International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 07 | July - 2022 Impact Factor: 7.185 ISSN: 2582-3930

COMPARATIVE ANALYSIS OF TOOLS AND TECHNIQUES FOR WEB-BASED AUTOMATION TESTING

Jennifer Preetam Kumar

Department of MCA, Vivekanand Education Society's Institute of Technology, University of Mumbai, Mumbai, India

Abstract: The life cycle of software enhancement must include software testing. Nowadays, a lot of software apps are created as web programs that operate on a web browser. The financial significance of a web application raises the need of monitoring and enhance its quality. Automation testing, which lowers test costs and boosts work efficiency, is used to confirm a system's quality. Automation testing tools are used to evaluate different web-based systems as well as applications. Test automation makes use of the automation testing script. To choose the optimal instrument for work, several factors including integration ease must be considered and balanced against cost and execution. The tools should also work with an application's planning and execution. In this paper, I've covered a variety of testing methodologies, with a focus on web automation testing technologies that may help us better understand automated testing and the tools available for it. Also, analyze the tool that is best fit for web-based application testing.

Keywords: Automation testing tools, software quality, web automation testing, software testing type, web application.

1. Introduction

Software testing, which accounts for 40% to 70% of the whole software development procedure, is one of the most crucial stages. It is a method for evaluating the utility of a software application. Software testing is an additional interaction used to evaluate software to find faults and fix them. It is also used to evaluate the product's quality and the crucial element of a software system. The software testing is will reduce the risk of failure of the product when executed in the development phase. It can ultimately be described as the validation and verification procedure that a computer software, product, or application must:

Fulfill the criteria utilized in the design and development of software.

- May be carried out with similar qualities.
- Functions are true to form and fulfill the necessities of stakeholders.

Software testing is essential because errors in software may result in losses in both financial resources and human lives. In other terms, software testing is a work carried out for ensuring that the software program is error-free and to determine if the real outputs correspond to the anticipated results.

2. Category of Testing

Software testing is a broad theme. It combines many testing techniques, approaches, and stages or levels. The following explanations of the important testing techniques are included in Figure 1.1(a): -

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

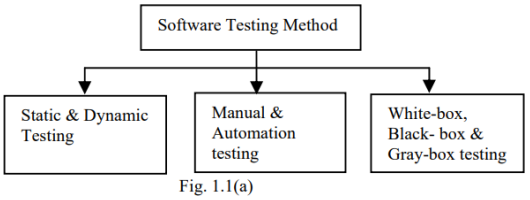

Fig. 1.1(a)

A. Dynamic and Static Testing

Such testing technique may be delineated with the assistance of distinction (TABLE I beneath): -

TABLE I. Comparing Static and Dynamic Testing

S. No.	Static Testing	Dynamic testing
1.	Without running the software, static testing is carried out.	By running the software, dynamic testing is carried out.
2.	It is carried out before compilation.	After compilation, it is carried out.
3.	Static testing focuses on defect prevention.	Finding and fixing flaws is what dynamic testing is all about.
4.	It is inexpensive.	It is comparatively costly.
5.	The testing needs lots of discussions	The testing almost makes fewer discussions necessary.
6.	It is a method of verification.	It is a method of validation.
7.	The proper steps and protocols must be followed during static testing.	Test cases for execution are part of dynamic testing.

B. Box Testing:

The testing is primarily categorized into black- and white-box testing. These techniques are utilized to demonstrate the viewpoint a test engineer adopts while drafting test cases.

TABLE II. Comparison of various functionalities of black- and white-box Testing

S. No.	Black-Box	White-Box
1.	Functional, closed-box, and data-driven are further names for this testing.	Code-based, structural, and clear-box are other names for white box testing.
2.	It is not anticipated that an application's internal workings would be revealed.	In this test, the tester is fully aware of the internal functioning of the program.
3.	End users, testers, and developers often carry out this testing.	Mostly carried out by developers and testers
4.	Testing is dependent on predicted results since the application's internal workings are unclear.	The tester can set up test data since inner processes are thoroughly understood.
5.	Not fit for algorithm testing	Fit for algorithm-testing

6.	This is the comprehensive and minimum time-consuming method.	The most extensive and time-consuming testing is this sort.
7.	Only the trial-and-error approach may be used to carry out this testing.	This testing facilitates the assessment of internal and data domain borders.

The Grey Box Testing technique is another available box strategy. Such testing is otherwise called clear testing as the tester has restricted information within application. It combines the characteristics of the black- and white-box testing methods. End users, testers, and developers all participate in the testing. Data flow diagrams and high-level data diagrams serve as a basis for testing. It is mostly tedious as well as comprehensive.

