Volume 7, Issue 4, April 2017 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Survey on Manual and Automation Testing strategies and Tools for a Software Application

Ritu Patidar, Anubha Sharma, Rupali Dave
Department of Computer Science, SVVV University, Indore,
Madhya Pradesh, India

Abstract: ***In the current Digital information age, Software industries, government organization and other IT industries has the priority of business across the world has focused on quality and reliability delivered software through process and technology. Quality of software is given with greater attention, which is resulting in reduction of software error in order to make quality software. In software engineering trend, software testing plays a significant role. Software testing is not a standalone process; it is series of steps for finding defects that is variance between expected result and actual result. This paper contributes the study which includes theoretical aspects about various software testing techniques, a discussion of different testing tools and its working by taking practical example. Through this paper, we analyse that how the test cases are prepared and applying in software application through manual and automation testing.***

Keywords— CMMI (Capability maturity model), RQM, RFT, TC (Test Case), selenium IDE.

I. INTRODUCTION

In recent years, there is tremendous enhancement in software enabled systems but the major concern is about software reliability and security. Software testing insures the reliability hence increases the customer confidence [3]. There are many open source and cost effective tools are available for improving software quality by reducing the software defects. Insures the reliability hence increases the customer confidence. There are many open source and cost effective tools are available for improving software quality by reducing the software defects .customer confidence. There are many open source and cost effective tools are available for improving Software quality by reducing the software defects .customer confidence. There are many open source and cost effective tools are available for improving software quality by reducing the software defects. Today's most of Research effort is done on software testing techniques but for faster evolution and software development this field is growing for improving software quality by reducing the software defects.

Most of the Research effort is done on software testing techniques but for faster evolution and software development this field is growing rapidly. Today's most of Research effort is done on software testing techniques but for faster evolution and software development this field is growing for improving software quality by reducing the software defects

Our case study shows the manual and automation testing methods and tools. This paper organizes as follows:-

Section 2 discusses about the methodologies and testing principles of software testing.

Section 3 shows the literature survey on various software testing techniques and tools.

Section 4 describes manual testing by giving an example of web application.

Section 5 gives a description of various automation tools such as selenium, rational functional tester and so on.

Section 6 Concludes the paper.

II. TESTING METHODOLOGIES

A. Black Box Testing

It is data driven testing technique that ignores the internal mechanism of a system or component and focuses on the output generated in response to selected input and execution condition.

B. Black Box Test Case Design Methods

Equivalence Class Partitioning: It is applicable only one condition in which the input data is dividing into classes.

Steps to be followed

- It divides the input domain of a program into classes of data.
- It derives test cases based on these partitions.
- An equivalence class represents a set of valid or invalid states for input condition.

Example

Let we take an example in which we apply black box testing on a webpage containing customer name and an account number.

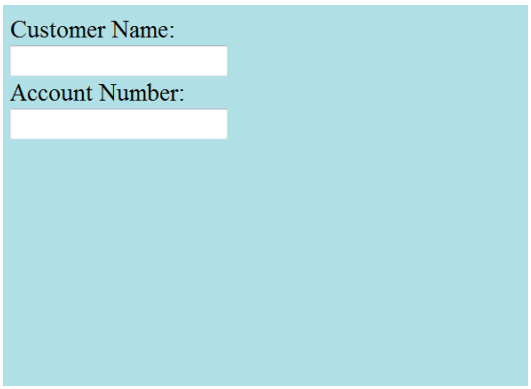

Figure 1: Web Page of Customer application

Table 1 Black Box Testing

Conditions	Valid Partitions	Invalid Partitions
Customer name should accept only 2 to 64 characters.	2 to 64 chars	< 2 chars and > 64 chars
Account number field should accept maximum 6 digits	Any non zero digit less than 6	< 6 and >6

C. Boundary Value Analysis

Many systems have tendency to fail on boundary. So testing boundary values of application is important.

- It is test functional testing technique where the extreme boundary values are chosen.
- It includes maximum, minimum, just inside and outside boundaries and error values.

