Web based Automation Testing and Tools

Monika Sharma, Rigzin Angmo

Department of Information Technology, University Institute of Engineering & Technology,
Panjab University Chandigarh (U.T.), India

***Abstract*—Software testing is an essential phase of software development life cycle. Today many software applications are written as web based application that runs in an Internet Browser. The economic relevance of web based application increases the importance of controlling and improving its quality. The quality assurance of a system depends upon automation testing that decreases the test cost and increases work efficiency. In this paper we have discussed various web automation testing tools which will help us to understand the automation testing as well as the tools available for automation testing. A variety of web based systems and applications are tested by automation testing tools. The automation testing script is used in test automation. To choose the best tool for a task, various issues like ease of integration should be considered and weighed against the cost and performance. Also the tool needs to be compatible with the design and implementation of an application.**

***Keywords*—Web automation testing, automation testing tools, web application, software testing type, software quality.**

I. Introduction

Software testing is one of the most important and crucial phase in the software development life cycle process, consuming an average of 40% to 70% of software development process [1]. Software testing is a method, which is used for assessing the functionality of a software program. Software testing is also a process through which software item is evaluated to detect the defects and then correct them. It is also used to assess the feature of a software system and as well as the quality of the product. The software testing should be done during the development process. In other words software testing is stated as verification and validation process that a computer program/application/product should:-

- Meets the requirement that is used in software designing and development.
- Can be implemented with the same characteristics.
- Works as expected and satisfies the needs of stakeholders.

Software Testing is very important because software bugs can potentially cause monetary as well as human losses. In simple words, software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect free.

II. Type Of Testing

The software testing is a very wide topic. It includes varies methods, types and levels or stages of testing. The basic testing methods are elaborated below (in fig. 1.1(a)):-

- Static testing and Dynamic testing
- The box approach method
- Manual and Automation Testing

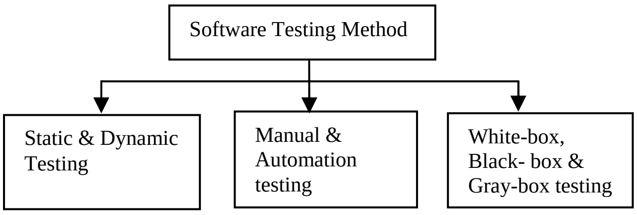
Software Testing Method
Static & Dynamic Testing
Manual & Automation testing
White-box, Black- box & Gray-box testing

Fig. 1.1(a)

A. Static and Dynamic Testing

This type of testing method can be illustrated with the help of difference (TABLE I below):-

TABLE I. Static vs. Dynamic Testing

S. No.	Static Testing	Dynamic testing
1.	Static Testing is done without executing the program	Dynamic Testing is done by executing the program
2.	It is performed before compilation	It is performed after compilation
3.	Static testing is about prevention of defects	Dynamic testing is about finding and fixing the defects
4.	It s less expensive	It is more expensive
5.	This testing requires loads of meetings	This testing comparatively requires lesser meetings
6.	Static Testing is a verification process	It involves validation process
7.	Static testing involves checklist and process to be followed	Dynamic testing involves test cases for execution

B. The Box Approach:

Traditionally software testing is divided into white- and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

TABLE II. Compares various features of black Box Testing And white box Testing

S. No.	Black Box Testing	White Box Testing
1.	Also known as closed box testing, data driven testing and functional testing	Also known as clear box testing, structural testing or code based testing
2.	The Internal Workings of an application are not required to be known	In this testing Tester has full knowledge of the Internal workings of the application
3.	Testing is performed by end users, testers and developers	Normally done by testers and developers
4.	Testing is based on external expectations as Internal behaviour of the application is unknown	Internal workings are fully known and the tester can design test data accordingly
5.	Not suited to algorithm testing	Suited for algorithm testing
6.	This is the least time consuming and exhaustive.	This type of testing is the most exhaustive and time consuming
7.	This testing can only be done by trial and error method	Data domains and Internal boundaries can be better tested with the help of this testing

Also there is another box approach is available known as *Grey Box Testing.* This type of testing is also known as translucent testing as the tester has limited knowledge of inside the application. It has the mixed features of white and black box testing approach. The testing is performed by end user, testers and developers. Testing is done on the basis of high level data diagrams and data flow diagrams. It is partly time consuming and exhaustive.

