



e-ISSN: 2319-8753 | p-ISSN: 2320-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 7, July 2021

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.569

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com



Quality Engineering and Assurance with Selenium and Java

Navya VS*¹, Nandini B*²

Student, Department of Industrial Engg. & Management, RVCE, Bengaluru, India

Assistant Professor, Department of Industrial Engg. & Management, RVCE, Bengaluru, India

ABSTRACT: For a company developing a product, quality assurance is the most important step in the development process. Quality assurance enables the delivery of a stable product for consumer consumption through automation testing. Software testing is an essential component of the software development process. Many software applications are now available as web-based apps that may be accessed through a web browser. Automation testing reduces test costs while increasing human productivity, resulting in a high-quality, reliable product. Web apps add to the program's complexity, making manual testing difficult. It will take longer and will be more expensive. The goal of this work is to use the Selenium software testing tool to provide test automation for Web applications. Automation testing reduces the cost of testing and increases labor productivity, resulting in a high-quality, trust worthy product. Because web apps increase the program's complexity, manual testing is difficult. It will be more time consuming and expensive. The goal of this work is to use the Selenium software testing tool to develop test automation for Web applications. It has practically all of the properties needed to automate tests. It is impossible to produce accurate findings. Using test automation, defects can be prevented.

KEYWORDS: Software testing, manual testing, automation testing, selenium, java, quality assurance.

I. INTRODUCTION

Quality engineering can be defined as an engineering discipline concerned with service quality control and assurance, as well as product practices and principles. Software testing identifies gaps, mistakes, and missing requirements rather than the actual requirements. It can be done manually or with the help of automated solutions. Software testing is a fundamental and crucial component of the software development process. Testing is the process of evaluating a system or a module by providing defined inputs and comparing them to the desired. We can get a lot of advantages by including automated testing into the software development process: The use of a model to automate the generation of both manual test cases and automated test scripts saves time and money while also increasing coverage and reducing time-to-market.

As the software development cycle gets shorter and shorter, software testing becomes increasingly complex. Manual testing is a time-consuming and labor-intensive technique that necessitates human intervention. To avoid these issues, automation testing was developed. Automation testing refers to the use of effective software automation technologies to automate the testing process or activities, such as the creation and execution of test scripts. Automation testing raises the quality of software testing while reducing the need for human intervention.

Various commercial and open-source technologies, such as Watir, JMeter, Selenium, QTP, and others, are available to help with these duties. The Selenium automation tool is the most widely used and open- source tool for web application testing. Browsers can be used to retrieve data and complete a range of functions. Web browser automation technologies can automate error-prone and repetitive operations like filling out large HTML forms. T. Automation tools must cater to a variety of skill levels. Whether the automation is performed by a non-coder or a skilled programmer determines the sophistication of libraries and scripts. This paper presents a selenium web driver and java-based automated testing framework.



II. RELATED WORK

Initially the accepted articles were examined and sorted based on the area of the research and the factors studied in these papers were selected based on their outcomes.

1. Javeria Imtiaz, Muhammad Zohaib Iqbal, Muhammad Uzair khan, et al. (2020)¹

For automated testing of online applications, capture–replay techniques are extensively employed. The web parts of web applications are intimately tied with the scripts created for these Capture–Replay technologies. These test scripts are vulnerable to changes in web elements and will need to be updated as the web sites evolve. A model-based test repair technique that is independent of the underlying capture–replay technology for evolving web apps. The approach seeks to repair test scripts that have broken due to any of the breakage types in the web test breakage taxonomy. In addition, we assess the utility of the corrected test scripts based on the opinions of expert testers. We run a test to see how our approach compares to WATER, a state-of-the-art DOM-based test repair approach. The results of the comparison demonstrate that our method fixes more test breakages than WATER.

2. Satish Gojare, Rahul Joshi, Dhana Shree Gaigaware, et al. (2015)²

A new automated testing framework for web-based applications based on the Selenium Web Driver that reduces the time required to prepare test cases while raising the pass rate of test cases. This work can be expanded to create easy-to-understand automation test scripts using this framework. Testers can write test cases quickly and efficiently using this framework. The tester does not need to study the selenium web driver tool in depth. Developers can utilize the framework's screen capture capability to analyses their code. For the tester, this framework creates a personalized test report. These web applications are quite difficult to understand. Testing such complicated web apps is quite difficult. To eliminate human intervention and repeated operations, automation testing employs automation tools. We built and implemented an automation testing framework for web applications in this article.