Table 2 Boundary Value Analysis

Conditions	Valid Partitions	Invalid Partitions
Customer name should accept only 2 to 64 characters.	2 chars and 64 chars	1 chars, 65 chars, 0 chars
Account number field should accept maximum 6 digits	Zero and 6	Less than 0 i.e -1and greater than 6 i.e 7

D. Testing principles

- **Testing is event dependent**
  Testing is done differently in different situations (contexts). For example safety critical software is tested differently from an different perspective.
- **Extensive testing is impossible**
  Testing everything (all combination of inputs and prerequisite condition) is not feasible except for trivial cases.
- **Early Testing**
  Testing exercises ought to begin as ahead of schedule as could reasonably be expected in the product or system development life cycle and should be emphasis on define objectives.
- **Defect Clustering**
  A small number of modules contains most of the defects found during pre release testing or demonstrates the most operational failure.
- **Pesticide paradox**
  If the same test case repeated over and over again, eventually same set of test cases will no longer find any new bugs. To overcome this pesticide paradox the test cases needs to be regularly reviewed and revised to potentially find more bugs.
- **Testing shows presence of defects**
  Testing can demonstrate that defects are available, but cannot prove that there are no defects. Testing exercises ought to begin as ahead of schedule as could be expected under the circumstances in the software or system.

III. LITERATURE SURVEY ON VARIOUS SOFTWARE TESTING TECHNIQUES AND TOOLS

Adnan causevic and Daniel sundmark presents a survey on "An industrial survey on contemporary aspects of software testing [9]. This paper focuses on current practices and describes the aspects of software testing in an industrial scenario. The survey contains five categories of respondents such as agility of development process, domain of product, safety criticality of product, distribution performed by respondents. In this survey the author shows an industrial perspective seems to be test driven development.

Tara Astigarraga and Christana lara propose a methods based on "Current Curricula guidelines in software testing field" [13]. They concluded that a student is to be prepared for jobs in testing, curricula guidelines and available courses must be reformed. The courseware developed by IBM experts across the multiple phases of test is a part of an ongoing approach to bridge these skill gaps and target to achives the best education for undergraduate students in software development and testing field.

Shivkumar Hasmukhari trivedi presents a survey paper "Software testing techniques"[12]. In this paper various theoretical aspects of software testing techniques is elaborated. They described various testing tools and methodologies used at the time of testing a software. It also describe how the test plan template is created and defined.

P.K kapur, A. K shrivastav presents a survey "Release and testing stop time of software: A new Insight" [10].They concern about the optimal duration of testing. Here in this paper testing is divided into two phases Pre release and post release (before and after testing stop time). A generalized approach for optimal scheduling policy to minimizing overall testing ease. Numerical analysis included in the paper shows that if firm is providing software before its scheduled released time (without patching) and can provide option for post release testing which leads to reduce software testing cost. In future to increase reliability, the model extend to find optimal released and stop testing time within budget of software.

Lashand dukes and Xiaohong yuan proposes a paper "A case study on web application security testing with tools and manual testing"[7]. It describes case study on manual testing and presents comparative studies between manual and automation testing. According to testers observation and it is important to utilize a variety of tools as well as conduct manual testing in web application. Based on case study, manual testing is most important for improving the web security.

IV. MANUAL TESTING

Manual testing is the most established and most rigorous type of software testing. In this, software testers manually execute test cases without using any automation tools [2]. It requires a tester to perform manual test operations on the software application without the help of Test automation. Manual testing is the most primitive of all testing types and helps find defects in the software system.

A. Disadvantages of Manual Testing

- It is time consuming and tedious task.
- It requires a heavy investment in human resources.
- Time constraint often makes it impossible to manually test every feature of an application before released.

B. Test case

A test that tests the functionality of specific object. It is a description of what to be tested, what types of data to be given and what activities to be done to check the actual result against the expected result [1]. A test case has a component that describes an input, action or event and an expected response, to determine if a feature of an application is working correctly. The main purpose of writing test cases is to validate the testing coverage of the application.

