Impact of Selenium in Automation Testing- Challenges and Opportunities

Anand Singh Gadwal[a], Lalji Prasad[b]
[a] SAGE University, Indore, India, anand.gadwal@gmail.com
[b]SAGE University, Indore, India, lalji.research@gmail.com

Abstract

Automation testing has become more important because it is directly related to quality and productivity. A variety of web application testing tools has been developed, and each has its own set of apparent specifications, advantages, and limitations. It is necessary to have a thorough understanding of the operation, characteristics and limitations before implementing such testing tool, which leads to the desired benefit of automation. Objective of the work is to assess an open source automation testing tool and to provide a synthesized view to researchers and QA teams. Furthermore, to analyze and evaluate features and drawbacks of the tool, to make the process of adoption and implementation easy and accurate. Selenium is most popular, widely used and accepted tool in the industry. The work provides a complete analysis of Selenium' straits and its impact in automated testing. Google Collaboratory has been utilized to illustrate Selenium's Python component. The results will be helpful to quality assurance teams, academicians and professionals in the direction of more insight in this region.

Keywords: Software testing, Web application, Web driver,Testing Framework, Colab.
DOINumber:10.48047/nq.2022.20.22.NQ10272 **NeuroQuantology2022;20(22):2789-2801**

1.Introduction

Our reliance on software enabled systems has substantially expanded because of the prevalent use of the mobile computing and Internet. This dependency causes serious questions regarding software security and dependability because a software malfunction could have terrible results. Testing a software system with the goal of identifying any fault is one of the most important ways to ensure the system's quality. Automating test construction and effecting isrequired to increase proficiency and lower expenditures. Because of repeatable tests and further recurrent test runs, automation allows for supplementary test cycles. It also allows for the quick and efficient confirmation of requirement fluctuations and bug repairs, as well as the reduction of human mistakes.

Automation testing and its success are predominantly hooked on the tool, framework, testing method and approach. For this reason, understanding the testing tool is the first stage in the direction of successful testing [1].Before implementing a testing tool, it is necessary to have a thorough understanding of its operation and features, as well as its limitations. Then only one can achieve the desired benefit of automation. According to the available literature (both in form of published and grey– websites, technical reports, MLR, etc.), Selenium is indisputably the most popular and widely used testing tool in the industry.

It is an open source toolkit or collection of tools for automating browsers which can be used to control browser instances remotely and imitate user communication [2].Considering this, this work focuses on the parameters that are not adequately addressed, eventually making automated testing easier, more useful and beneficial using Selenium.

Operating Systems	Languages	Web Browsers
Windows	Python	Microsoft Edge

Linux	Java	Google Chrome
Mac OS	Ruby	Mozilla Firefox
Solaris	Perl	Internet Explorer
	C#	Opera
	PHP	Safari
	R	HtmlUnitDriver
	JavaScript	

Table 1. Selenium Compatibility

Traditionally, testing teams are using Java and Eclipse for their test setup. For this work, Google Colaboratory has been used for executing selenium python scripts. Colab enables us to develop and run Python in the browser with no new configuration, and at no expense for GPU access. Google Drive is linked with Colaboratory. We may simply share Colab notebooks with co-testers, encouraging them to leave comments or even modify our notebooks/test scripts. Colab notebooks are identical to Colab-hosted Jupyter notebooks [3].

Automation Framework

Test automation framework is bunch of rules, standards, libraries and tools that may be used to develop test cases. Using test automation framework such as Selenium, these test cases may then be setup and implemented into the release operation by means of a CI/CD channel. Framework for automating tests includes processes and tools created to generate effective test cases. These include coding standards, methods for processing test data, object repository management, and access control management for the test conditions.However, testers have greater autonomy. The testers are not restricted by these regulations and guiding principle. They are free to develop test cases in the manner of their choosing. Nonetheless, a framework standardises the procedure of testing, resulting in a well-organized, safe, and consistent test procedure.

There are now six prominent frameworks available for test automation:

a) Linear
b) Modular driven
c) Library architecture
d) Datadriven
e) Keyword driven
f) Hybrid testing framework [4].

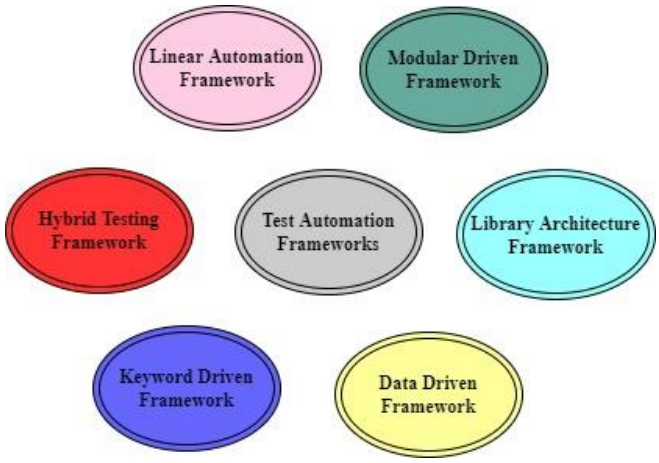

Fig.1. Test Automation Frameworks

There are several significant benefits to following policiesalong with recommendations set by automation framework. These benefits contain:

- Acceleration and enhancement of the entire testing procedure
- Test case precision and reproducibility are enhanced
- Reduce maintenance costs with consistent procedures and practises
- Reduced the use of manual processes and human mistake

- Maximum possible test coverage of all software modules, from the user interface to the core application logic[5].

