Test Automation using Keyword Driven Approach

[1]Santosh kumar Sharma, [2]Shivam Gaur
[1]Assistant Professor, Department of Computer Science and Engineering, Birla Institute of Technology,Mesra, Ranchi, Jaipur Campus, Jaipur

[2]B.E. Department of Computer Science and Engineering, Birla Institute of Technology,Mesra, Ranchi, Jaipur Campus, Jaipur

Abstract: *The goal of software testing is to find errors, and a good test is one that has a high probability of finding an error. Effective Testing produces high quality software.Testing can be conducted manually as well as automated. This paper presents the concept of automation and manual testing and problems with manual testing and benefits of automatic testing.We have also presented a process of Test Automation using Keyword Driven approach. The input in the form of natural language is converted to machine readable code snippet and that will be executed under framework to produce reports. Using this framework, we can improve reusability of automated test.The main objective of this research paper is to focus on, effectiveness and importance of automation testing.*

Keywords: *Software Testing, Manual Testing, Automation Testing, Keyword Driven Framework*

INTRODUCTION

High quality of software is must in these times when science and technology is at its peak.
Software is used everywhere these days. Theonly way to deliver a high quality of software is to perform high quality of software testing. Software testing is the task that can ensuremore reliable software.
Software testing involves experimentally and systematically checking the correctness of Software. It is a process that ensures the quality of the product to its stakeholders with information about the quality of the product or service under test.Testing demonstrates that software function appear to be working accordingly to specifications. It makes software defect free so user can easily access software and makes better use of that software to carry out operations. Testing improves the quality of software so maintenance cost is reduced to great extent. It improves reliability and efficiency of the software.Different types of software have different types of requirement. For example, hotel management software is completely different from banking software. The requirement of hotel management user is different from banking user. In such situations when an organization develops or invests in a software product it needs to ensure that the software product will be acceptable to its end users, its target audience, its purchasers, and other stakeholders. Software testing is the process of attempting to make this evaluation. Software can be tested either manually or automatically.

Following are the concepts related to testing:

1) *Test Data:* For testing some feature of the software you need to enter some data as input. Any such data which is used in tests are known as test data.
2) *Test Case:* Test case is a smallest unit of testing, a snippet containing set of inputs, test environment, execution preconditions and expected output. Inputs are the specific values, tables, database, or may be file names.
3) *Test Suite:* Test suite is common term used for the collection of test cases. It is a container of the set of tests that helps tester in executing and reporting the test execution status.
4) *Test Plan:* Test plan narrates the whole strategy that team will follow, for testing the final implementation. The task of test plan is to regulate all testing activities. It includes: What to test? What strategy/method will be used for testing? Who will test the software? When to test the software? What risks are present?

Testing Approaches:

1) *Manual Testing:*To create test cases manually and execute them without any tool support is known as manual testing. Manual software testing is performed by a human sitting in front of a computer carefully going through application screens, trying various usage and input

combinations, comparing the results to the expected behaviour and recording their observations.

2) Automation Testing: This type of testing uses an automation tool to execute test suite. Goal of automation is to reduce number of test cases to be run manually and not eliminate manual testing all together. Some automation tools are: Winrunner, Loadrunner, JUnit, Silktest etc.

MANUAL TESTING

Manual testing is the process of manually testing software for defects. It requires a tester to play the role of an end user, and use most of all features of the application to ensure correct behaviour.Large scale engineering projects that rely on manual software testing follow a more rigorous methodology in order to maximize the number of defects that can be found. Manual testing plays an important role in applications where functionalities change quite often.

Problems Associated with Manual Testing:

1) Time consuming and tedious: Since test cases are executed by human resources so it is very slow and tedious.
2) Huge investment in human resources: As test cases need to be executed manually so more testers are required in manual testing.
3) Less reliable: Manual testing is less reliable as tests may not be performed with precision each time because of human errors.
4) Non-programmable: No programming can be done to write sophisticated tests which fetch hidden information.
5) Manual Testing can become boring and hence error prone.
6) There is nothing new to learn when one tests manually.
7) People tend to neglect running manual tests.
8) None maintains a list of the tests required to be run if they are manual tests.
9) Manual Testing is not reusable.
10) The effort required is the same each time.
11) One cannot reuse a Manual Test.
12) Manual Tests provide limited Visibility and have to be repeated by all Stakeholders.
13) Only the developer testing the code can see the results.
14) In a typical manual test it is very difficult to test a single unit.
15) Scripting facilities are not in manual testing.

