the appropriate actions.

By incorporating the main driver with the framework in a separate function, the effort required to develop the main driver is reduced. A call to the function will perform the operation of main driver.

FABRIC FUNCTIONS:

This module consists of functions which forms the backbone of the framework. All the coding logic is in the form of java script. All these functions are stored in the function library.

LIBRARY FUNCTIONS:

Function library is the place where most of the scripts reside and the place where customization can be done in the script for the project. The function library is the only component in the framework that has to be changed in case the application is migrating from one platform to other.

COMMON FUNCTIONS:

Common functions are the functions which are reusable across all platforms. These functions are independent and do not depend on the technology used to develop the application.

Separating the common functions from the function library ensures maximum utilization of reusable scripts and in turn reduces the maintenance effort of scripts.

OBJECT REPOSITORY:

The Object Repository is the place where all related information such as properties, values are stored. With this Object repository the tool identifies the application under test and executes the business flows as per the functions in the test script.

If one or more of the objects property values in the application differs from the property values stored the object repository, tool will not identify the object and the script will fail. Therefore it is necessary to change the change the object repository if it differs from the property value in the application.

There are two types of object repositories to store the objects.

- Shared Object Repository
- Local Object Repository

In shared object repository the objects stored in a file that can be shared and accessed by multiple tests.

In Local object repository the objects stored in a file that is related with one specific action and only that action can access the stored objects.

A combination of both local and shared object repositories can be used. The best practice here is to keep a shared repository that can be used across all the test cases.

REPORTING MANAGER:

After execution of the test script, it is necessary to get the results of the execution. Apart from the test execution report generated by the automation tool, tester can customize reports in an external spread sheet format.

This provides report details for which test scripts have failed or passed while running a test suite. The reports detail exceptional cases handled, and error logged. These functions are customized in the driver script. This helps in performing effective analysis on the execution report.

CONCLUSION

Software Automation technique helps to achieve quality assurance, cost minimization, reliability, recoverability, reusability, security and improved performance in automating web based applications or in standalone applications. In order to test the application within a given time and to deliver the product on time automation techniques are used.

REFERENCES

1. Michael T. Grater Intermediate Methods Analyst United Parcel Service.
2. QTP Open Source Test Automation Framework Introduction Version 1.0.
3. www.Google.com
4. http://www.softwaretestinghelp.com
5. Armour, P. (2004, Oct). Not-Defect: The Mature Discipline of Testing.
6. Communications of the ACM, Vol. 47 Issue 10, p15, 4p.
7. Armour, P. (2005, Jan). The Unconscious Art of Software Testing. Communications of the ACM, Vol. 48 Issue 1, p15, 4p.
8. Asada, M., & Yan, P. (1998, May) Strengthening software quality assurance. Hewitt-Packard Journal, Vol. 49, Issue 2, p. 89-98.
9. Baer, T. (1997, Dec) Putting APPS to the test. Software Magazine, Supplement *Automated* Software, Vol. 17 Issue 14, p11,3p.