2 Related works

This work describes the necessity and significance of automation in software testing in order to deliver a quality product. The authors also discussed the development of Webdriver frameworks as solution. This publication also included both risks and prospects for testing with WebDriver [6].

Mentioned Selenium Webdriver specific information and proposed automated testing outline using Selenium Webdriver and TestNG[7].The authors attempted to demonstrate integration of static and dynamic analysis in their systematic map review. Furthermore, the arrangement of various testing coupled with inspection methods has been discussed[8].It extends the test libraries—Selenium WebDriver, NUnitto provide an in-house build-upsolution for multifaceted and dynamic web page testing in various web browsers. They also talked about their experiences, the challenges they faced, and how they dealt with them [9].This survey provides taxonomy of existing software testing tools. They believe classification is necessary for the same. They also presented a comparison of more than 30 tools based on their attributes [10].They compared tools like Selenium, QTP, and Watir based on the effort involved in test script generation, cost, efficiency, reports, and so on. They examined features to reduce the resources required for script maintenance [11].This research discussed an automatic test creation procedure for integrated testing of software applications.They also mentioned that,Present-Day Model-Based Functional testing has been the focus of testing tools. Due to the distinction between functional test models and security test models, they cannot be easily adapted to security testing [12].The authors discussed the disparity between practitioner and academic perspectives. They consider experiments and case studies demonstrate the benefits of automation, but experience reports include limitations[13].When it comes to testing of webapps, it is always superior to make an automated framework for testing. Selenium WebDriver is an applicable solution when making such a framework. Selenium is still the most prominent and commonly used tool for developing automated testing frameworks [14].In this paper, an automated tool is described as the best approach for increasing the production, success, and exposure of software testing. QTP and Selenium were compared, with an emphasis on an examination of the benefits and drawbacks of each tool. This can make it easier for developers to choose one that meets their needs [15].This paper demonstrates how the use of big data can help attain better outcomes while saving time and money. It also demonstrates how businesses should embrace big data technologies, reducing test execution time and effort by applying ML systems to automate the entire testing process [16]. This article showed the key characteristics of various frameworks for automating testing. In addition, a summary of various scripting methodologies is provided. The writers discussed recording/replay and a keyword-driven strategy. The second approach is the framework for programmable automation testing, which seeks to automate programs using all the characteristics, principles, and best practices of conventional development [17].The authors cited the selection of an automated-testing tool and an acceptable framework as one of the most critical challenges for automation. The purpose was to evaluate and contrast twenty-one different automation-testing technologies across twenty criteria. They also outlined the merits and disadvantages of various frameworks [18].For the purpose to assess the present practices and prospects for enhancement of STMTs, the authors performed a survey with broad range of software development organizations and industry experts. The outcomes of their study reveal five major results concerning the present practices of STMTs and possibilities for betterment [19].This study examined the various growing Patterns of Testing methods and tools presently available. They also claimed that the primary concern of engineers in every project is that project during design is of high quality. Based on available publications, they have compared various testing tools as well as conducted a comparative analysis of various automated testing procedures, which aids in the identification of testing tools/ methodologies that are both cost and time-effective [20].The authors described the

difficulties that people working in software artefact maintenance face when it comes to understanding code from a testing standpoint. They concluded that the complexity metrics make it easier to understand design quality, the variety and quantity of testing required [21]. Multiple testing solutions, including Apache JMeter, QTP, Selenium, TestComplete, WinRunner, Watir, TestNG, Performance Tester, SahiPro, etc., were surveyed by the authors. They made an effort to provide an overview of the tools that are available to support the practise and to summarise the extant research. They also made an effort to give testers information about tools so they could select the one most suited to their AUT [22].

3 Research method and motivation

Webapplication testing is quite challenging. WebApplication test automation is one more type of software automation, but this deserves special attention. However, any type of application can benefit from or be tested by automated testing. Table 2 contains research questions for conducting analysis and evaluation on Selenium, for testing Web applications.

RQ1	What are the main specifics of Selenium which makes it superior in automation testing?
RQ2	What are the challenges identified by practitioners and listed in the literature related to Selenium?”
RQ3	What are the future directions of automation testingusing Selenium?

Table 2. Research questionnaire

An in-depth comprehension of a testing tool's functionality, capabilities, and restrictions is required prior to implementation. Many research papers and reports have discussed the use of Selenium for automation. Additionally, comparisons with other commercial and free tools are provided in the literature. However, holistic analysis and evaluation of its architecture, operation, characteristics, and limitations are not available in a single research document orlocation. Therefore, a detailed description is needed to bridge the gap. As a consequence, supplementary research has been conducted in this reference to make automated testing easier, more useful and beneficial using Selenium.

