Automated Testing: An Edge Over Manual Software Testing

Jyotsna
CSE , Sharda University,
Noida, India

Mukul Varshney
CSE , Sharda University,
Noida, India

Shivani Garg
CSE , Sharda University,
Noida, India

Abha Kiran Rajpoot
CSE , Sharda University,
Noida, India

ABSTRACT

Software Testing is a process of finding errors while executing a program so that we get a zero defect software. It is aimed at evaluating the capability or usability of a program. Software testing is an important means of accessing quality of software. Complex systems are being built and testing throughout the software development cycle is valid to the success of the software. Testing is very expensive process. Manual testing involves a lot of effort, Measured in person per month. These efforts can be reduced by using the automated testing with specific tools.

Keywords: TET, RTH, Selenium, WaTin

I. INTRODUCTION

Software testing is the most critical phase of the Software Development Life Cycle. Software under test goes through various phases, which as per the study are test analysis; test planning, test case/data/environment preparation, test execution, bug logging and tracking and closure. There is lot of research which has been done in past to optimize overall testing process with intent of improving quality of software in a minimum amount of time. After evaluating all available testing processes it has been found that different development models are used for different types of applications and different testing techniques are performed to test the same. Software testing is also an important component of software quality assurance (SQA), and a number of software organizations are spending up to 40% of their resources on testing. There are four main objectives of testing.

Detection: Various errors, defects, and deficiencies are detected. System capabilities and various limitations, quality of all components, the work products, and the overall system are calculated

Prevention: In this information to prevent or reduce the number of errors, to clarify system specifications and system performance is provided. Different ways to avoid risks and to tackle problems in the future are identified.

Demonstration: It shows how the system can be used with various acceptable risks. It also demonstrates functions with special conditions and shows how products are ready for integration or use. Improving quality: By doing effective testing on software, errors can be minimized and thus quality of software is improved

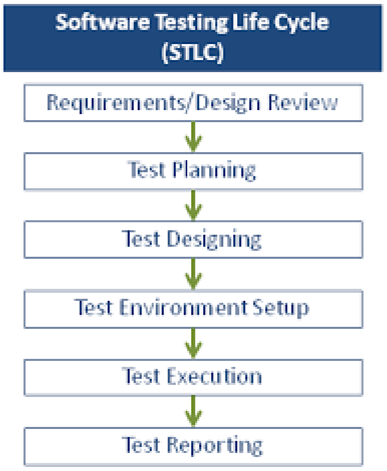
Software testing is the main activity of evaluating and executing software with a view to find out errors. It is the process where the system requirements and system components are exercised and evaluated manually or by using automation tools to find out whether the system is satisfying the specified requirements and the differences between expected and actual results are determined. Testing process can be performed by two ways that are manual or automation:

Manual testing is a process to test the software manually to find out the bugs. Manual testing is performed without using any automated tool. While performing the manual testing a test plan is used that describe the systematic and detailed approach of testing a software application. The goal of the testing is to make sure that the software application under test is defect free. Manual testing is not suitable for large projects as it requires more resources and time [1,2]. Automated testing is a process in which tools execute a pre defined scripted test on software to find defects.

Automated software testing is the finest way to increase the effectiveness and efficiency of software testing. Automation testing can does what manual testing does not. Automation testing also improves the accuracy and saves the time of the tester & organization’s money. It is best appropriate in the environment where the requirements are repeatedly changing & huge amount of regression testing is required to be performed.

II. Software Testing Process

STLC phases deals with detecting and rectifying any error by using various software testing techniques. This paper presents the required phases of testing lifecycle without which no software life cycle would be completed efficiently [3]. Testing basically furnishes a criticism or a comparison that determines the state behavior of the system against its specifications, mechanisms, principles, characteristics and relevant standards. Software testing process can be customized according to the customer or the project needs. The optimization process which one can use while testing software is analysis, planning and preparation, execution and closure. The software process provides the flow of the system and enhances the assurance of the product to be produced.



III. Software Test Automation Tools

Various types of tools are used for automated testing and they can be used in different areas of testing. The selection of tool is based on the type of application which we want to test like automated web testing tools, GUI testing tools. Selection of tools is totally based on the project requirements. The tools are divided into different categories as follows:

- Test Management tools
- Functional Testing Tools
- Load Testing Tools

a) Test Management tools

➢ TET (Test Environment Toolkit)

The goal behind creating the Test Environment Toolkit (TET) was to produce a test driver that accommodated the then current and anticipated future testing needs of the test development community. To achieve this goal, input from a wide sample of the community was used for the specification and development of TET’s functionality and interfaces.

➢ TETware

The TETware is the Test Execution Management Systems which allows you to do the test administration, sequencing of test, reporting of the test result in the standard format (IEEE Std 1003.3 1991) and this tools is supports both UNIX as well as 32-bit Microsoft Windows operating systems, so portability of this is with test cases you developed. The TETware tools allow testers to work on a single, standard, test harness, which helps you to deliver software projects on time[5]. This is easily available for download on ftp download.

➢ Test Manager

The Test Manager is an automated software testing tool is used in day to days testing activities. The Java programming language is used to develop this tool. Such Test Management tools are used to facilitate regular Software Development activities, automate & mange the testing activities.

