A Critical Analysis of Software Testing Tools

F. Okezie¹, I. Odun-Ayo¹ and S. Bogle²

¹Department of Computer and Information Sciences, Covenant University, Ogun State, Nigeria
²Department of Computer Science, Humboldt State University, USA
Corresponding Author; fiona.okezie@stu.cu.edu.ng, isaac.odun-ayo@covenantuniversity.edu.ng

Abstract
Software testing is a crucial part of software development, it ensures that the software been developed performs all functional requirements and is free from any form of defect and errors. This ensures that the software is of good quality and standard. While testing a software, it is important to be time and cost consciousness. This reason has made most testers switch from the manual testing process to automation of software testing, to reduce time and cost. But then selecting a software testing tool for automated testing that best fit a project is important yet challenging task, the objective of this paper is to evaluate some of the most used software testing tool, identify their strength and weakness and also the field where they can be employed, either for mobile testing, web service testing or both. The method utilized in the paper involved the analysis of recommended literatures to pinpoint necessary testing tools selected based on inclusion and exclusion criteria, that were evaluated. The result of the analysis indicted that based on the selected criteria, testing tools that supports web platform testing made up 17%, while tools that supported desktop and mobile platforms was 10% and 7% respectively. also, 7% of the tools examined were found to be open source tools while 12% were licensed tools. 10% of the testing tools examined supported the test result and report generation criterion while tools that require the knowledge of programming language was least as only 3% of examined tools supported this criterion. It was observed that there is no one perfect tool for testing, but for a particular testing purpose, tradeoffs can be made to select the best tool depending on the size of the project, the budgeted cost for testing, the platform of the application and also the language that is used to develop the project.

Key words: Software Testing, Testing Tools

1. Introduction
In the world today, peoples’ jobs, comforts, safety, entertainment, decisions, and their very lives depends on computer software, so it's better to be done correctly [1]. This is one reason software testing is of utmost importance. Software testing is as essential as any other stage of the software development life cycle. Testing must be done before deploying the software for use, this helps to discover errors on time and ensure the functionalities of the software works as necessary, it also minimize defect discovered after the product has gone into use. Software testing is a significant part of software quality assurance (SQA), it is an activity used for evaluating and improving software quality [2]. Software testing involves set of activities carried out with the sole aim of finding errors in software. It validates and verifies if the software or product is functioning correctly without any errors or bugs capable of incurring defect. In the testing phase, errors from previous phases should be detected, this ensures software reliability and quality assurance [3].

The product quality relies on different parameters such as response time, performance, reliability, maintainability, correctness, testability, usability and reusability are also measured to know how well the quality of a software has been achieved, this is attainable via testing [3,4]. Although,
testing can be time consuming and about 50% of the budget of software organizations spent on testing [2,4], it is better to get it right initially than handling the cost of a failed software afterwards. Hence, the reason and objective for testing is to conduct a stepwise and systematic detection of different classes of errors within a minimum amount of time and less amount of effort [2,5].

With the increase in software functionalities, it is necessary to use a testing methods and tools to ensure that time spent on testing is minimized. When it comes to testing the functionality of a software, there are two types; manual and automation. Manual testing is carried out by the tester. Informal review, inspection, walk through and technical review are the techniques of manual testing. Manual testing is time-consuming and requires more effort, this are major issues with this kind of testing. Automation testing helps to solve and curb these issues. Automated testing can be categorized into four; performance testing, safety testing, accuracy testing and testing of reliability. Using automation tools, steps involved in manual testing are been automated. These tools test the process and are targeted to specific testing environment [3].

Software testing tools facilitate generated information handling, communication, test execution, generation and production [6]. Hence, the utilization of appropriate software testing tool effectively and efficiently enhances the testing process. Presently, a large amount of software testing tools are available to assist in any phase of the testing process [7]. Even though many software testing tools are mainly useful in managing and keeping track of software tests that are scheduled or done; however, some software testing tools provide automation for core testing task [4]; this reduces the costly, time-consuming and error-prone manual testing [8]. Test automation facilitates the identification of errors and faults effectively and efficiently in particular developed software; as it enhances the reliability factor, saves time and increases productivity in human efforts and also reduces costs over the long term [4].

Due to the need to ensure software quality and recommend various testing tools that would be effective in testing software, many studies have been conducted; [4] examined the different testing tools based on usability metrics such as ease of use, technical support, ease of installation and configuration records and playback and GUI. The result of the study suggests that appium is an appropriate testing tool for android platform while Ranorex is the best and appropriate testing tool for website application. Nevertheless, [8] conducted a comparative study of 2 automated testing tools; Open Script and Selenium stating that both automated testing tools are efficient tools. The study further stated that Selenium could be utilized for browser-based applications while Open Script is appropriate for automating Java-based frameworks.