C. Manual And Automated Testing

Manual Testing: In manual testing type the software are tested manually i.e. without using any tool.

Automation testing: It is also known as test automation. In this the tester writes scripts and uses another software or tool to test the software. It is basically a process that involves automation of a manual process. The difference between manual and automation testing can be clearly specified with help of following table III:

TABLE III. Compares Manual and Automation Testing

S. No.	Manual Testing	Automation Testing
1.	The execution speed of Manual testing is slow as test is executed by human resources so it is Time consuming and tedious as well	Automation testing executes test cases significantly faster than human resources.
2.	Automation testing executes test cases significantly faster than human resources.	Test cases are executed by using automation tool so less testers are required in automation testing. As a result Less investment is required in human resources
3.	In manual testing no programming can be done to write complicated tests which fetch the hidden information.	Automation testing programmable in this testers can program sophisticated tests to bring out hidden information.
4.	Manual testing is less reliable as tests may not be performed with precision each time because of human errors.	Automation testing is more reliable and less error prone

There are many others type of software testing are available, in fact any type of software testing type can be executed both manually as well using an automation tool[1]. The other software testing types under above two testing methods are: Unit testing, Integration testing, System testing, Load testing etc. As illustrated in fig. 1.1(b) below:

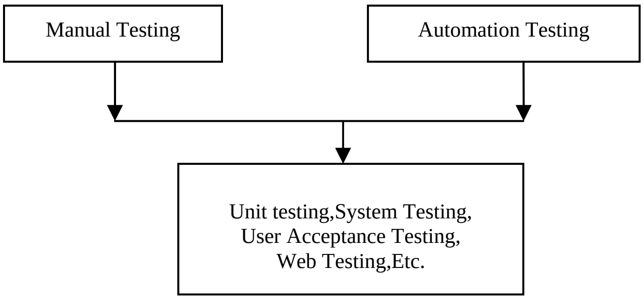

Figure 1.1(b)

Fig. 1.1(b)

III. EVOLUTION OF WEB AUTOMATION TESTING TOOL

The test automation is not at all new but the automation testing has been around the day one of the computing industry. In 40's Programmers used to test the software and not the testers and testing was often not distinguished from debugging as well. Throughout the sixties, papers about testing, such as the proceedings of the IFIPS conferences, almost assumed that programmers tested the software they built. As larger and more complex systems arose, the idea of software testers came into trend. The first conference on testing software was organized in 1972 at Chapel Hill where the proceedings of the conference presented that the discipline of testing is distinct from programming. The automation of technology allows us to share information limitlessly. Several researchers used the web automation testing tools in their research work. The researchers also enhanced some of tool for better performance and results. These tools are applied independently as well as combined to solve the challenges in software testing. Here, we discuss work done on web automation testing with the use of automation testing tools.

Leott et al. [2] presents an Industrial case study about test automation & test suite maintenance in context of Learning Content management System. In this author proposed realignment effort, which conclude that ID based method for locating web page elements that is better than Xpath methods. The goal of author work is to understand which Selenium WebDriver method is suitable for locating web page element which reduces the maintenance effort needed to realign the test cases to a new release.

Leott et al. [3] also presents another report on an industrial case study which is done to investigate the potential benefit of adopting the page object pattern to improve maintainability of selenium webDriver test cases. The paper compares the two equivalent test suits, one is built using the page object pattern and one is built without it. The results will indicate a strong reduction of time and number of modified LOCs to repair the test suite when the page object pattern is used.

Cheluvaraju et al. [4] proposed a novel method to investigating relationship between the files that are committed together by applying advance SNA (social network Analysis) to a network. In this paper they performed the empirical analyses on revision histories of well known open source web application testing system, selenium and results parameters like extracting cross – language change dependencies, change propagation, impact analysis are reported.