➢ RTH

RTH is called as “Requirements and Testing Hub”. This is a open source test management tool where you can use as requirement management tool along with this it also provides the bug tracking facilities.

b) Functional Testing Tools

➢ Selenium

Selenium is an open source web testing tool which is used to test the web browsers across different platforms. It is divided into four components: First is, Selenium IDE which is used as a prototyping tool and no programming language is required. Second Selenium Remote Control that allow users to use the programming language. Third Web Driver which implement a stable approach by direct communication between the test scripts and browsers. Forth, Selenium Grid that helps to execute parallel tests on different browsers by using with Selenium Remote Control.

- **SoapUI**

SoapUI is the world's most widely-used open source API testing tool for SOAP and REST APIs. SoapUI offers SOAP Web Services functional testing, REST API functional testing, WSDL coverage, message assertion testing and test refactoring.

- **Watir**

Watir (Web Application Testing in Ruby, pronounced water), is an open-source family of Ruby libraries for automating web browsers. It drives Internet Explorer, Firefox, Chrome, Opera and Safari, and is available as a RubyGemsgem. Watir was primarily developed by Bret Pettichord and Paul Rogers.

- **WatiN**

WatiN stands for Web Application Testing In .NET, and it's a framework that enables web application testing through Internet Explorer. The way that WatiN works is very "easy", knowing that there's a lot of work behind a framework like this. WatiN lets you open Internet Explorer instances, and through interop, interact with the elements in a form.

- **Webcorder**

Webcorder is a freeware testing application which was developed in VB for allowing you to make web testing.

- **Solex**

Solex is a free open source Web application testing tool built as a plug-in for the Eclipse IDE. It provides functions to record a client session, adjust it according to various parameters and replay it later typically in order to ensure non regression of the application's behavior. Solex acts as an HTTP proxy and records all HTTP requests and responses going through the wire between a Web client (eg. a Web browser) and a Web server. The task of replaying a scenario consists in sending the previously recorded and eventually customized HTTP requests to the server and asserting each response.

- **Imprimatur**

Imprimatur is a tool for performing automated functional testing on web applications. The tests are described in a simple test script. Along with the standard HTTP methods, Imprimatur handles authentication, file uploads and HTTPS. The responses are validated using regular expressions. Imprimatur is a web application testing tool. The tests are described in a simple XML file. Along with the standard HTTP methods, Imprimatur handles authentication and file uploads. The responses can be validated using regular expressions; Imprimatur is designed to be the simplest web application functional testing tool that could possibly work.

- **WebInject**

WebInject is a free tool for automated testing of web applications and web services. It can be used to test individual system components that have HTTP interfaces (JSP, ASP, CGI, PHP, AJAX, Servlets, HTML Forms, XML/SOAP Web Services, REST, etc), and can be used as a test harness to create a suite of [HTTP level] automated functional, acceptance, and regression tests. A test harness allows you to run many test cases and collect/report your results. WebInject offers real-time results display and may also be used for monitoring system response times. WebInject can be used as a complete test framework that is controlled by the WebInject User Interface (GUI). Optionally, it can be used as a standalone test runner (text/console application) which can be integrated and called from other test frameworks or applications

c) Load Testing Tools

- **Jmeter**

Apache JMeter is open source software, a 100% pure Java desktop application, designed to load test functional behavior and measure performance of web sites. It was originally designed for load testing web applications but has since expanded to other test functions. Apache JMeter may be used to test functional and performance both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more). It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types.

➢ FunkLoad

FunkLoad is a functional and load web tester, written in Python, whose main use cases are:

- Functional testing of web projects, and thus regression testing as well.
- Performance testing: by loading the web application and monitoring your servers it helps you to pinpoint bottlenecks, giving a detailed report of performance measurement.
- Load testing tool to expose bugs that do not surface in cursory testing, like volume testing or longevity testing.
- Stress testing tool to overwhelm the web application resources and test the application recoverability.
- Writing web agents by scripting any web repetitive task.

IV. Conclusion

This paper presents a study on various automated testing tools that used on different platforms. Automation testing tools helps the tester to easily automate the whole testing process. Automation testing improves the accuracy and also save time of the tester as compared to the manual testing.

REFERENCES

1) Richa Rattan, “Comparative Study Of Automation Testing Tools: Quick Test Professional & Selenium”, Vsrd International Journal Of Computer Science & Information Technology, Vol. 3 No. 6 June 2013.

2) Manjit Kaur, Raj Kumar, “Comparative Study Of Automated Testing Tools: Test Complete And Quick Testpro”, International Journal Of Computer Applications (0975-8887) Volume 24- No. 1,June 2011.

3) Zhi Quan Zhou, Bernhard Scholz, Giovanni Denaro, “Automated Software Testing And Analysis: Techniques, Practices And Tools”, Proceedings Of The 40th Annual Hawaii International Conference On System Sciences (Hicss'07) 0-7695-2755-8/07

4) Manual Testing Http://En.Wikipedia.Org/Wiki/Manual_Testing

5) D. Gelperin and B. Hetzel, “The Growth of Software Testing”, Communications of the ACM, Volume 31 Issue 6, June 1988, pp. 687- 695.

6) A. P. Mathur, “Foundation of Software Testing”, Pearson/Addison Wesley, 2008.

7) IEEE Standard 829-1998, “IEEE Standard for Software Test Documentation”, pp.1-52, IEEE Computer Society, 1998.

8) Pressman, R.S. 1997. Software Engineering: A practitioner Approach.4th Edition. Tata McGraw Hill.

9) Sommerville, I. 1998. Software Engineering. 5th edition. Addison-Wesley.