AUTOMATION TESTING

Automated tests execute a sequence of actions without human intervention. It is also defined as a testing of a system with different data sets again and again without intervention of human. Simply automated testing is automating the manual testing process currently in use. Automation is the use of strategies, tools, and artefacts that augment or reduce the need of manual or human involvement or interaction in repetitive or redundant tasks. It is the best way to increase the effectiveness, efficiency and coverage of software testing. Automation testing requires considerable amount of investment for buying the software & compatible hardware resources. It does what manual testing does not. Automation testing improves the accuracy & it saves the time of the tester & organization's money. It is best suited in the environment where the requirements are frequently changing & huge amount of regression testing is required to be performed. Automation testing is best suited in the environment where there are critical test cases that are to be executed repeatedly. It increases the quality of testing structure & reduces the future maintenance cost. Various benefits of Automation testing are fast run of test case. Reusable test cases are made & these test cases are reliable, comprehensive & Programmable.

Automating software testing involves developing test scripts using scripting languages such as Python, JavaScript or Tcl (Tool Command Language), so that test cases can be executed by computers with minimal human intervention and attention. The automation software can also enter test data into the system under test, compare expected and actual results and generate detailed test reports. Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required.Goal of automation is to reduce number of test cases to be run manually and not eliminate manual testing all together.

Benefits of Automation Testing:

1) *Fast:* It is faster than the manual testing.
2) *Cost Effective:* Test cases are executed by using automation tool so less tester are required in automation testing.
3) *Repeatable:* The same test case (record and replay) can be re-executed using testing tools.
4) *Reusable:* Test suits can be re-used on different versions of the software.
5) *Programmable:* Testers can program sophisticated tests that bring hidden information.

6) *Comprehensive:* Testers can build test suites of tests that cover every feature in software application.
7) *More reliable:* Automation tests perform precisely same operation each time they are run.
8) *Test Coverage:* Wider test coverage of application features.

TESTING FRAMEWORK

A testing framework is an execution environment for automated test. It is defined as the set of assumptions, concepts, and practices that constitute a work platform or support for automated testing. An organized test framework helps in avoiding duplication of test cases automated across the application. In short Test frameworks helps teams organize their test suites and in turn help improve the efficiency of testing.

The Testing framework is responsible for:

1) Defining the format in which to express expectations.
2) Creating a mechanism to hook into or drive the application under test
3) Executing the tests
4) Reporting results

Properties of a testing framework:

1) It is application independent.
2) It is easy to expand, maintain and perpetuate.

It is not possible to automate every test case. Hence testers must first decide which tests are to be automated. Generally the tests which can be reused or take longer time to perform are automated. In addition to this, testing all GUI items, connections with database, validations etc. can also be efficiently automated. Following factors are kept in mind while deciding to automate tests:

1. Products that need performing the same tests again and again.
2. Requirements of product do not change frequently.

Types of Frameworks:

1) *Linear Framework:* It is the simplest & basic framework. Just a single program for sequential steps in a test script is written. No modularity is present.
2) *Data driven Framework:* This type is used to test the behaviour of an operation with variable set of data.
3) *Keyword driven Framework:* In this type, set of keywords are defined. Main program is based on the functions that are invoked with the help of keywords as parameters. It is possible to develop the scripts without having programming knowledge.
4) *Modular Framework:* The Modularity testing framework is built on the concept of abstraction. This involves the creation of independent scripts that represent the modules of the application under test. These modules in turn are used in a hierarchical fashion to build large test cases. Thus it builds an abstraction layer for a component to hide that component from the rest of the application. Thus the changes made to the other part of the application do not affect that component.
5) *Hybrid Framework:* Hybrid testing framework is the combination of modular, data-driven and keyword driven testing frameworks. This combination of frameworks helps the data driven scripts take advantage of the libraries which usually accompany the keyword driven testing.

KEYWORD DRIVEN FRAMEWORK

Keyword driven testing is an application independent framework utilizing data tables and self-explanatory keywords to explain the actions to be performed on the application under test. Not only is the test data kept in the file but even the directives telling what to do which is in the test scripts is put in external input data file. These directives are called keywords. Keywords can be divided into base and user keywords. Base keywords are keywords implemented in the libraries. User keywords are keywords that are defined in the test data by combining base keywords or other user keywords. The ability to create new user keywords in the test data decreases the amount of needed base keywords and therefore amount of programming. The test scripts can be added, deleted and modified. The test script modification helps in the parameterization of the test and in dividing a test into multiple actions.

The generic flow of a Keyword Driven Framework is something like this -

1) Test Script or Driven Script calls the main function library. Library contains code to identify the keywords.
2) Function library checks the keyword and searches for the first function that matches to the keyword.
3) Then it calls the function associated with this keyword.
4) All the required actions are then performed by the function on the application that is being tested.

5) Control returns to the main which then reads the next keyword (step 2). Steps 2 to 5 are repeated until the functions related to all keywords are called.