Wong et al. [5] presented an approach to mining an executable specification from given Selenium IDE test suite. There approach evaluate by number of test suits for real world system. There work also demonstrates integration of the two opposite processes of model mining and model based test generation in one framework.

Nagowah et al. [6] proposed a design of an automated testing tool, Kishanium tool by analyzing the existing tools. Kishanium tool is an innovative tool with improvement on the record and playback approach. The tool manages the set of test data and also reuses existing test data even if there are number of changes in the user interface. The tool provides additional features like Data generator, spylink, Snapshot.

Wang et al. [7] introduced a new automation framework integrated by selenium and Jmeter. This automation framework shares the test steps and test data which is convenient to switch in various types of testing for web application. It supports multiple browser and operating System. With use of this software framework one can efficiently improve the extensibility and reusability of automation test.

IV. Web Automation Testing And Tools

A. Web testing

Web testing is the name given to the software testing that focuses entirely on web applications. This helps to cut down the cost, minimize the efforts required to test web applications as well as web sites, increase software quality, reduces time-to-market and uses reusable test cases. There are various web testing available like: Functionality Testing, Compatibility Testing, Localized Testing, Stress Testing, Load Testing, Web Services Functional Testing, Web Services Performance Testing, Regression Testing etc.

B. Web automation testing

In the fast changing and highly competitive web-based business environment, it is critical for organizations to test their web sites and web applications using a manual testing tool. So an automated web testing should ensure that the web applications/web sites/web services usual functionality works correctly, provides the ability to reuse and extend the tests across multiple browsers / platforms/ languages / databases / servers and ensure that all the users accessing the web applications get results in an acceptable time.

C. Features of web automation testing:-

- Automated Software Testing Saves Time and Money
- Testing Improves Accuracy
- Increase Test Coverage
- Automation Does What Manual Testing Cannot
- Automated QA Testing Helps Developers and Testers
- Team Moral Improves

D. Web automation testing tool

To choose the best tool for the task, there are a lot of things that need to be considered. First thing is the ease of integration and should be weighed against the cost and performance i.e. how it performs in the environment with the network traffic and hardware. Also the tool has to be compatible with the design and implementation of application. There is different type of tool available in the market. We will discuss here at least 10 best web automation testing tool and their purpose (TABLE IV):-

- Selenium
- HP –QTP
- FitNesse
- Watir
- testComplete
- LoadRunner
- TestNg
- TOSCA
- SilkTest
- WinRunner

1.) Selenium: Selenium is not just a single tool but a suite that comprises of 4 tools[1]: selenium IDE, selenium RC, selenium Webdrive, selenium grid. Selenium is a portable and an open source automated testing suite that is built for web applications supporting different browser and application. Selenium was developed by Jason Huggins in 2004 who working in ThroughtWorks which is a privately owned global software delivery and products company. Ever since then Selenium has been making a tremendous improvement and has come out as a strong tool for web based automation testing.

2.) HP –QTP: QTP is the part of HP quality centre tool suite[2]. It provides regression and functional testing automation for major software environment and applications. It was originally written by Mercury Interactive[4], which was acquired by HP (Hewlett Packard) in 2006.

3.) FitNesse: Fitnesse is an automation testing tool to write, or organize and execute table based tests. Fitnesse is wiki wrapper over the fit sever that provides a web interface to the test suite. It is based on Ward Cunningham's Framework for Integrated Test. FitNesse is a tool for enhancing collaboration in software development[3]. Fitnesse enables customers, testers, and programmers to discover what their software should do, and to automatically compare that to what it actually does. Fitnesse compares customers' expectation to actual result. It is also compatible with Junit.

4.) Watir: Watir is pronounced as water. It is acronym for web application testing in ruby. Watir is an open source family that uses ruby libraries for automation web browsers[4]. Watir allows testers to write tests that are easy to read and maintain. Watir is simple and flexible too.

5.) TestComplete: TestComplete is an automated testing tool which allows testers to create, manage and run tests for any Windows, Web or Rich Client software. Testcomplete is developed by SmartBear Software[5]. The aim of this automated tool is to create software quality tests.