[6] conducted a study to examine several software testing tools that aid software testing process in different approaches. The study examined different software automated testing tools and identified the distinctiveness of each of the software testing tools. A number of the examined software testing tools simulate the final execution environment as a means of accelerating the test execution; however, some software testing tools automate the test plan development; nonetheless, during execution, other examined tools collect performance data. Higher quality, productive testing is facilitated by automated testing tools. However, the acquisition of such testing tools is often complicated [9].

Additionally, [10] provided a detail in-depth explanation of an automated software testing tool known as selenium. While the research conducted in [11] provided a detailed explanation of openscript as an automated software testing tool which provides programmers and stakeholders
with in-depth details on technical implementation for the tool and automation engineering. The report aimed to discover the developer perspective on the various automation testing tool available and their setbacks. [12] conducted a comparative study which provided criteria that described tools like HP Quick Test Professional, selenium and Test Complete functional testing tools.

[13] conducted a study which categorizes and distribute several automated testing tools over the different types of testing method for three different types of software product such as web application, application software and network protocol. [14] studied the release document of software testing tools and evaluated the terminology present in the release document. The result from the study suggests that differences exist between terminologies of software testing tool and that of the learning resource. However, [15] stated that if a reasonably well-structured system implementation exists, the addition of a mechanism to capture integration with operation provided by the system to generate meaningful playback would be easy.

Finally, [16] investigated the practitioner view and evaluation of the different software testing tools using an online survey. The study result suggests that evaluation based on experience in utilizing software testing tools shows more positivity than of that of opinion-based evaluation. Furthermore, expert provides a consistent evaluation of the utilized software testing tools. [17] proposed evaluation criteria that can be used by both management and technical teams for the selection of appropriate software automation testing tools.

Hence, the objective of this paper is to evaluate mostly used automation tools in software testing and to do a comparison among these tools based on certain criteria to highlight their performance and where they can be used to proffer solutions in different environment, so it can be beneficial to both researchers and practitioners.

2. Materials and Methods

Selecting a perfect tool for testing of a software could be a tasking sometimes considering the factors to put into consideration before choosing a tool. The decision to choose a test tool is a basic factor in the achievement of test automation. This requires studying the extent of testing and test methodology, then afterward selecting the correct test tool that meets the necessities of automating test-suite for a specific item and release [18]. A testing tool can serve for web application testing, desktop application, mobile application testing or combination of two applications also it may involve any testing functionality like unit test, regression test, integration testing etc. The tools evaluated below were selected based on an inclusion and exclusion criteria of the most discussed testing tools from literature which can also be considered as the widely used tools by industry practitioners. These tools are briefly introduced and a tabular comparison of their strength and weakness highlighted based on certain factors like reusability, reliability and cost etc., these factors were identified from literature and used as a base for comparison.

2.1 Selenium

Selenium is a free and open source testing system that performs web application testing. It works crosswise over different browsers and platforms like Windows, Mac, and Linux. Selenium encourages the testers to compose tests in different programming language such as Java, PHP, C#, Python, Groovy, Ruby, and Perl. Selenium suite includes 4 essential segments; Selenium IDE, Selenium RC, WebDriver, Selenium Grid, it is planned in a manner to help and energize automation testing of functional parts of online applications. It can likewise be utilized to perform black box testing on the web application [12, 19, 20].
Some advantages of selenium include its ease of use, flexibility, enabling users to debug and set breakpoints in test cases, test can be changed into diverse programming languages which aids testing for dynamic web applications. Major challenges with selenium are; it does not support conditional and iteration statement, database testing and no error handling capability, a level of programming skill is also required to write test cases [19].

2.2 TestComplete

TestComplete is an automated testing tool used for functional testing, it was created by SmartBear Software. TestComplete enables testers to generate tests for Windows, Web, Android, and iOS applications. These tests can be recorded effectively. Automated scripts can be composed from Python, C++Script, VBScript, Jscript, or JavaScript languages as well. It supports different testing types and methods, for example; functional testing, unit testing, and GUI testing. Making catchphrase utilized for tests are visual, simple and does not require any programming abilities. Scripting requires comprehension of scripting directions, however enables the tester make all the more dominant and adaptable tests [21]. Unlike selenium, test complete is not open source, it requires license to be able to use after the free 30 days trial.