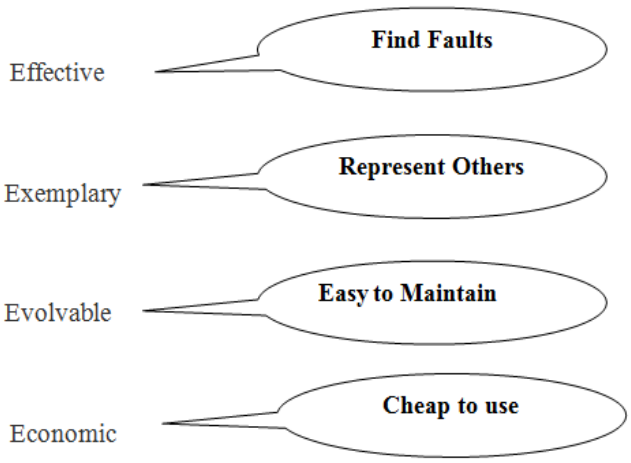

Figure 2: - Characteristics of good test cases

C. Structure of Test Cases

It contains the following parameters:-

- Identifier: A unique identifier of test case.
- Test Case Author /creator i.e Name of tester.
- Version of current test case.
- Name of the test case to identify test case purpose and scope.
- Type of testing.
- Short description of test case that is what functionality it checks.

Test Case Template

Table 3 Sample test case

Test Case ID	Test Case Name	Objective /Description	Pre-Requisite /Precondition	Steps to be Followed	Expected Result	Actual Result

Here, a student registration form in which we want to apply the manual testing procedure and write the test case for username and password field.

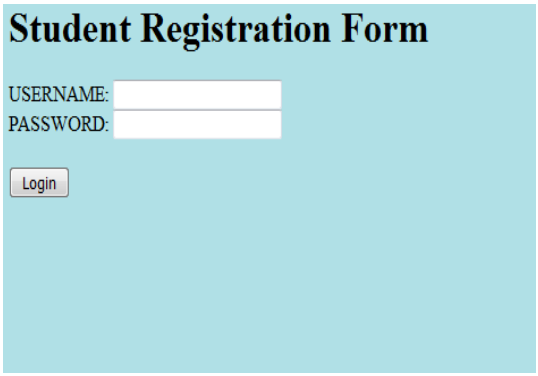

Figure 3: - Student registration form

Username

- This field accept only characters as a input data.
- It does not accept numbers, special characters and alphabetnumeric values.
- The size of username (length) should be 8 characters.

Password

- For password field, it should display password in encrypted format.
- The length of the password should be 10.
- Password should have compulsory 1 special character.
- Password should not be numeric.
- Only alphabets and 1 special character is allowed.
- Alpha numeric should not be accepted.
- Click on login button will allow user to login.

Table 4 Test Cases for Student Registration Form

Test Case ID	Test Case Name	Objective /Description	Pre-Requisite /Precondition	Steps to be Followed	Expected Result	Actual Result
TC-01	Username (availability)	To check whether username field is available on login page	1. Login page should open	Observed username field on login page	Username field should be available on login page	Username field should be available on login page
TC-02	Username (editibility)	To check whether username field editable on login page.	1. Login page should open. 2. Username field should be available.	Enter data in username field.	Username field should be available	Username field should be available
TC-03	Username (username should accept only character)	To check whether username should accept only character	1. Login page should open 2. Username field should be available. 3. Username field should be editable.	Enter data (character) in username field.	username should accept only character	username should accept only character

TC-04	Username (username should reject numeric values)	To check whether username should accept only character	1. Login page should open 2. Username field should be available. 3. Username field should be editable	Enter data(numbers) in username field.	username should reject numbers	username should reject numbers
TC-05	Username (username should not accept alphanumeric values)	To check whether username should not accept alphanumeric values	1. Login page should open 2. Username field should be available. 3. Username field should be editable.	Enter data (alphanumeric values) in username field.	username should reject alphanumeric values	username should reject alphanumeric values
TC-06	Username (username should accept maximum 8 characters)	To check whether username should accept maximum 8 characters	1. Login page should open 2. Username field should be available. 3. Username field should be editable.	Enter data in username field.	username should accept maximum 8 characters	username should accept maximum 8 characters
TC-07	Username (username should accept maximum 8 characters)	To check whether username should accept maximum 8 characters	1. Login page should open 2. Username field should be available. 3. Username field should be editable.	Enter data in username field.	username should accept maximum 8 characters	username should accept maximum 8 characters
TC-08	Username (username field should reject greater than 8 characters)	To check whether username field should reject greater than 8 characters	1. Login page should open 2. Username field should be available 3. Username field should be editable.	Enter data in username field.	username field should reject greater than 8 characters	username field should reject greater than 8 characters

D. Test case Execution Process

Execution and its results play a important role in the testing. Each and every action should have proof. The following activities should be taken as consideration while at the time of test case execution:-

- Total number of test cases executed.
- Total number of defects found.
- Screen shots of successful and failure executions should be taken in word document.
- Time taken to execute.

