MANTECH Publications

Journal of Software Engineering & Software Testing
Volume 7 Issue 3
ISSN: 2457-0516 (Online)

Estimating the Advantage of Automated Software Testing Over Manual Efforts

Dr. Alok Kumar[1], Rekha Prajapati[2], Suhani Kashyap[3]

Professor[1], Students[2, 3]

Department of CSE

CGC Technical Campus, Jhanjeri

Corresponding Author's Email id: *prajapatirekha@rediffmail.com[2]*

Abstract

Software testing is the act of identifying mistakes made while a programme is being run in order to produce software that has no faults. It aims to assess a program's functionality or usefulness. Software testing is a crucial tool for evaluating the calibre of software. Throughout the software development cycle, complex systems are constructed and tested, which is essential to the product's success. The procedure of testing is highly costly. Manual testing requires a significant amount of work, expressed in individuals per month. The use of automated testing and certain technologies can minimise these efforts.

Keywords*: - TET, RTH, Selenium, WaTin*

INTRODUCTION

The most important stage of the software development life cycle is software testing. Test analysis, test planning, test case/data/environment preparation, test execution, issue recording and tracking, and closure are some of the processes that software under test goes through. Numerous studies have been conducted in the past to streamline the entire testing procedure with the goal of enhancing software quality quickly. After analysing every testing method accessible, it was discovered that different development models are utilised for various application kinds, and various testing approaches are used to test the same. Software testing is a crucial part of software quality assurance (SQA), and many software companies devote up to 40% of their resources to it. The four major goals of testing are as follows.

Page 108-114 © MANTECH PUBLICATIONS 2022. All Rights Reserved

Detection: Numerous mistakes, flaws, and shortcomings are found. Calculations are made regarding the system's capabilities, different restrictions, and the quality of each component, the outputs, and the overall system.

Prevention: This information is given to clarify system specs and system performance as well as to prevent or decrease the amount of mistakes. There are several approaches to reduce future hazards and deal with issues.

Demonstration: It demonstrates how to use the system while taking a variety of reasonable risks. Additionally, it exhibits particular functionalities and explains how items are prepared for integration or usage. Enhancing quality Effective software testing may reduce mistakes, which raises the level of the program's quality. The primary task of analysing and running software with the intention of identifying flaws is called software testing. In order to assess if the system is meeting the defined needs and the discrepancies between expected and actual outcomes, the system requirements and system components are tested and reviewed manually or through automated techniques. There are two ways to carry out the testing process: manually or automatically: The process of manually testing software to detect problems is known as manual testing. Without utilising any automated technologies, manual testing is carried out. A test plan is used to define the methodical and comprehensive approach to testing a software application when carrying out manual testing. Making ensuring that the software programme being tested is free of errors is the aim of the testing process. Large projects shouldn't use manual testing since it takes more time and resources [1, 2]. To detect flaws in software, automated testing is a procedure in which tools run a programmed test that has been previously created. The best technique to improve the efficacy and efficiency of software is through automated testing. What manual testing cannot achieve, automation testing can. Additionally increasing accuracy, automation testing also saves the tester time and money. It works well in a situation when there is a lot of regression testing to be done and the requirements are continually changing.

SOFTWARE TESTING PROCESS

Through the use of various software testing approaches, errors are found and fixed during the STLC phases. In order for the software lifecycle to be successfully completed, the necessary steps of testing must be presented in this document [3]. In

Page 108-114 © MANTECH PUBLICATIONS 2022. All Rights Reserved

essence, testing provides a critique or comparison that establishes the state behaviour of the system in relation to its specifications, methods, principles, features, and pertinent standards. The software testing procedure can be tailored to the client's or the project's demands. Software testing may be optimised by using the steps of analysis, planning, preparation, execution, and closure. The software development process establishes the system's flow and improves the assurance of the final result.



Figure 1

SOFTWARE TEST AUTOMATION TOOLS

Automated testing utilises a variety of tool types that may be used to various testing scenarios. The sort of application we want to test will determine the tool we use, such as GUI testing tools or automated web testing tools.

Selection of tools is totally 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 System, which enables you to administer tests, sequence tests, and report test results in accordance with the IEEE Std 1003.3 1991 format. This utility is portable since it supports both UNIX and 32-bit Microsoft Windows operating systems. You may deliver software projects on schedule by using the TETware tools, which let testers operate on a single, standardised test harness [5]. This is readily accessible for download via ftp.