2.3 Ranorex

Ranorex is a testing tool that is easy, thorough and cost-effective for automatic testing. It is a better solution to some other testing tool, as it tests applications from the viewpoint of a user. Ranorex covers testing in desktop, web, and mobile applications, it supports a variety of platforms, including Android, iOS and Windows. Test scripts are written in a number of languages, including C#, Python, C++, VB.net and XML. Ranorex is made up but not limited to the following features: reusable test codes, integration with various tools, GUI recognition, record and playback, bug detection etc. [20, 22]. It gives some additional capacity for conducting regression testing and guarantee the reusability of test activities; It also carries out platform compatibility testing to ensure excellent quality software. A significant benefit of Ranorex is that it is user-friendly and inexpensive and can be used by any testing team and organisation [19, 23].

2.4 Appium

Appium offers a totally fresh revolution in automation testing that promises effective, bug-free and high-quality apps, which saves a project time, cost and labor. Appium is an open source, cross-platform mobile testing tool that enables developers to write tests on various platforms such as iOS and android. It has three primary parts: Appium server, Inspector and Doctor. Appium supports multiple languages like java, python, ruby, JavaScript etc, which enables developers having various expertise to effectively make use of the tool. Appium centers around testing user interaction with mobile web apps content. Test results are utilized to assess accuracy and user involvement with the mobile application as well as ease of use or availability of the features [23, 24].

2.5. Quick Test Professional

QTP (Quick Test Professional) a Windows based programming testing tool used to test applications on desktop or the web, it offers testing automation for regression and functional testing, developed by Hewlett Packard (HP). QTP utilizes Visual Basic Script language to run the scripts and to enlist the test forms, it also works on different objects and controls testing of applications [19, 20]. QTP offers the sector with a useful solution for automating regression testing
and functional testing, addressing every significant implementation and environment of software. Quick Test Professional is generally used for UI-based test case automation, but some non-UI-based test cases such as file system activities and database testing can also be automated [12].

QTP comes with an IDE that is user-friendly and simple hence makes it simple to be used and understood by non-programmer enabling them create and add test cases with ease. It supports various addins like Java, Oracle, SAP, but a major disadvantage is its license and maintenance costs are high and multiple threads or instances cannot be executed. QTP runs slowly compared with open source tools such as selenium [19]. It has been renamed to HPE Unified Functional Testing.

2.6. OpenScript

Openscript is an IDE for Oracle Application Testing Suite, this IDE had been based on Eclipse as a module to help creation and upkeep of Functional and Load testing automated scripts. openscript can be utilized to make test scripts for various purposes like useful functional automation testing, database testing, web administrations testing, load testing. The tool has the ability to record the applications in various browsers like Internet Explorer and Mozilla Firefox and Google chrome. As openscript is created over the eclipse framework and has java as its scripting language there are no restrictions for the capacity. It can likewise be reached out to high states. It has profoundly easy to use UI features which lets the non-software engineers to use it easily. Openscript can't automate windows-based applications that are either written using VB, Dot Net, VC++, etc.[25]

2.7. Janova

Janova is a web-based, automated software testing that runs securely in the cloud. This tool does not require any scripts to be written, it includes the utilization of basic English-based tools that streamline the task of software implementation with effective and simple to utilize tools, hence it is easy to be used by programmers and non-programmers. There is no such software to download and consequently no infrastructural venture is required. Since it is utilized in the cloud, it has a fast and simple setup that incorporates no installation, its speed of execution is also faster than the traditional web testing tools. Even though it is not open source, its license cost is not expensive [22].

2.8. Rational Functional Tester

Rational Functional Tester (RFT) was created by IBM, it is an object- oriented programming based automated testing tool which involves regression, functional, GUI, and data-driven testing. It works with Java, Web based, and Microsoft Visual Studio, .NET applications. RFT ensures test plans and cases are maintained and executed properly by the Quality Assurance departments of organizations, it also identifies areas of application that were and were not performed during the testing [22].

The choice of specific automated testing tools is based on the application type that is to be tested, the complexity of the project, the cost related with the tool to be used and the budget of testing stage in the organization
3. Results and Discussion

After review of the literatures on the following tools, a detailed description of these tools is presented in Table 1 below, based on certain criteria selected from literatures this are recommended by researchers and beneficial to industry practitioners in selection of tool to be used for testing. The table show the tools discussed, and the criteria satisfied by each tool. This analysis would help industry practitioners select the best tool required to test software; be it large-scale or small-scale project.