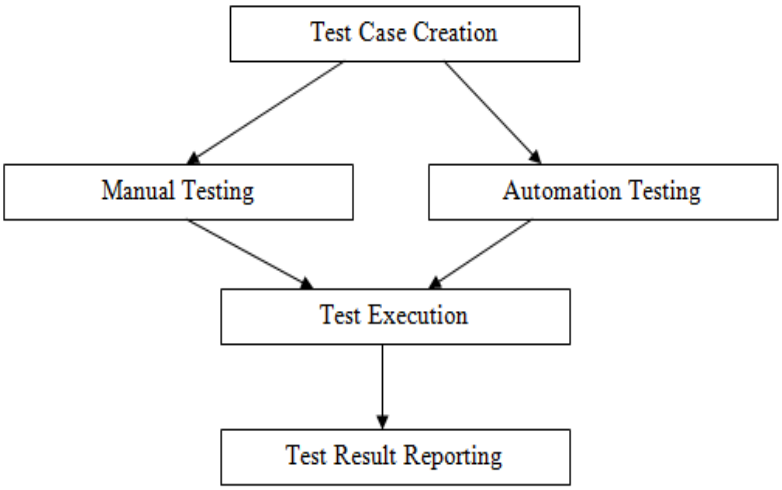

Figure 4: -Test case Execution Process

V. AUTOMATION TESTING

Every Organization has unique reason for automating software quality activities, but several reasons are common across industries. Automated testing tools are ability of executing tests, describes outcomes and estimate results with earlier test runs [5]. Tests completed with these tools can be run over and again at any time.

The procedure is being used to implement automation is called a test automation framework. Several frameworks have been developed over the years by commercial vendors and testing organizations. Successive development cycles will require execution of same test suite repeat many times. Using a test automation tool it's possible to record this test suite and re-play it as required.

Automation testing is important because

- It improves efficiency of testing.
- Reducing testing costs.
- Replicating testing across different platforms.
- To give consistent and accurate results.

A. Automation Tools

There are various tools that help software teams build and execute automated tests. Many software engineers are actively using unit tests as part of their development efforts to verify critical parts of their projects such as libraries, models and methods.Choosing an automated software testing tool is an important process. Generally a good testing tool should:

- It should test all functions in application software.
- It should have good debugging facilities.
- It should have clear help file and a user manual.

For improving the long term efficiency of software Automation testing has specific advantages team testing processes.

Table 5 Table of Various Testing Tools

Functional Testing Tools	Performance Testing Tool	Test Management Tool
Rational Functional Tester(RFT)	Load Runner	Rational Quality Mnager Manager
Win Runner	Apache JMeter (open source)	qTest (open source)
Win Runner	Apache JMeter (open source)	qTest (open source)
Quick Test Professional	Rational Performance Tester	PractiTest (open source)
Rational Robot(SQA)	Silk Performer	Test Collab
Silk Test	WebLOAD	HP ALM/Quality center
Selenium(Open source)		QMetry

B. Automation Testing Tools

Functional Testing Tools

- **RFT(Rational Functional Tester)**

IBM Rational Functional Tester is a tool for regression testing tool and automated testing of software applications from the Rational Software division of IBM. It allows users to create tests that mimic the actions and assessments of human tester.It Provides testers with automated testing capabilities for functional testing, regression testing, GUI testing and data-driven testing.

RFT is an object oriented , automated testing application that enables you to test java, HTML, VB.net and windows running application running on the following platforms:

- Microsoft windows XP, 2000.
- Microsoft windows server 2003
- Red hat Linux etc.

It is platform independent and browser independent test playback. It has different features:-

- **Storyboard testing: -** Simplifies test visualization and editing using natural language and rendered screenshots. Rational Functional Tester provides a visual, storyboard format for representing test actions.
- **Data-driven testing:** - Lets you perform the same series of test actions with a varying set of test data. Rational Functional Tester can automatically detect data entered during test recording and prepare the test for data-driven testing..
- **Test scripting**: - Combines a recorder of user actions with multiple customization options and intelligent script maintenance capabilities. Rational Functional Tester gives you a choice between either Java or Visual Basic.