4 Selenium

One of the open source, software automation tool kit or framework. This can be used for validation of web apps on various platforms and crosswise diverse browsers. It supports various computer programming languages for writing code or test scripts, such as Python, Ruby, Java, and Kotlin. According to the available literature and industry trends, Selenium is indisputably the most popular and widely used testing tool in the domain. Selenium WebDriver is an applicable solution when making a webapp testing framework.

4.1 Project Components of Selenium

Selenium is a collection or bunch of testing tools, libraries and or APIs, instead of a sole software application or program. Each one is designed to meet specific testing and test setup specifications of a business [23].

Available tools in the Selenium array are:

1. Selenium IDE
2. Selenium1 or RC
3. WebDriver
4. Selenium Grid

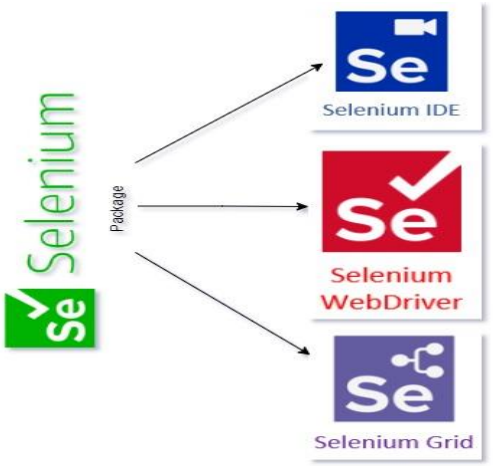

Fig. 2. Selenium Suite

Selenium IDE

Learning and practicing Selenium IDE is easier than other tools included in the Selenium suite. It is available as a Firefox plugin that can be installed just like any other plugin. It provides recording and playback features that make it extremely simple to learn, even if QA teams are unfamiliar with any programming language. Selenium test cases are created using an IDE. It is a simple Firefox and Chrome extension that is proficient method to write test cases. With available commands it can record user actions in a web browser, along with well-defined criterion by the element's context. It is also a great way to learn Selenium script syntax. With several advantages, Selenium IDE also had a few drawbacks, making it unsuitable for use with more advanced test scripts [23, 24, 25].

Selenium RC (Remote Control)

For a considerable duration, Selenium1 or RC was the premier framework of the entire Selenium project. It is the inaugural automation web testing instrument that enables users to use their preferred coding language. Perl, Java, Ruby, PHP, C# and Python were supported by RC [23]. Selenium RC was the primary Selenium initiative, until WebDriver and Selenium merged to form Selenium 2. It was a Java-based tool that allows users to create scripts for web apps. Selenium RC was created to address the various shortcomings of Selenium IDE. Selenium RC was compatible with a variety of operating systems and browsers [24,25].

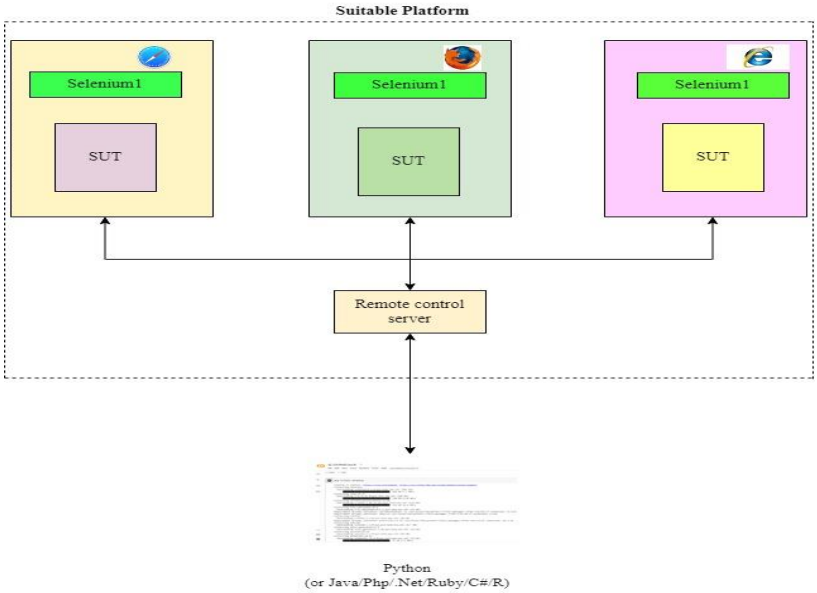

Fig. 3.Architecture of Selenium RC[26]

A developer or tester writes test code in their favorite computer programming language. A client library exists per each computer language. Test instructions are sent to the server by the

client library. The Selenium server decodes and translates those commands into JavaScript commands, which are then sent to the browser. The browser uses Selenium Core to start executing the commands and reports the output back to the server. The test results are delivered to client library by the Selenium server. There is indeed a programming function inside every interface that claims to support every Selenese command. It is important to note that Selenium RC is no longer offered [20, 23, 26].