6.) LoadRunner: HP LoadRunner is an industry standard based an automated performance and test automation product from HP for load testing of application: that examines system performance and behaviour. HP LoadRunner works by using the virtual users. It also simulates thousands of concurrent users to put the application through various real life user loads and analyses the results in detail to discover the particular behaviour.

7.) TestNg: TestNG refers to "Testing, the Next Generation", is a testing framework that is inspired from JUnit[6] and NUnit[7]. It has also introduced some new functionality that makes it more powerful and easier for testers.

8.) TOSCA: Tosca tool is a testsuite for the automated execution of functional and regression software testing. TOSCA testsuite includes integrated test management, design, execution and data generation toolset for functional and regression tests.

9.) SilkTest: SilkTest is an automation tool that is specifically designed for regression and function testing. Originally SilkTest is developed by Segue Software Inc. which was acquired by Borland in 2006 and later Borland was acquired by Micro Focus International in 2009. The SilkTest provides the flexible and robust test scripting language that is a built in recovery system for unattended testing, and silkTest has the ability to test across multiple platforms, browsers and technologies. Silk Test offers test planning, validation, management, and direct database access.

10.) WinRunner: HP WinRunner software is an automation Functional GUI testing tool. This tool allowed User to capture, verified and replayed UI interaction as test script. WinRunner is originally written by Mercury Interactive.

TABLE IV. Web Automation Testing Tools

S. no	Name of tool	Language Use	Operating system	Type	Stable / initial release / development year	Lice-nse	Development status	Developer	Language supported	Browser supports
1	**Selenium**	Java	Cross - platform	Software testing framework for web application	2.33/May 22, 2013, (2004)	Apache 2.0	Active	Jason huggins & later joined other programmer and tester of thoughtwork	Domain specific language	All major browser
2	**Watir**	Ruby	Cross - platform	Software testing framework for web application	4.0/ September 30, 2012	BSD	Active	Bret Pettichard & Paul Rogers	Java, .NET, c#	Originally only for Internet Explorer, later supports multiply browser
3	**HP-QTP**	VB Script	Microsoft Windows	Test Automation	11.0/ HP acquire by Mercury Interactive in 2006, (now 11.51 & 11.52)	Pro-prietary	Active	Originally by Mercury Interactive and later by HP Software Division	VBscript	IE 6,7,8,10, Firefox 3.0 and later, QTP 11.51 provides reply support for Google chrome browser.
4	**Test Complete**	Java	Microsoft windows	Test Automation Tool	9.31/ September 3,13; initial release in 1999 by automated QA	Proprietary	Active	SmartBear Software In.	VBscript, Jscript, C++, Delphi Script, c#Script.	IE, Firefox, Google chrome
5	**FitNesse**	Java	Cross - platform	Test Automation Tool	V20130530/ May 30, 2013	CPL	Active	Rober C, Martin & other	C++, Python, Ruby, Delphi, c# etc	Platform independent
6	**HP Load Runner**	C	Microsoft windows & Linux (load generator only)	Load Testing Tool	11.52/ first vesion in 1989	Pro-prietary	Active	Hewlett-Packard, HP Software Division	VB, VBscript, java, javaScript, c#	Any browser
7	**TestNG**	Java	Window, Linux, MAC	Testing Framework	All versions are stable, current version-6.8.1, /April 27th, 2004	Apache 2.0	Active	Cedric Beust	Java, also include more object-oriented feature	IE, FireFox, chrome
8	**TOSCA**	C#, java, VB6	Microsoft windows	Test Automation	8.1/October 2013	Pro-prietary commercial Software	Active	TRICENTIS Technology & Consulting	Delphi, .NET including WPF, java swing/SWT/ AWT visual Basic	IE, FireFox
9	**Silktest**	4Test Scripting Language	Microsoft Windows	Test Automation	13.5/ October 10, 2012	Pro-prietary	Active	Microfocus International, originally developed by Segue Software	Java, 4Test, VB, c#, VB.net	IE, FireFox
10	**Win Runner**	C	Microsoft Windows and Linux	Load Testing Tool	11.52	Pro-prietary	Active	HP Software Division	Test Scripting Language (TSL)	IE, Netscape