3. Nitin Upadhyaya, Aniket Pant, S. Karthikeyan, Anusha LS, et al. (2020)³

This paper delves deeper into the process of a website's automation in a corporate setting. The evaluation of a product goes through two stages, the first of which is manual testing. Manual testing confirms that the product is working properly. The Automation stage is the next step. When a product is updated, the test case automation scripts must be updated as well. As a result, the test script's maintainability becomes a flaw. Companies' website automation processes differ slightly, but they all follow a common strategy. From how work is allocated to how it is to be completed, there is a talent that may be learned via practice. Writing a script for a test case isn't all that automation testing entails. It entails preparatory knowledge, manual testing abilities, coding rules knowledge, and the practice of employing a variety of software tools for automation.

4. F. Okezie1, Odun-Ayo1 and S. Bogle, et al. (2019)⁴

Software testing is an important aspect of the product development process since it guarantees that the software meets all functional requirements and is free of defects and mistakes. This ensures that the software is of high quality and conforms to industry standards. It is critical to be time and cost mindful when testing software. The goal of this article is to analyze some of the most widely used software testing tools, highlight their strengths and weaknesses. The paper's strategy involved analyzing recommended literatures to identify necessary testing instruments that were chosen based on inclusion and exclusion criteria that were assessed. There is no perfect tool for testing, however sacrifices can be made to select the best tool for a certain testing purpose based on the size of the project, the testing budgeted cost, the platform of the application, and the language used to design the project.

5. Deepthi Wilson. R, Manju prasad. B et al. (2017)⁵

To understand the need of test automation, the notion of testing and several testing approaches were introduced. This work could be expanded to serve as a guide for future Selenium and testing researchers. Software testing is an essential component of the software development process. Many software programmers are now available as web-based

applications that may be accessed through a web browser. The purpose of an automation testing procedure is to assist in the testing of software programmer applications. There were several web automations tools available, ranging from HP-QTP to Selenium. Our study paper focuses on giving a review of all web test automation tools, and as a result, it serves as a reference for future Selenium researchers. Before choosing a tool, testers must compare the features of numerous tools and choose the best one for the job.

6. Kadambari Metha, Sachin Chavan., et al. (2014)⁶

Developers and testers may easily automate the entire software testing process using testing automation tools. This paper will help us understand GUI automation and recent attempts and developments in Selenium-based GUI automation tools and QTP. Selenium is a software tool that may be used to test a wide range of browsers, operating systems, programming languages, and other applications. The purpose of this study is to examine the various GUI automation tools available for web-based applications, such as Selenium and QTP. Testing automation technologies make it simple for developers and testers to automate the complete software testing process. This paper will assist us in better comprehending GUI automation, as well as recent attempts and developments in Selenium-based GUI automation.

7. Rekha B S, Geetha V, et al. (2018)⁷

By creating an abstract method with the help of state machine automata variations and specification formulae in the form of temporal logic variations, online applications can be evaluated. Selenium is a free and open-source testing framework that may be used to evaluate a wide range of web applications. However, there are certain disadvantages to selenium, such as the inability to generate structured results and cross-browser testing. To address these challenges, selenium is widely used in conjunction with other tools like as JMeter, Junit, and Test-NG.

8. Shariff, Shahnaz Mohammedi, Heng Li, Cor-Paul Bezemer, Ahmed E. Hassan, Thanh HD Nguyen, and Parminder Flora, et al. (2019)⁸

Load testing is required for web applications in order to examine their behavior under varied load levels. Tests can be run more realistically - inside a browser - by using a browser-based testing framework like SELENIUM. Nevertheless, because a browser instance must be started in order to conduct a test, app testing has a large performance overhead, limiting its use for load tests. The scripts written for these Capture–Replay technologies are inextricably linked to the web elements of web applications. These test scripts will need to be updated when the web sites evolve due to changes in web elements. The most critical step in the SDLC is software testing, and validation of the completed product is required before it is handed to the client. Continually explore the topic in order to understand the benefits and downsides of various SDLC approaches.