WebDriver

WebDriver is capable of working with all modern browsers after a simple setup. Real world user interactions can be automated in Google Chrome, Safari, Microsoft Edge, IE, Firefox and other browsers [27].Selenium 2 was created following an examination of RC and web driver compatibility. Web Driver's capability to drive a browser naturally, like a user, on local machine or remotely via the Selenium server; represents a significant advancement in browser automation. It refers to the specific browser governing code implementations with language bindings.

Fig. 4. Supported Browsers

Each browser is supported by a unique WebDriver implementation known as a proxy or driver. The driver is in-charge of delegating to the browser and handling information exchange to and from Selenium. This splitting is deliberately done to make browser vendors accept some responsibility for browser implementation. Selenium uses third party drivers whenever possible, and also offers its own drivers managed by the project in instances where this is not the case. Each of these parts is connected via an interface (viewer) by Selenium framework that makes various backends of browsers to be utilised in a transparent way, allowing for cross platform and multi browser automation. Using WebDriver to create a test suite requires an understanding of and proficiency with multiple components. These parts or components includes: 1) collection of "commands" used to control WebDriver- i.e. API, 2) A code module containing the APIs with associated implementation code i.e. Library, 3) Responsible for managing the browser itself. The majority of drivers are developed by browser makers. Drivers are often executable programs that operate on the same system as the browser, and not the system conducting the test suite- i.e., Driver, 4) Complementary library used as supplement to WebDriver suites. These frameworks might include test frameworks like JUnit / NUnit. There are also frameworks that support natural language features, such as Cucumber and Robotium- i.e., Framework [27].

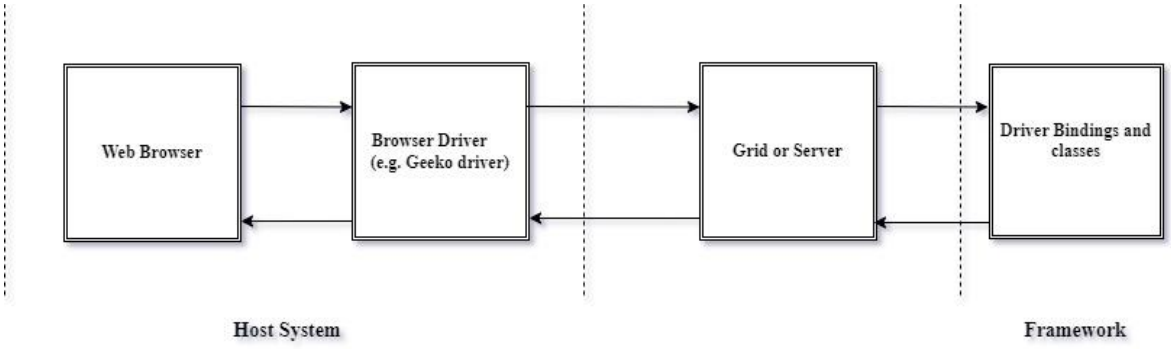

Fig.5Seleniuminternal parts [27]

WebDriver communicates with browser through driver at a minimal level. Interaction is bidirectional: WebDriver sends messages to browser via driver and gets data back along the same path [27].WebDriver is really a pure object-oriented framework. It automates by using browser's native suitability rather than a third party entity. WebDriver is a remotely controlled interface that permits for user agent self-examination and control. It offers a language and OS neutral wire protocol that allows out-of-process programs to remotely instruct web browser behaviour [28].

WebDriver was indeed developed to be a programming interface that is more straightforward and precise. WebDriver is a portable API that is object-oriented in nature.It can drive the browse flawlessly. W3C endorses this WebDriver [29].

Selenium Grid

Grid allows us to execute our test scripts on multiple nodes parallelly. Running more than one test at once is called parallel testing (execution). By simultaneously executing tests on numerous workstations, Selenium Grid elevates WebDriver and speeds up testing across various operating systems and browsers. Grid enables to run automation scripts on a variety of browsers, operating systems, and version combinations. Reduce the total runtime of automation test codes and accelerate script execution. But it still requires parallelizable scripts. Selenium Grid enables the running of WebDriver test code on distant nodes by forwarding client commands to distant browser instances. The goal of Grid is to: 1) Offer a simple method for running tests simultaneously on various machines. 2) Permit testing with several browser versions. 3)Make platform-neutral testing possible [30].