Table 1: Tools and Criteria

| Tool/ Criteria                        | Open Source | License | Supporting Platform (Mobile) | Supporting Platform (web) | Supporting Platform (Desktop) | Learning Ease/ Ease of Use | Coding/ Programming Skills | Code Reusability | Test Results Report | Record & Playback |
|---------------------------------------|-------------|---------|-------------------------------|---------------------------|-----------------------------|---------------------------|---------------------------|-------------------|-------------------|------------------|
| Selenium [12, 19, 20]                 | ✓           |         | ✓                            | ✓                         | ✓                           | ✓                         | ✓                         | ✓                 | ✓                 | Plug In           |
| TestComplete [21]                     | ✓           | ✓       | ✓                            | ✓                         | x                           | ✓                         | ✓                         | ✓                 | ✓                 |                  |
| Ranorex [19, 20, 22, 23]              | ✓           | ✓       | ✓                            | ✓                         | x                           | ✓                         | ✓                         | ✓                 | ✓                 |                  |
| Appium [23, 24]                      | ✓           | ✓       | ✓                            | ✓                         |                             | ✓                         | ✓                         | ✓                 | ✓                 |                  |
| Quick Test Professional [12, 19, 20]  | ✓           | ✓       | ✓                            | ✓                         | x                           | ✓                         | ✓                         | ✓                 | ✓                 |                  |
| OpenScript [25]                      | ✓           | ✓       | ✓                            | ✓                         | x                           | ✓                         | ✓                         | ✓                 | ✓                 |                  |
| Janova [22]                           | ✓           |         | Cloud based                  |                           | x                           | ✓                         | ✓                         | ✓                 | ✓                 |                  |
| Rational Functional Tester [22]       | ✓           | ✓       | ✓                            | ✓                         |                             | ✓                         | ✓                         | ✓                 | ✓                 |                  |

From the analysis of tool in table 1, we find out that majority of the tools are licensed based which implies, for them to be used for testing, the organization or the testers must subscribe to them, although some offer free days trials. The opensource tools are free and can be used without subscription. The applications and platforms each tool support can also be seen in the table, some tool support all three platforms (Mobile, Web, Desktop) while some support just one. These show the kind of applications these tools can be used to test. Some tools are easy to use and learn while others require programming skill to be able to use them effectively. These criteria should be considered when selecting tools for testing in an organization, these would save time as well as cost incurred during testing. If the software or project to be tested is a small one, then open source tools can be used instead of incurring more cost in purchasing the licensed tool.

From the criteria selected in the table, 7% of the tools examined were found to be open source while 12% were licensed tools, this implies that majority of the software testing tools are licensed which incurs more cost during the software testing phase. More testing tools examined supported web platform testing from the analysis carried out with 17%, tools supporting desktop platforms...
were 10% while those supporting the mobile platform was 8%, making more tools examined web-based testing tool even though some tools like testcomplete and ranorex performs well across all three platforms. Janova was the only tool among the testing tools examined which is cloud based even though it is supports web platform testing. Learning ease or ease of use the tools were highlighted in three tools making up for 7% of the criteria considered.

Only Selenium testing tool was discovered to require knowledge of programming language to effectively use the testing tool, this made up 3% of the criteria used for the tool analysis. Five testing tools supported the ability to reuse the same code for similar testing instead of having to rewrite the codes from scratch, this encompassed 12% of the testing criteria. 10% of the testing tools had the capacity to document test results and generate reports after every test, selenium testing tool do not support this feature, but requires the use of plug-in to be able to generate test results and reports. Finally, 17% of the tools examined supported the record & playback feature, which helps improve efficiency when using a testing tool, it records the actions of the testers during testing and performs playback of these script when required.

Figure 1 & 2 below gives a summarized diagrammatic representation of table 1 above and the results obtained from it.

![Pie Chart Of Tools Composition](image-url)
As seen in table 1 above, the evaluated testing tools works on various platforms and provide different capabilities when it comes to testing applications, table 2 gives a detailed summary of these tools and the testing type each of them supports.

Table 2: Tools Testing Type

| Tool            | Testing Type                                      |
|-----------------|---------------------------------------------------|
| Selenium        | Functional testing                                |
| TestComplete    | Functional testing, Graphical User Interface testing, Unit testing |
| Ranorex         | Graphical User Interface testing, Compatibility testing |
| Appium          | Graphical User Interface testing, Functional testing |
| Quick Test      | Functional testing, Regression testing            |
| Professional    |                                                   |
| OpenScript      | Functional testing, Load testing, Database testing |
| Janova          | Functional testing                                |
| Rational Tester | Functional testing, Regression testing, Graphical User Interface testing |

4. Conclusion

Software testing is a significant part of the life cycle of software development. It guarantees that the software been deployed to the market is free from error and effects. To meet up with market demands and time factor, software testers has employed the use of automated testing tool to carry out testing, this is used over the manual approach to testing, as automated testing saves time and also minimize cost incurred in the organization during the testing phase. In this paper, we evaluated some automated testing tools which includes selenium, testcomplete, Ranorex, OpenScript, Janova, etc, highlighting their basic features and characteristics. All testing tools are efficient to be used for testing but depending on the scenario, some may tend to be more efficient
than the others, this paper highlights the details of some selected testing tool and their testing types in tabular format, their licensing, the platform they best operate, the tools that support code reusability among others.