Advantages

- RFT uses an object map between the script and the application under test. This means that when an application changes you don't need to find and replace object properties in the scripts. This will save you a lot of time.
- RFT uses data pools for driving large datasets into a test.
- RFT has very powerful connectors to multiple application styles like SAP, Siebel, Oracle, VB, PowerBuilder, various web and terminal server functions. This means one tool for every situation where selenium provides only one solution.

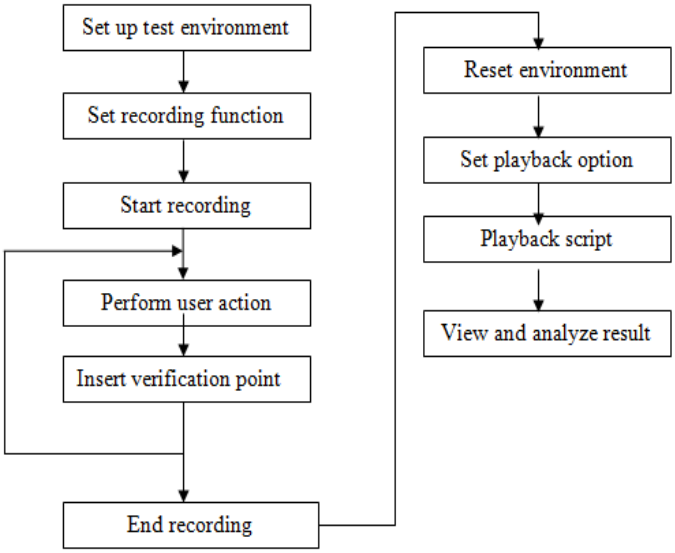

Figure 5:-Lifecycle of test cases execution

Selenium

In automation testing, we are using a software tool to run repeatable tests against the application to be tested. The most important advantage to test automation is that we apply same test case with more number of times fastly as compare to manual testing.

Selenium is most widely used open source tool. It is set of different software tools each with different approaches to supporting test automation. In selenium, the entire suite of tools and its operations are highly adaptable, allowing many options for locating UI elements and comparing expected test case results with actual behavior of application. It support for multiple platforms for executing the test cases [11].

Selenium Tool Suit

Selenium is composed of multiple software tools. Each has specific role.

Selenium RC

Selenium Remote Control (RC) was the flagship testing framework that allowed more than simple browser actions and linear execution. It makes use of various programming languages such as Java, C#, PHP, Python, Ruby and PERL to create more complex tests.

Selenium Web Driver

Selenium Web Driver is the successor to Selenium RC which sends commands directly to the browser and retrieves results.

Selenium Grid

It is a tool used to run parallel tests across different machines and different browsers concurrently which results in minimized execution time. Selenium grid allows running test in parallel that is different tests can run at the same time on different remote machines.

Selenium IDE

Selenium Integrated Development Environment (IDE) is a Firefox plug-in that lets testers to record their actions as they follow the workflow that they need to test. It is a prototyping tool for building test scripts. It is a Firefox plug-in and provides an easy to use interface for developing automated tests. It has recoding feature, which records user actions as they are performed and then exports them as a reusable script. It has save feature that allows users to keep the test in a table based format for later import and execution It provides a Graphical User Interface for recording user actions using

Firefox which is used to learn and use Selenium, but it can only support with Firefox browser as other browsers are not supported.