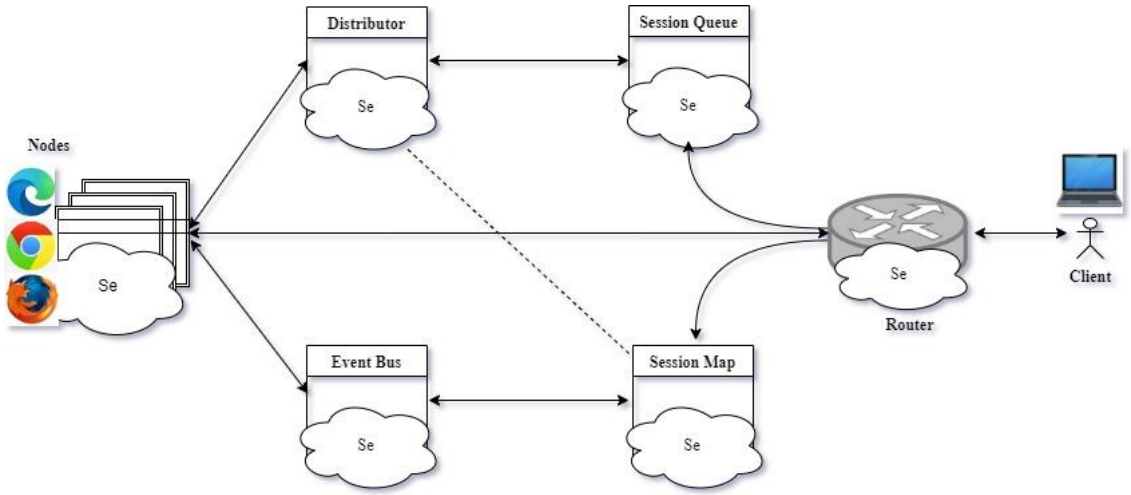

Fig. 6 Communication between components [30]

Remote WebDriver

WebDriver can be utilised remotely in the same manner as locally. A remote WebDriver must be configured in order to execute our tests on a distant machine. The components of a remote WebDriver are a client and a server. The first one is our WebDriver test, whereas the second one is only a Java servlet that can be hosted on any contemporary JEE application server. To execute a remote WebDriver client, we must first

establish connection to the remote WebDriver. This is accomplished by pointing the URL to the server executing our tests. To specify our configuration, we define desired capabilities [31].

Selenium 4

The Selenium IDE, WebDriver, and Grid have all undergone significant changes in Selenium 4. So now Selenium supports CDP (Chrome Debugging Protocol), testing team can use network and performance toolkits that are a part of CDP in their automated tests. The new Grid is convenient and more effective than earlier iterations of Selenium Grid, which had a complicated setup. The latest Selenium Grid eliminates the requirement to establish hubs and nodes individually because it now functions as both. The W3C has completely standardized Selenium Webdriver. Capabilities are switching from Legacy/JSON Wire Protocol to WebDriver protocol as a result of the move to W3C. Cross-browser testing is now more effective and reliable as a result [32].

Running Python-Selenium in Colab

Colab by Google is Jupyter based open-source environment where Python code may be written and executed. Selenium Python libraries offer a straightforward API for writing functional tests with Selenium WebDriver. By using Selenium Python API, we may get every Selenium WebDriver features indirect manner. Multiple programming paradigms are supported by Python, making it a valuable option for test automation.

Python offers various inbuilt testing frameworks, like PyUnit, Pytest, and Robot, which facilitate a quicker workflow and debugging. We must install packages like Webdriver and Selenium in Colab to make it run. Having gone through details of all the major components, now look at source code for illustrative purpose [33].

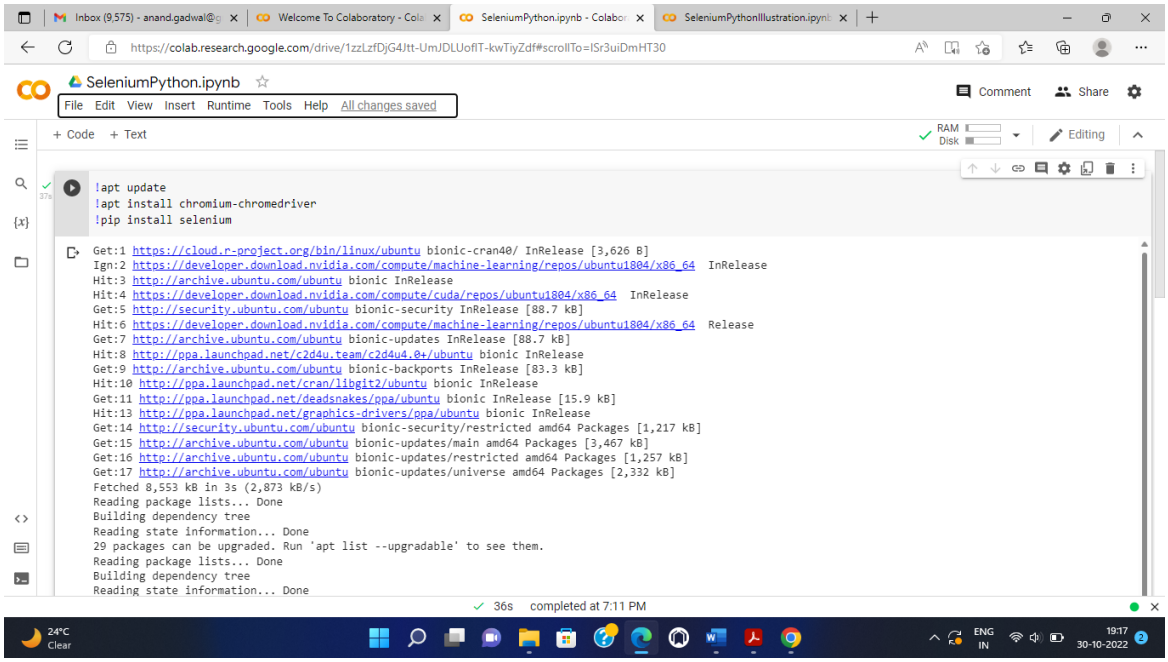

Fig. 7 Selenium-Python Colab environment

Colab comes along with modules already installed. No installation of Python and its package is needed. It operates on web browsers. It enables us to interact with Jupyter notebooks in browser. Colab is open-source platform, meaning that everyone can access it for free. It also provides GPU capability. We can perform headless browser testing using Colab.