9. Ramya, Paruchuri, Vemuri Sindhura, and P. Vidya Sagar, et al. (2017)⁹

Software testing has been proved to be the most important step in the SDLC. The main purpose of testing methods is to compare the findings to those that the end-user is used to. This literature will show you how to use Selenium's webdriver to test an application and display the results. After observing a program's execution sequence with the purpose of detecting faults and completing a software with zero defects, Selenium is used for software testing. This paper focuses on using Selenium Webdriver to test a web application and demonstrating how to use it in conjunction with other tools such as Maven, TestNG, and others to make testing easier and increase the quality of the process.

10. Bhoomika, Nagashreyas S, Prof. Rekha B, Prof. Geetha V, et al. (2020)¹⁰

Cloud Automation Testing is a concept that comprises testing cloud-based applications. Because the cloud allows test servers to be provided as needed, companies can save time on provisioning by using a cloud infrastructure system for testing. Selenium is a free and open-source testing framework that may be used to evaluate a wide range of web applications. However, there are certain disadvantages to selenium, such as the inability to generate structured results and cross-browser testing. To address these challenges, selenium is widely used in conjunction with other tools like as JMeter, Junit, and Test-NG. By combining and comparing different cloud automation testing solutions that are believed



to be effective alongside selenium. This article examines some of the tools that have been found to be useful in cloud automated testing.

Relevant journal articles are distributed across many fields, conducting a literature study can take a long time, especially in the software testing field. Because it is linked to practically all major online journal repositories, including Science Di, 'Google Scholar, Crossruff, and PubMed' were chosen as the primary source of searching literature to save time and effort., since it is linked to almost all major online journals repositories, including Science Direct, IEEE Xplore, NCBI, Oatext, Springer link online libraries, Research Gate, Semantic Scholar, Jyoung Pharma, Journal TOCs etc. A variety of key terms were used to scan the possible literature, such as inventory management, inventory control techniques, pharmaceuticals, drugs etc. Since it is linked to practically all major online journal repositories, such as Science Direct, IEEE Xplore, NCBI, Oatext, Springer link online libraries, Research Gate, Semantic Scholar, Jyoung Pharma, and Journal TOCs, among others. To search the literature, a number of key terms were employed, including software testing, manual testing, automated testing, selenium, quality assurance, and so on.

A. Resource Identification and Search Strategy

The research papers on software testing, namely automation testing, were chosen for this study. In order to conduct a complete and exhaustive literature search, the aforementioned keywords were chosen that are closely related to our goal.

Therefore, in the mid-1990s, the modernization of the automation testing system was considered, the literature search was limited to studies published after 1990. All of the papers that were retrieved were saved, and papers that were similar or duplicated were destroyed. The identical manuscript published in journals or conference proceedings with the same title or authors was also deleted. Only a few were chosen for the review study at the end of the process.

B. Selection of Relevant Papers

At the first phase, papers written in English were searched for keywords in the titles, abstracts and 25 papers were selected for further review. The second phase examined 20 papers specifically on the subject. Furthermore, for thorough review, 15 relevant papers were selected, and for the purpose of this study.

III. COMPARISON BETWEEN MANUAL TESTING AND AUTOMATION TESTING:

Table 1: comparison between manual and automation testing

Manual Testing	Automation Testing
Manual testing is a time-consuming and labor-intensive procedure that takes a long time to perform.	Automated tests can be done faster than manual testing since test scripts can be executed an endless number of times once, they are developed.
A large amount of human effort is required for the manual exam. Testing takes time, so more testers are needed to finish the job.	Because automated tests may be executed without the need for human intervention, they just demand a one-time effort to write the test scripts. As a result, testers are required to invest less.
Testing is done by testers manually, and no one can escape human flaws, thus it will never be perfect or entirely reliable.	It can be done with automated tools, thus there's little possibility of making a mistake while running the test.