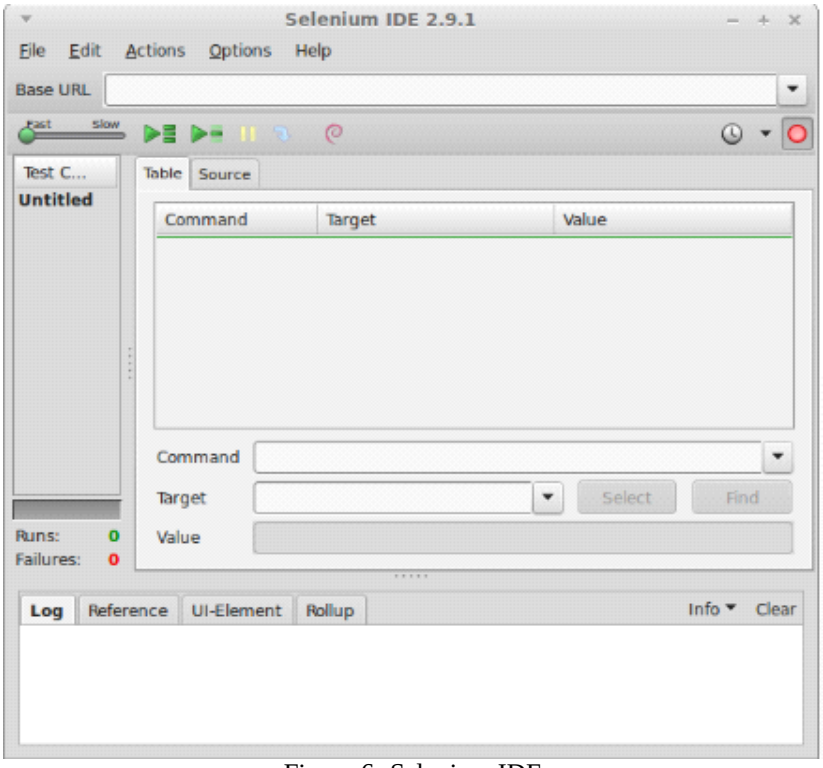

Figure 6:-Selenium IDE

Features of Selenium

Selenium Ide is implemented as a Firefox extension. It allows recording, editing and debugging tests.

- Selenium Recorder:-On start-up of the Firefox plug-in, the record feature is automatically turned on, allowing the user to record any action done inside the web page.
- In Selenium IDE scripts may be automatically recorded and edited manually providing auto completion support and ability to move commands around quickly
- Walk through tests
- Debug and set breakpoints
- Easy customization through plug-in.

Selenium Commands

Selenium commands called selenese are the set of commands that run your tests. A sequence of these commands is a test script.

- **Actions: -** These are generally used to manipulate the state of application. They do things like "click this link" and "select that option". If an action fails or has an error, the execution of the current test is stopped.
- **Assessors: -** These commands are used to examine the state of application and store the results in variables. For example "Store Title". They are also used to automatically generate assertions.
- **Assertions**: - These are like accessors, but they verify that the state of application conforms to what is expected. For example "make sure the page title is "X". Using base URL field at the top of selenium IDE window is very helpful for allowing test cases to run across different domains.
- **Selenium Buttons**
- 

Playback Speed. This controls the speed of your Test Script Execution
-  **Record.** This starts/ends your recording session. Each browser action is entered as a Selenese command in the Editor.
-  **Play entire test suite**. This will sequentially play all the test cases listed the test case pane.
-  **Play current test case**. This will play only the currently selected test case in the Test Case Pane.
-  **Pause/Resume**. This will pause or resume your playback.
-  **Step**. This button will allow you to step into each command in your test script
-  **Apply rollup rules**. This is an advanced functionality. It allows you to group Selenese commands together and execute them as a single action.