5Result and discussions

This section mainly goes over the answers to the formulated RQs. For each case, the authors discuss how these findings can help both industry professionals and researchers.The time and effort invested in tool evaluation activity can decide the success of the testing project. The tool is chosen based on a number of factors, including:

- The software and technology stack that will be tested
- Specific testing criteria and requirements in detail
- The organisation's skill sets
- The tool's licence fee [34]

5.1 Assessment Characteristics

There is a necessity to identify aspects for analysis because assessment can be performed using various parameters or characteristics.

Characteristics	Meaning
OS Compatibility	Operating systems supported
Browser Compatibility	Cross browser testing supported
Record and playback	capability of tools to capture and replay tests
Scripting language support	Languages support for writing and editing scripts
Curve of learning	How easily the tool can be used
Data driven	The capability of the programme to reduce effort by allowing scripts to access many groups of source data from external sources such as data tables and excels.
Programming skills	Level of programming skills required
Cost effectiveness	Whether free or licensed
Function	Type of testing supported
Reporting facility	Representation of test results

Table 3. Characteristics of tools

Selenium supports for following characteristics:

Criteria/ Tool	Platform Compatibility	Browser Compatibility	Programming Skills	Cost	Modifiability	Language
Selenium	Cross platforms	Cross Browser	Needed	Open source (Free)	Yes, modifiable	Java, Ruby, R, python, php,,C#, .Net

5.2 Characteristics of Selenium which makes it superior are:

It supports multiple browsers. It is open source hence, no licensing cost or maintenance fee. It gives high performance at speed. It supports multiple platforms, compatible with a variety of operating systems.

Browser Name	Selenium IDE	Selenium RC	WebDriver
Microsoft Edge	Yes	No	Yes
Google Chrome	Yes	Yes	Yes
Mozilla Firefox	Yes	Yes	Yes
Internet Explorer	No	Yes	Yes
Opera	No	Yes	Yes
Safari	No	Yes	Yes
HtmlUnit Driver	No	No	Yes

Table 4. Browser support

It supports various computer programming languages for writing code or test scripts.IDE is available as a Firefox and Chrome plugin. It provides recording and playback features. It requires less programming knowledge, simple to learn with little knowledge ofany computer language. It offers a language and OS neutral wire protocol, Fully W3C compliant. WebDriver is a portable API that is object-oriented in nature. Actively developing repository. Grid allows us to execute our test scripts on multiple nodes parallelly.

Numerous plugins for Selenium allow it to be used in ways other than its intended purpose. We can discover other unsupported types on Github, but some of them (like Appium and Selendroid) are kind of formally endorsed by Selenium. These plugins are available for Selenium IDE, Grid in addition to WebDriver. The latter ones are very thorough and can strengthen the operationally weaker tool and prepare it for production.

5.3 Challenges identified by practitioners and listed in the literature

Resources and Knowledge

To work on Selenium and associated frameworks, a significant amount of technical skill, expertise, and understanding are required. To utilize the Selenium to its entire potential, developers and testing teams need to be familiar with its architecture.

Maintenance and Scalability

For test automation professionals, maintaining tests when using Selenium can be difficult. Tests can be broken by small UI update. As a result, debugging and maintaining such scripts is difficult without having appropriate experience in preserving tests in Selenium framework. Selenium enables cross-browser and cross-OS testing. However, depending on the number of hub/nodes are accessible, the pace and amount of testing that is permitted is constrained [35].

Lack of Test Reporting Capabilities

Selenium is excellent for creating, automating, and running tests. However, Selenium appears to lack exposure into test data generation and reporting. There is no inbuilt test reporting mechanism available.

WebDriver does not support new browsers easily

Keep in mind that WebDriver functions at the operating system level. Also consider the fact that the operating system and distinct browsers communicate in various ways. When a new browser is released, it might communicate with the operating system in an unique manner than other browsers. In order for the WebDriver group to incorporate the new process in the upcoming WebDriver release, you must give them a good amount of time to figure it out [36].

Utilizing WebDriver to crawl over URLs is not recommended. Not because it is impossible, but just because WebDriver is not optimal tool for this purpose. WebDriver requires time to initialise, and it can take a few seconds to a minute, based on how your tests are written, to reach the webpage and explore the DOM [37].

Selenium users typically struggle with the difficulties of developing a functional testing framework and keeping it up to date in order to generate value over time, amidst its important benefits being its widespread adoption and good community assistance [38].