Data Access module is responsible for data storage, including add scripts, modify scripts, read scripts, enquiry scripts, delete scripts, and other functions. The script has three levels.

Interface module is to enhance the framework's usability. It realizes a GUI interface, the graphical interface allows users to edit, drag and drop the modalities script; provides a user friendly guide to understand and use; provides view and editor which make it easily for users to view, modify the existing test scripts.

Support module consists of two parts: one is the libraries that all of the tests can share, including the log library and the test supporting library. The log library is responsible for providing the functions of log records to testers; the test supporting library provides the functions that all of the tests can be shared. The second is the testing library for GUI; this part provides the controls libraries.

Proposed Approach

To demonstrate the keyword driven framework we use the calculator program. Consider the basic functions of the calculator such as addition, subtraction, multiplication, division which are part of the Standard view.To demonstrate the keyword driven testing we take the actions performed by the mouse when making calculations. We create a table that maps the actions performed with the mouse on the window of the calculator application. In this table,

1) The windows column represents the application for which we are performing the mouse action.
2) The control column represents the control that we are clicking with the mouse.
3) The action column represents the action performed by the mouse.
4) The argument column contains the specific control value.

Our task is to i) Interpret the test cases written in natural language. ii) Create test scripts ready for execution. It is not easy to interpret natural language by computers so we will be focusing on basic parsing just to recognize keywords, actions and data. The test steps in keyword script will be despatched to our test application framework to generate test scripts automatically for final execution.

Window	Control	Action	Arguments
Calculator	Menu		View, Standard
Calculator	Pushbutton	Click	3
Calculator	Pushbutton	Click	-
Calculator	Pushbutton	Click	1
Calculator	Pushbutton	Click	=
Calculator		Verify Result	2
Calculator		Clear	
Calculator	Pushbutton	Click	6
Calculator	Pushbutton	Click	*
Calculator	Pushbutton	Click	5
Calculator	Pushbutton	Click	=
Calculator		Verify Result	30

After creating the table, we create a set of scripts for reading in the table, executing each step based on the keyword contained in the action field and log any relevant information.

The below pseudo code represents this test of scripts.

```
Main Script / Program
        Connect to data tables.
        Read in row and parse out values.
        Pass values to appropriate functions.
        Close connection to data tables.
Return
Menu Module Set focus to window.
        Select the menu pad option.
Return.
Pushbutton Module Set focus to window.
        Push the button based on argument.
Return.
Verify Result Module Set focus to window.
        Get contents from label.
Return
Compare contents with argument value.
        Log results.
Return.
```

Function calls will be generated from the keyword scripts those will produce test script. Test suite will contain various functions that are used while

creating test script. Test script is created in JavaScript. Automation framework will then execute it and will produce report for test cases.

CONCLUSION

This paper proposed the concept of testing and various testing approaches. It will help to understand test automation and its importance. In this paper, we also presented a technique to automate test automation using keyword driven framework. This is helpful to create test scripts and generate report automatically without human intervention using natural language input in the form of excel sheet keyword script.

REFERENCES

[1] Prof. (Dr.) V. N. MAURYA, Er. RAJENDER KUMAR, "Analytical Study on Manual vs. Automated Testing Using with Simplistic Cost Model", International Journal of Electronics and Electrical Engineering, January 2012, ISSN : 2277-7040.

[2] Vishal Sangave, Asst. Professor Vaishali Nandedkar, "A review on Automating Test Automation", International Journal of Advance Research in Computer Science and Management Studies, December 2014, ISSN: 2327782**.**

[3] Vishawjyoti, Sachin Sharma, "STUDY AND ANALYSIS OF AUTOMATION TESTING TECHNIQUES", Journal of Global Research in Computer Science, December 2012, ISSN-2229-371X.

[4] Rashmi, Neha Bajpai, "A Keyword Driven Framework for Testing Web Applications", International Journal of Advanced Computer Science and Applications, 2012.

[5] Vivek Kumar, "COMPARISON OF MANUAL AND AUTOMATION TESTING", International Journal of Research in Science And Technology, April 2012, ISSN: 2249-0604.

[6] Assistant Professor R. M. Sharma, "Quantitative Analysis of Automation and Manual Testing", International Journal of Engineering and Innovative Technology, July 2014, ISSN: 2277-3754

[7] "Software Engineering, A practitioner's Approach", Roger S. Pressman, Seventh Edition.

[8] Ayal Zylberman and Aviram Shotten, "Test Language: Introduction to Keyword Driven Testing". 2010, pp 1-7

[9] http://en.wikipedia.org/wiki/Keyword-driven_testing

[11] "Software Testing" Second Edition By: Ron Patton, Pearson Education

[12] www.softwaretestinghelp.com