C. Automation and Manual Testing

Manual Testing: Without the use of any instruments, the product is physically tested during manual testing.

Automation testing: It is often called test automation. Such kind of testing involves the creation of scripts and the usage of an additional program or tool for testing the software. It is said to be a method for automating manual testing. The following table III may be used to illustrate the differences between manual and automated testing:

TABLE III. Manual vs Automation Testing

S. No.	Manual Testing	Automation Testing
1.	Execution speed is time-consuming as this testing is carried out by humans thus, it is tedious as well as monotonous.	Compared to human resources, automation testing is completed much more quickly.
2.	In manual testing, tests are carried out manually by a human which is time-consuming and results in more investment in human resources as a human requirement to carry out manual testing is more.	Automation tools are used to carry out automation testing of test cases which results in less investment in human resources as fewer humans are required to do the testing.
3.	In manual testing, the tester cannot write complicated tests as it is not possible to carry them out manually.	In automation testing, complicated tests are written as it is programmable as sophisticated tests can be performed easily with automation tools
4.	Manual testing is less reliable because errors made by humans make it possible for tests to not be done accurately every time.	Automation testing is more reliable and less likely to make errors.

There are many other types of software testing available, and depending on the test cases, either manual testing or automated testing may be done [1]. The additional software testing types included in the aforementioned two testing approaches are Integration testing, Unit testing, Load testing, System testing, and so on. As seen in the following Fig. 1.1(b):

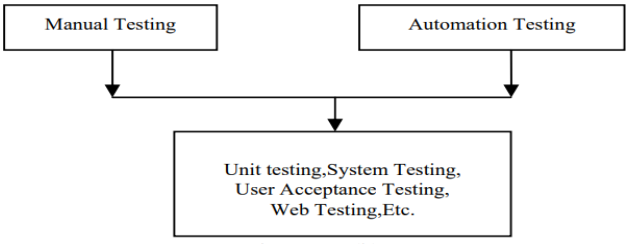

Figure 1.1(b)

1. Web Automation Testing Tool Evolution (Work done)

Automation testing has existed from the very beginning of the computing industry, therefore test automation is by no means a novel concept. In the beginning, programmers tested the software instead of testers, and testing and debugging wasn't always distinguished from one another. Throughout the 1960s, testing-related articles like the IFIPS conference proceedings almost always supposed that developers would test the software they created. As more complex and larger systems started to appear, software testers started to make sense. The first conference on software testing took place at Chapel Hill in 1972, and it was revealed in the conference papers that testing is a distinct discipline from programming. Sharing information has never been easier because of the automation of technology. The technologies for web automation testing were used by several researchers. The researchers also made several tool improvements for better performance and outcomes. To meet the challenges in software testing, these technologies are used individually as well as together. We cover work on web automation testing utilizing automated testing tools in this article.

Leott et al. [2] represent a documented related to test suite and test automation maintenance in terms of learning content management systems. The researcher's realignment attempt concluded that the ID-based strategy is more practical than the Xpath method for locating web page items. This study aims to identify the Selenium WebDriver approach that is practical for locating web page components while minimizing the support step necessary to rearrange the test cases for new output.

Leott et al. [3] provide a further description of an industry case study that was completed to analyze the anticipated benefit of the page object pattern for improving the maintenance of selenium webDriver test cases. The analysis of the two identical test suits, one of which was created utilizing page object pattern and the other without it, is covered in the study. When the above-mentioned pattern is utilized, the outputs will highlight the main regions where the time needed is decreased and the number of altered LOCs to correct the test suite.

Cheluvaraju et al. [4] introduced an ingenious method for investigating the relationships between the files that are executed together using advanced SNA (informal network analysis) on a. Resulting parameters, including the change propagation, extraction of cross-language change dependencies, and effect analysis, are given.

Wong et al. [5] showed a method for dealing with extracting an executable pattern from a Selenium IDE test suite. The number of test suites for actual systems is used to evaluate its strategy. Additionally, their study shows how model mining and model-based test production are integrated into one framework, which is an inverted process.

Nagowah et al. [6] designed an automation testing tool called the Kishanium tool by studying the current tools. The record and playback method has been improved with this modified tool. No matter how many UI changes there are, the tool still works with the set of test data and reutilizes the previous test dataset. Additional features offered by the programme include Spylink, Snapshot, and Data generator.

Wang et al. [7] presented another automation system incorporated by Jmeter and selenium. This software system shares the test methods and test dataset, making it simple to

switch between multiple web application testing methods. It supports a variety of operating systems and browsers. One may work effectively on the reusability and extensibility of automation tests by using this software platform.