Manual testing is appropriate for functional and exploratory tests (in which the tester explores the program's many functionalities in order to find faults).	Test automation is useful in a wide range of situations. Non-functional testing includes regression testing and virtually all sorts of nonfunctional testing, such as load testing. Manually performing performance tests is exceedingly difficult.
The only way to avoid paying for automation technologies is to conduct manual testing.	Using automated testing, you may avoid recurrence and greater expenditures. Maintenance and manual labor, for example.
Manual tests are frequently the only approach to discover new or critical issues.	Automated tests don't help much in finding new bugs because the same scripts can run several times.
In comparison to automated testing, manual testing requires a lesser initial expenditure. In the long term, Automation testing has a lower ROI than manual testing.	Automated testing requires a larger initial cost. Though, in the long run, the ROI is superior.
Small changes to a button's id, class, or other properties would not prevent a manual tester from running.	Automated Test Scripts must be adjusted to perform as expected for even minor changes in the AUT's UI.
Manual Test results are typically documented in Excel or Word, and they are not readily available.	All stakeholders can log into the automation system and check test execution results with automation testing.
Manual testing allows for human observation, which could be beneficial in developing a user-friendly system.	Human consideration is not taken into account in automated testing. As a result, it can never guarantee user friendliness or a great client experience.

IV. COMPARISON BETWEEN DIFFERENT AUTOMATION TOOLS

Table 2: comparison between different automation tools

Features	Test sigma	Selenium	Tosca	UFT/QTP
Types of applications that are supported	Any	Any	Any	Any
Testing of Web Applications	✓	✓	✓	✓
Testing of mobile web applications	✓	□	✓	✓
Apps for iOS are being tested.	✓	□	✓	✓
Testing of Web Services and APIs	✓	□	✓	✓
TESTING NEEDS				



Cross Browser Testing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Observational Testing	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
End to End Testing	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TEST CREATION FEATURES				
Test Script Development	Codeless	Code-based	Model-based	Script-based
Programming Knowledge	No	Yes	Needed for customization	VB Script
Prior Automation Knowledge	No	Yes	Need for complex scenarios	Yes
Reusability of test assets and data	Yes	Yes	Yes	Yes
TEST AUTOMATION TOOLS				
Test designing	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Test planning	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Tracability	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
TEST EXECUTION FEATURES				
Schedule test plan execution	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Scalable/Parallel test execution	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Test on local devices	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Remote test execution	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
REPORTING FEATURES				
Screenshots	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Video of test execution	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Selenium WebDriver is a web testing tool. It's open source (which means it's free), supports a variety of programming languages, including Java, Python, C#, PHP, JavaScript, Ruby, and Perl, and operates on all three major desktop operating systems: Windows, Mac, and Linux. It's a powerful testing tool that automates by communicating directly with a browser. The team behind the tool also provides a modest yet useful prototyping plug-in — Selenium IDE, which records your actions and uses them to test accordingly in the future



V. CONCLUSION

The software testing life cycle is an important part of the software development life cycle. It ensures that software released to the market is free of errors and side effects. To meet market expectations and time constraints, software testers have used automated testing tools to conduct testing. This is preferred over manual testing because automated testing saves time and reduces the amount of money spent by the organization during the testing phase. Selenium and QTP dominate the market for web-based GUI automation frameworks. Because it is more efficient, QTP is intended to be the preferred option. However, due to the high cost, it is only available to large corporations. Selenium, on the other hand, is growing more popular among testers as an open-source tool. However, it is less user-friendly than QTP, requires basic Java programming skills, and only supports limited features.

An automation testing framework has been implemented to test web-based applications based on the Selenium web driver. The proposed automation framework for testing the web application reduces the time required to create test cases while increasing the test case pass rate. It also reduces the tester's workload. This framework enables the creation of customized test reports as well as the analysis of failures through the use of screenshots of failed test cases. The tester can manage all data from a single location. This framework is ideal for frequently changing online applications. This framework simplifies the understanding of automation test scripts. Automation testing entails more than just writing a script for a test case. It necessitates preparatory knowledge, manual testing abilities, knowledge of coding rules, and practice with a variety of software tools for automation. Evidently, Selenium is a promising automated testing solution. Other tools are frequently used in conjunction with Selenium to provide additional functionality. When combined with JMeter, Selenium enables lower overhead, distributed load testing, and thus manages massive loads. The integration of Selenium with Test-NG enables the generation of structured results, as well as code annotations and cross-browser testing.