One intriguing aspect is that the IDE was previously restricted to Firefox users only; however, as of the latest version upgrade (3.17.0), Selenium team has released a Selenium IDE Edge and Chrome Add-on in 2020, allowing Chrome users to access the same environment that Firefox users have access to [39].

5.4 Future directions of automation testing using Selenium

Testing and development teams are combined as one collaborative pipeline to consistently develop and test web apps in order to maintain a balance between profit and quality. Selenium can be integrated easily with DevOps and other CI/CD technologies. It is possible to work out the perfect, and simple integration of WebDriver with other testing frameworks. To assist in automating repetitive operations, AI can be incorporated into selenium testing. This may involve creating, evaluating, and running test codes. Automation testing cannot be replaced by AI, but it may be made to go more quickly and effectively. To expedite the testing of products, software testing teams require AI and machine learning approaches. Maven can be used to quickly configure and maintain the Selenium Testing Environment.

6 Concluding remarks

Selenium is built to inspire and assist test automation of functional aspects of internet-based apps, across a variety of web browsers and operating systems. It is a straightforward and simple solution for increasing testing productivity.The drawbacks of IDE are actually not drawbacks of selenium. They merely represent IDE's potential accomplishments' boundaries. WebDriver can be used to get around these restrictions. In many ways, Selenium IDE and Selenium RC are outperformed by the WebDriver. It uses a more up-to-date and reliable technique to automate browser actions. JavaScript is not required by WebDriver for

testing. Through direct communication, it manages the browser [40].

A crucial and difficult task in software development practice is determining which test automation tools are best suited for the current project. If we select inappropriate tool/ framework for automation testing, then it may negatively affect the costs [41]. The ability to fully automate testing is still not possible with a single tool. Tools can, however, be integrated to fulfill testing requirements.This paper provides an in-depth analysis and evaluation of selenium in order to meet the objectives. This work also discussed all components of the Selenium Project, including from the first to the latest version of the Selenium release. The findings will point researchers, IT experts, and SQA teams in the direction of further information about this domain.This researchalso offersinsight to developers and testing teams that can drive essential decisions concerning their mission. This knowledge will aid in a better comprehension of the Selenium tool. The bottom line is that business optimization is a result of automated software testing.Future research can focus on enhancing WebDriver's functionalities to make it a one-stop solution, for fully automating all testing requirements. The key will be WebDriver's flawless, easy, and seamless integration with other testing frameworks.

REFERENCES

[1] Umar, M Albarka& Chen, Zhanfang. A Study of Automated Software Testing: Automation Tools and Frameworks. International Journal of Computer Science Engineering (IJCSE). https://doi.org/10.5281/zenodo.3924795 (2019).

[2] A deeper look at Selenium: https://www.selenium.dev/documentation/overview/details/, last accessed 2022/07/20.

[3] Welcome to Colab!:https://colab.research.google.com/, last accessed 2022/06/02.

[4] Test Automation Frameworks: The Ultimate Guide. https://www.bmc.com/blogs/test-automation-frameworks/,last accessed 2022/02/10.

[5] Test Automation Frameworks: The Ultimate Guide. https://blogs.bmc.com/test-automation-frameworks/?print-posts=pdf, last accessed 2022/02/10.

[6] Elior Vila, GaliaNovakova, and Diana Todorova. 2017. Automation Testing Framework for Web Applications with Selenium WebDriver: Opportunities and Threats. In Proceedings of the International Conference on Advances in Image Processing (ICAIP 2017). Association for Computing Machinery, New York, NY, USA, 144–150. DOI:https://doi.org/10.1145/3133264.3133300.

[7] Satish Gojare, Rahul Joshi, DhanashreeGaigaware, “Analysis and design of Selenium testing tool web driver automation testing framework,” published by Elsevier, ISBCC 15, pp. 341 346, 2015.

[8] Frank Elberzhager, Jürgen Münch, Vi Tran Ngoc Nha, A systematic mapping study on the combination of static and dynamic quality assurance techniques, Information and Software Technology, Volume 54, Issue 1, 2012, Pages 1-15, ISSN 0950-5849, https://doi.org/10.1016/j.infsof.2011.06.003.

[9] V. Debroy, L. Brimble, M. Yost and A. Erry, "Automating Web Application Testing from the Ground Up: Experiences and Lessons Learned in an Industrial Setting," 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), Vasteras, 2018, pp. 354-362, doi: 10.1109/ICST.2018.00042.

[10] K Shaukat, “Taxonomy of automated software testing tool”, International Journal of Computer science and innovation,” vol. 2015, pp. 7-18, 2015.

[11] V Malik, M Ghalan, “Comparative study of automated web testing tools”, Volume 6 Issue 3-January 2016, 367-374.

[12] Xu, D., Xu, W., Kent, M., Thomas, L., & Wang, L. (2015). An Automated Test Generation Technique for Software Quality Assurance. IEEE Transactions on Reliability, 64(1), 247–268. doi:10.1109/tr.2014.2354172.