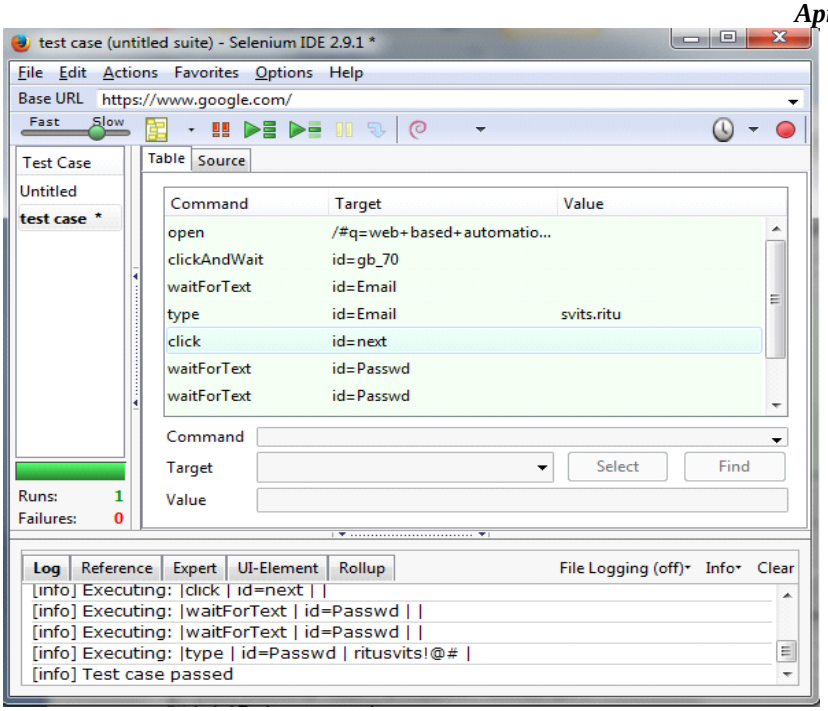

Figure 7:- Snapshot of Test cases through selenium IDE

Procedure Of Selenium IDE:-

- Firstly put the base URL on the base URL title bar, suppose named http://login .yahoo.com. Any test cases for these sites begin with an open statement should specify a relative url as the argument to open rather than an absolute url.
- Selenium IDE will then create an absolute URL by appending the open commands argument onto the end of the value of base URL.
- After putting the URL, We apply different commands on browser such assert title, clickandwait etc.
- After applying all the test cases, click on run button for executing the test cases.
- After successful execution of test case selenium tool shows the green signal otherwise failure test case represent with red signal.

Web Load

WebLOAD is an enterprise scale load testing tool which features a comprehensive IDE, Load Generation Console and a sophisticated Analytics Dashboard.

Applications

WebLOAD has built-in flexibility, allowing QA and Devloper teams to create complex load testing scenarios thanks to native JavaScripting. Web LOAD supports hundreds of technologies – from web protocols to enterprise applications to network and server technologies

Features

- Flexible test scenario creation
- Supports every major web technology
- Powerful correlation engine
- Automatic bottleneck detection
- Generate load on-premise or in the cloud

VI. CONCLUSION

Quality and reliability are the two most important parameter in a software development field. In this paper we describe the theoretical aspects of various software testing techniques. We analyze both the manual and automation testing, both are preferable in order to reduce bugs in software but different prospects such that if same test cases using repeatedly number of time, then instead of using manual testing we go through test automation to create scripts or programs. Through this paper we present an evaluation of our testing techniques with the help of example. In future, we believe that it is important to improve the research efforts in software testing field for improving the quality and efficiency of software and emphasis the enhancement of open source testing software.

REFERENCES

[1] Kasurinen, J., Taipale, O., Smolander, K., "Software Test Automation in Practice: Empirical Observations," Advances in Software Engineering, Volume 2010, pp 1-18.

[2] Lee, J., Kang, S., Lee, D., "Survey on software testing practices," IET Software, Volume 6, Issue 3, 2012, pp. 275-282.
[3] Mohamad Kassab ,"Software Testing Practices in Industry: The State of the Practice" 2016 IEEE
[4] Marcel Bohme and Soumya Paul,"A Probabilistic Analysis of the Efficiencyof Automated Software Testing", Ieee transactions on software engineering, vol. 42, no. 4, april 2016.
[5] M. Bohme and S. Paul, "On the efficiency of automated testing,"In, Proc. 22nd ACM SIGSOFT Int. Symp. Found. Softw. Eng., 2014,pp. 632–642.
[6] Kent C. Archie And Robert E. Mclear, "Environments for testing software systems",AT&Ttechnicaljournal.march/april1990.
[7] LaShanda Dukes, Xiaohong Yuan, Francis Akowuah, "A case study on web application security testing with tools and manual testing", IEEE 2013.
[8] Rasneet Kaur Chauhan and Iqbal Singh," Latest research and development on software testing techniques and tools", International journal of current engineering and technology, Vol 4, Page no 4 , Aug 2014
[9] Adnan Causevic, Daniel Sundmark, Sasikumar Punnekkat,"An industrial survey on contemporary aspects of software testing", IEEE 2010.
[10] P.K Kapur,A.K Shrivastava,"Release and testing stop time of software : A new insight, IEEE 2015.
[11] Rigzin Angmo, Monika Sharma ,"Slenium tool: Aweb based automation testing framework", International journal of emerging technologies in computational and applied science(IJETCAS), March-May, 2014, Page no-351-355.
[12] Shivkumar Hasmukhrai Trivedi," Software Testing Techniques", International Journal of Advanced Research in Computer Science and Software Engineering(IJARCSSE), Volume 2, Issue 10, October 2012.