Page 108-114 © MANTECH PUBLICATIONS 2022. All Rights Reserved

➢ **Test Manager**

An automated software testing tool called the Test Manager is employed in routine testing procedures. To create this utility, the Java programming language was employed. These test management solutions are used to manage and automate testing processes as well as ordinary software development activities.

➢ **RTH**

The requirements and testing hub is known as RTH. This is an open source test management application that offers issue tracking capabilities in addition to being a tool for managing requirements.

b) FUNCTIONAL TESTING TOOLS

➢ **Selenium**

It consists of the following four parts: The first is Selenium IDE, a prototyping tool that doesn't need the usage of a programming language. User-friendly second Selenium remote control for the programming language. Third Web Driver, which uses direct connection between test scripts and browsers to achieve a reliable strategy. Selenium Grid, which works with Selenium Remote Control, enables the execution of tests in parallel across many browsers.

➢ **SoapUI**

The most popular open-source API testing tool for SOAP and REST APIs is called SoapUI. Functional testing for SOAP Web Services, REST APIs, WSDL coverage, message assertion testing, and test refactoring are all available through SoapUI.

➢ **Watir**

Watir (pronounced "water") is a series of open-source Ruby utilities for automating web browsers. It stands for Web Application Testing in Ruby. It is accessible as a RubyGems gem and powers Internet Explorer, Firefox, Chrome, Opera, and Safari. Bret Pettichord and Paul Rogers were mostly responsible for creating Watir.

➢ **WatiN**

Web application testing using Internet Explorer is made possible through a framework called WatiN, which stands for Web Application Testing in.NET. Despite the fact that a framework like WatiN requires a lot of work, the way it operates is incredibly "simple." WatiN enables you to launch instances of Internet Explorer and interact with form elements via interop.

Page 108-114 © MANTECH PUBLICATIONS 2022. All Rights Reserved

- **Webcorder**

A freeware testing programme called Webcorder was created in VB to enable you to create web tests.

- **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 (e.g., a Web browser) and a Web server. The task of replaying a scenario consists of sending the previously recorded and eventually customised HTTP requests to the server and asserting each response.

- **Imprimatur**

Imprimatur is a tool for doing functional testing on web applications in an automated manner. An easy test script outlines the tests. Imprimatur handles HTTPS, file uploads, and authentication in addition to the common HTTP protocols. Regular expressions are used to verify the replies. Imprimatur is a tool for testing web applications. In a simple XML file, the tests are described. Imprimatur manages file uploads and authentication in addition to the typical HTTP methods. Imprimatur is intended to be the most basic functional testing tool that may possible operate; replies can be checked using regular expressions.

- **WebInject**

A free tool for automated testing of online services and apps is called WebInject. It may be used as a test harness to build a set of [HTTP level] automated functional, acceptability, and regression tests, and it can be used to test specific system components that have HTTP interfaces (JSP, ASP, CGI, PHP, AJAX, Servlets, HTML Forms, XML/SOAP Web Services, REST, etc.). With the aid of a test harness, you may execute several test cases and record or present your findings. Real-time results may be viewed via WebInject, which can also be used to track system response times. The WebInject User Interface of WebInject may be used as a full-featured test framework (GUI). It may also be used independently as a test runner (text/console programme), which can be incorporated into and called from other test frameworks or programmes.

Page 108-114 © MANTECH PUBLICATIONS 2022. All Rights Reserved

c) LOAD TESTING TOOLS

➢ **Jmeter**

Apache JMeter is a desktop tool that is open-source, 100% pure Java, and created to load test the functionality of websites. Although it was first created for web application load testing, it has recently been expanded to include various test procedures. To evaluate the functionality and performance of static and dynamic resources, utilise Apache JMeter (files, servlets, perl scripts, Java objects, data bases and queries, FTP servers, and more). To test a server's durability or to evaluate overall performance under various load types, it may be used to simulate a large demand on a network or other object.

➢ **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.

CONCLUSION

This research of several automated testing technologies used on various platforms is presented in this report. Testing automation technologies make it simple for the tester to automate the whole testing process. Compared to manual testing, automation testing increases accuracy and reduces testing time for the tester.

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.

Page 108-114 © MANTECH PUBLICATIONS 2022. All Rights Reserved

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.

Page 108-114 © MANTECH PUBLICATIONS 2022. All Rights Reserved