[13] Dudekula Mohammad Rafi, Katam Reddy Kiran Moses, K. Petersen and M. V. Mäntylä, "Benefits and limitations of automated software testing: Systematic literature review and practitioner survey," 2012 7th International Workshop on Automation of Software Test (AST), Zurich, 2012, pp. 36-42, doi: 10.1109/IWAST.2012.6228988.

[14] Elior Vila, GaliaNovakova, and Diana Todorova. 2017. Automation Testing Framework for Web Applications with Selenium WebDriver:

Opportunities and Threats. In Proceedings of the International Conference on Advances in Image Processing (ICAIP 2017). Association for Computing Machinery, New York, NY, USA, 144–150. DOI:https://doi.org/10.1145/3133264.3133300.

[15] R. K. Lenka, U. Satapathy and M. Dey, "Comparative Analysis on Automated Testing of Web-based Application," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida (UP), India, 2018, pp. 408-413, doi: 10.1109/ICACCCN.2018.8748374.

[16] Stouky A., Jaoujane B., Daoudi R., Chaoui H. (2018) Improving Software Automation Testing Using Jenkins, and Machine Learning Under Big Data. In: Jung J., Kim P., Choi K. (eds) Big Data Technologies and Applications. BDTA 2017. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 248. Springer, Cham.

[17]Ḥannā, M., Elsayed, A.E., & M., M. (2018). Automated Software Testing Frameworks: A Review. International Journal of Computer Applications.

[18]Ateşoğulları, Dilara & Mishra, Alok. (2020). AUTOMATION TESTING TOOLS: A COMPARATIVE VIEW. International Journal of Information and Computer Security. 12. 63-76.

[19] Lee, Jihyun& Kang, Sungwon& Lee, D. (2012). Survey on software testing practices. Software, IET. 6. 275-282. 10.1049/iet-sen.2011.0066.

[20] Ibrahim, Jawwad& Hanif, Sundas& Shafiq, Sobia &Faroom, Saeed. (2019). Emerging Trends in Software Testing Tools & Methodologies: A Review. International Journal of Computer Science and Information Security,. 17. 108-112.

[21] Yadav R, Patel R, Kothari A. Critical evaluation of reverse engineering tool Imagix 4D! Springerplus. 2016 Dec 23;5(1):2111. doi: 10.1186/s40064-016-3732-x. PMID: 28074171; PMCID: PMC5182245.

[22] Yadav, R., Vore, N, Prasad, L. (2021). A Systematic Literature Review of Automated Software Testing Tool. In: Abraham, A., Castillo, O., Virmani, D. Proceedings of 3rd International Conference on Computing Informatics and Networks. Lecture Notes in Networks and Systems, vol 167. Springer, Singapore. https://doi.org/10.1007/978-981-15-9712-1_10.

[23] https://www.softwaretestinghelp.com/selenium-tutorial-1/, last accessed 2022/06/12.

[24] https://www.guru99.com/introduction-to-selenium.html, last accessed 2022/06/16.

[25] https://www.selenium.dev/projects/, last accessed 2022/07/08.

[26] https://www.selenium.dev/documentation/legacy/selenium_1/, last accessed 2022/07/08.

[27] https://www.selenium.dev/documentation/webdriver/getting_started/, last accessed 2022/08/19.

[28] https://www.w3.org/TR/webdriver/, last accessed 2022/08/19.

[29] https://www.selenium.dev/documentation/webdriver/, last accessed 2022/08/02.

[30] https://www.selenium.dev/documentation/grid/, last accessed 2022/05/26.

[31] https://www.selenium.dev/documentation/webdriver/drivers/remote_webdriver/, last accessed 2022/04/20.

[32] https://www.selenium.dev/blog/2021/announcing-selenium-4/, last accessed 2022/03/08.

[33] Overview of Colaboratory features: https://colab.research.google.com/notebooks/basic_features_overview.ipynb

[34] Umar, M Albarka& Chen, Zhanfang. A Study of Automated Software Testing: Automation Tools and Frameworks. International Journal of Computer Science Engineering (IJCSE). https://doi.org/10.5281/zenodo.3924795 (2019).

[35] https://www.perfecto.io/blog/limitations-of-selenium, last accessed 2022/07/30.

[36] https://www.guru99.com/introduction-webdriver-comparison-selenium-rc.html, last accessed 2022/04/23.

[37] https://www.selenium.dev/documentation/test_practices/discouraged/link_spidering

[38] https://www.softwaretestinghelp.com/category/software-testing-tools/, last accessed 2022/04/23.

[39] https://www.browserstack.com/guide/what-is-selenium-ide, last accessed 2022/06/18.

[40] https://www.perfecto.io/blog/selenium-latest-version-selenium-releases, last accessed 2022/08/04.
[41] Romulo Martins da Silva, Cafer Cruz, Heleno de S. Campos, Leonardo G. P. Murta, and Vania de Oliveira Neves. 2019. What is the adoption level of automated support for testing in open-source ecosystems? In Proceedings of the IV Brazilian Symposium on Systematic and Automated Software Testing (SAST 2019). ACM, New York, NY, USA, DOI: https://doi.org/10.1145/3356317.3356325.