V. Conclusion

In this paper we have analysed different automated tools for web testing. Web based automation testing is concerned with testing web based applications in an automated way. A variety of automation tools are feasible for testing web applications. A popular way of testing web sites is using Selenium tool, which can record your behaviour as you use the web site and then play back the steps automatically in your web browser. The main benefit of using automated tools is that you can avoid the manual effort required to test each feature of your web site by automating the tests. From the study on web based automation testing and the automated tools for this type of testing, we conclude that Selenium is the best available automation tool for web applications till now.

References

[1] A. Gargantini and E. Riccobene, "ASM-Based Testing: Coverage Criteria and Automatic Test Sequence," *J.UCS: Journal of Universal Computer Science*, pp. 1050–1067, 2001.

[1] M. Leotta, D. Clerissi, F. Ricca, C. Spadaro, "Repairing Selenium Test Cases: An Industrial Case Study about Web Page Element Localization," in *Proceedings of the Sixth International Conference on Software Testing, Verification and Validation*. IEEE Computer Society Press, 2013, pp. 487-488.

[2] M. Leotta, D. Clerissi, F. Ricca, C. Spadaro, "Improving Test Suites Maintainability with the Page Object Pattern: An Industrial Case Study," in *Proceedings of the Sixth International Conference on Software Testing, Verification and Validation Workshops*. IEEE Computer Society Press, 2013, pp. 108-113.

[3] B. Cheluvaraju, K. Nagal, A. Pasala, "Mining Software Revision History using Advanced Social Network Analysis," in *Proceedings of the 19th Asia-Pacific Software Engineering Conference*. IEEE Computer Society Press, 2012, pp. 717-720.

[4] D. Xu, W. Xu, B. K. Bavikati, W. E. Wong, "Mining Executable Specifications of Web Applications from Selenium IDE Tests," in *Proceedings of the Sixth International Conference on Software Security and Reliability*. IEEE Computer Society, 2012, pp. 263-272.

[5] L. Nagowah and K. Doorgah, "Improving Test Data Management in Record and Playback Testing Tools," in *Proceedings of the International Conference on Computer & Information Science (ICCIS)*. IEEE Computer Society, 2012, pp. 931-937.

[6] L. Nagowah and K. Doorgah, "Improving Test Data Management in Record and Playback Testing Tools," in *Proceedings of the International Conference on Computer & Information Science (ICCIS)*. IEEE Computer Society, 2012, pp. 931-937.

[7] F. Wang, W. Du, "A Test Automation Framework Based on WEB," in *Proceedings of the 11th International Conference on Computer and Information Science*. IEEE/ACIS, 2012, pp. 683-687.

[8] R. A. Razak, F. R. Fahrurazi, "Agile Testing with Selenium", in *Proceedings of the* 5th Malaysian Conference in Software Engineering (MySEC), IEEE Computer Society Press, 2011, pp. 217-219.

[9] X. Wang, P. Xu, "Build an Auto Testing Framework Based on Selenium and FitNesse", in *Proceedings of the* International Conference on Information Technology and Computer Science, IEEE Computer Society, 2009, pp. 436-439.

[10] F. Cannizzo, G. Marcionetti, P. Moser, "Evolution of the Tools and Practices of a Large Distributed Agile Team", in *Proceedings of the* Agile 2008 Conference, IEEE Computer Society, 2008, pp. 513-518.

[11] B. Haugset, G. K. Hanssen,"Automated Acceptance Testing: a Literature Review and an Industrial Case Study", in *Proceedings of the* Agile 2008 Conference, IEEE Computer Society, 2008, pp. 27-32.

[12] A. Holmes, M. Kellogg," Automating Functional Tests Using Selenium", in *Proceedings of the* AGILE 2006 Conference (AGILE'06), IEEE Computer Society, 2006.