VI. FUTURE SCOPE

Since technology is advancing at such a rapid pace, there has never been a greater demand for projects to be completed as soon as possible. To complete projects on time, all techniques used throughout the software life cycle must be accelerated. In the field of software testing, automation can be used to save money and time, but only on time-consuming projects. When it comes to performing large-scale regression testing, automation testing is the way to go. It could be a good option. Test automation has several significant advantages, including improved software quality, reduced manual software testing operations, and elimination of redundant testing efforts, creating more systematic repeatable software tests, reducing repetitive work and producing more consistent testing results, and increasing consistency. More software tests can be run in a shorter period of time, resulting in greater testing coverage. Among other things, it increases productivity. It's a good way to eliminate or reduce manual labor during the execution of regression and functional test cases. Furthermore, the likelihood of a flaw escaping will be greatly reduced because errors will not occur once the script has been thoroughly developed. Automation will be seen as more cost-effective in terms of both quality and time in the coming years.

REFERENCES

1. Javeria Imtiaz, Muhammad Zohaib Iqbal, Muhammad Uzair Khan, "An automated model-based approach to repair test suites of evolving web applications", ELSEVIER, 28-September-2020, pg. 1-26.
2. Satish Gojare, Rahul Joshi, Dhana Shree Gaigaware, "Analysis and Design of Selenium Web Driver Automation Testing Framework", ELSEVIER, 2015, pg. 341-346
3. Nitin Upadhyaya, Aniket Pant, S. Karthikeyan, Anusha LS, "WEB AUTOMATION USING SELENIUM IN JAVA", International Research Journal of Engineering and Technology (IRJET),
4. Volume: 07, Issue: 05, May 2020
5. F. Okezie, I. Odun-Ayo and S. Bogle, "A Critical Analysis of Software Testing Tools", International Conference on Engineering for Sustainable World, 2019, pg. 1-11
6. Deepthi Wilson. R, Manju prasad. B, "A Comprehensive Review on Selenium Automation Testing Tool", International Journal of Engineering Research & Technology (IJERT), Volume 5, Issue 20, 2017, pg. 1-4
7. Kadambari Metha, Sachin Chavan, Manoj Patil, "Towards Developing a Selenium Based GUI Automation Test Pro: a Comparative Study", (IJERT) Vol. 3 Issue 2, February – 2014, pg. 254- 2544



8. Rekha B S, Geetha V, "Testing Tools with Selenium", International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 05, May 2018, pg. 7108-7110
9. Shariff, Shahnaz Mohammadi, Heng Li, Cor-Paul Bezemer, Ahmed E. Hassan, Thanh HD Nguyen, and Parminder Flora "Improving the testing efficiency of selenium-based load tests." IEEE, 2019, pg. 14-20
10. Ramya, Paruchuri, Vemuri Sindhura, and P. Vidya Sagar, "Testing using selenium web driver",
11. IEEE, 2017, pg. 1-7
12. Bhoomika, Nagashreyas S, Prof. Rekha B, Prof. Geetha V, "A Study of Potent Cloud Automation Testing Tools with Selenium", International Research Journal of Engineering and Technology, Volume: 07 Issue: 05 May 2020 Pg.1-3
13. Vahid Garousi Yusifog lu, Yasaman Amannejad, Aysu Betin Can, "Software test-code engineering:
14. A systematic mapping", ELSEVIER, 2014, Pg.1-25
15. Marcelo M. Eler, Andre T. Endo, Vinicius H.S. Durelli, "An empirical study to quantify the characteristics of Java programs", ELSEVIER, 2016, Pg.1-17 5464
16. María del Pilar Salas-Zárate, Giner Alor-Hernández, Rafael Valencia-García, Lisbeth Rodríguez- Mazahua, Alejandro Rodríguez-González, José Luis López Cuadrado, "Analyzing best practices on Web development frameworks: The lift approach", ELSEVIER, 2015, Pg.1-19



**Impact Factor:
7.569**



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details