4. Web Automation Tools and Testing

A. Web Testing

The term "web testing" refers to software testing that is completely focused on web programs. It facilitates in saving costs, reducing the amount of work needed to test websites and online applications, improving software quality, shortening time to market, and employing reusable test cases. Several web testing options are available, including Compatibility, Functionality, Stress, Load, Localized, Web Services Functional, Regression Web Services Performance, etc. are different testing types.

B. Web automation testing

It is challenging for companies to test their websites and online apps using a manual testing technique in the rapidly evolving and fiercely competitive web-based business environment. Therefore, automated web testing must ensure that the a variety of platforms, servers, browsers, languages, and databases and guarantee that each user of web applications receives results on time.

C. Highlights of web automation testing: -

- Saving Money and Time with Automated Software Testing
- Testing Enhances Accuracy
- Extended Test Coverage
- Manual testing cannot be done what automation can.
- Developers and testers benefit from automated QA testing
- Improves Team Morals

D. Web automation testing tool

There are several factors to consider while choosing the ideal testing tool. The first factor to consider is the simplicity of combining, which must be assessed against the cost and execution factors and how it works in a setting with network hardware and traffic. The tool must also work with the application's design and implementation. There are many different kinds of tools commercially available. The top ten web automation testing tools will be discussed, along with each tool's purpose:

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

1) Selenium: Selenium is a suite of four tools rather than just a single one[1]: selenium RC,selenium IDE, selenium grid, and selenium Webdrive. It is a compact, free, and open-source automated testing framework used for online applications that support various browsers and software. Jason Huggins, a former employee of Throught Works, a private, global provider of software services and solutions, developed Selenium in 2004. Since then, Selenium has significantly improved and has become a crucial tool for web-based automated testing.

2) HP-QTP: The HP Quality Center Tool Suite includes QTP[2]. It offers automated functional and regression testing for a variety of software environments and applications. Mercury Interactive, which HP (Hewlett Packard) acquired in 2006, originally created it[4].

3) FitNesse: Fitnesse is a tool for automating the composition, arrangement, and execution of table-based testing. The fit server, which offers a web interface for the test suite, is covered by Fitnesse, a wiki. It is based on the Integrated Test Framework developed by Ward Cunningham. A tool for improving coordination in software development is called FitNesse[3]. Fitnesse gives users, testers, and developers the ability to identify what their programme should do and then compare it to what it really accomplishes. Fitnesse looks at clients' demands for a genuine outcome. With Junit, it is also feasible.

4) Watir: Water is how Watir is expressed. In Ruby, it is introduced for testing web applications. Ruby libraries are used by the open-source Watir family to automate web-browsers[4]. Watir enables testers to create such tests that are simple to understand and follow. It is straightforward as well as adaptable.

5) TestComplete: Testers may create, organize, and execute tests for any Rich Client, Windows, and Web applications using the automated testing tool called TestComplete. SmartBear Software designed Testcomplete[5]. Making software quality tests is the major goal of this automated program.

6) LoadRunner: An automated execution and test automation tool from HP, called HP LoadRunner, is based on an industry-standard and used for load testing applications that simulate system behavior. The virtual users are how HP LoadRunner works. Additionally, it simulates a large number of synchronized users to subject the programme to several real-world user loads and thoroughly analyzes the outcomes for determining the specific manner in which the application behaves.

7) TestNg: A testing framework called TestNG, which is short for "Testing, the Next Generation," was influenced by JUnit6 and NUnit7. It has also included a few new features that make it easier and more practical for testers.

8) TOSCA: The Tosca tool is used to do functional and regression software testing automatically. Functional and regression testing may be done using the TOSCA test suite's integrated test design, management, data-generating, and execution toolset.

9) SilkTest: An automated tool called SilkTest is particularly developed for function as well as regression testing. It was first designed by Segue Software Inc., which Borland take up in 2006 before with a built-in recovery mechanism. SilkTest can also test across many technologies, platforms, and browsers. Planning, validating, managing, and directing database access for tests are all provided by SilkTest.

10) WinRunner: An automated functional GUI testing tool is HP WinRunner software. With the help of this tool, the user was able to capture, confirm, and replay UI interactions for test scripts. Mercury Interactive is the original composer of WinRunner.

2. Conclusion

We have examined a variety of automated web testing technologies in this study. Web-based automation testing is defined as the automated testing of web-based applications.

Several test automation solutions are effective for online apps. Using the Selenium tool is a well-known method of testing websites since it can record how you behave when using a website and resume the processes naturally in web browser. The primary advantage of these automation tools is that by automating the tests, you may save the manual work needed for testing each component of your website. According to research on the tools as well as web-based automation testing, Selenium is the optimal tool for web application automation.

3. 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 NetworkAnalysis," 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 fromSelenium 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 MalaysianConference 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.

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.

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