This study would be useful to the industry expertise to know which tool to best use for a particular project and for researchers, more tools can be compared and more criteria for comparison highlighted, this would avail software testers the ability to choose perfect tools for testing applications with ease, this would save more time and minimize cost when the right testing tool is selected. We can therefore conclude that there is no one perfect tool for testing, but for a particular testing purpose, tradeoffs can be made to select the best tool depending on the size of the project, the budgeted cost for testing, the platform of the application and also the language that is used to develop the project.

5. Recommendation

Quality is the primary focus of any project in software engineering, we therefore recommend that in selecting any tool, the size of the project and the cost budgeted for testing should be considered, also the platform where that project would be used should reflect in the criteria for selecting a testing tool. From the result obtained in this study, we recommend TestComplete and Ranorex testing tools for testing across all platforms and when the project to be tested is large, since they are both licensed tools, the budget for testing should be put into consideration. Appium is recommended for strictly mobile testing, while selenium for web testing with the advantage that it is open source. The study is also recommended for future work, where more tools and criteria can be covered.

Acknowledgement

We would like to acknowledge the support and sponsorship provided by Covenant University through the Centre for Research, Innovation, and Discovery (CUCRID).

References

[1] Pressman, R. S. (n.d.). *Software engineering*. New York: McGraw-Hill Book Company.

[2] Chauhan, R. K. & Singh, I. (2014). Latest Research and Development on Software Testing Techniques and Tools, 4(4), 2368–2372.

[3] Wala, T. & Sharma, A. K. (2014). Improvised Software Testing Tool, 3(9), 573–581.

[4] Merina, C. (2019). Tool Usability Parameter in Determining the Performance of Software Testing Tool, IJTB (International J. Technol. Business), 3(1), 8–18.

[5] Misra, S., Adewumi, A., Maskeliūnas, R., Damaševičius, R. & Cafer, F. (2018). Unit Testing in Global Software Development Environment, *Data Science and Analytics*, 309-317, DOI: 10.1007/978-981-10-8527-7_25.

[6] Tassey, G. (2002). “The Economic Impacts of Inadequate infrastructure for software testing”. *Nat. Inst. Stand. Technol.*

[7] Raulamo-Jurvanen, P., Mäntylä, M. & Garousi, V. (2017). Choosing the Right Test Automation Tool, in *21st International Conference on Evaluation and Assessment in Software Engineering (EASE 2017)*, 21–30.
[8] Jain, V. & Rajnish, K. (2018). Comparative Study of Software Automation Testing Tools: OpenScript and Selenium, *Int. J. Eng. Res. Appl.*, 8(2), 29–33. https://doi.org/10.9790/9622-0802032933

[9] Odun-Ayo, I., Falade, A., & Samuel, V. (2018). Cloud Computing and Open Source Software: Issues and Developments, *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists* 2018, 14–16 March, 2018, Hong Kong, 140-145.

[10] Uppal, N. & Chopra, V., (2012). Design and Implementation in Selenium IDE with Web, *Int. J. Comput. Appl.*, 46(12), 8–11.

[11] Santos, R. & Gehrke, K. (2017). Oracle ® Functional Testing OpenScript Programmer’ s Reference.

[12] Kaur, H. & Gupta, G. (2013). Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete, *Int. Journal of Engineering Research and Applications*, 3(5), 1739–1743.

[13] Mustafà, K. M., Al-Qutaish, R. E., & Muhairat, M. I. (2009). Classification of software testing tools based on the software testing methods, in *2009 International Conference on Computer and Electrical Engineering, ICCEE 2009*, 1, 229–233.

[14] Kolli, R. (2016). An Empirical Study on Software Test Estimation.

[15] Silverstein, M. (2003). Logical capture/replay. *STQE Magazine*, 5(6), 36–42.

[16] Raulamo, P., Hosio, S. & Mäntylä, M. V (2019). Practitioner Evaluations on Software Testing Tools. In *Proceedings of the Evaluation and Assessment on Software Engineering*, 57–66.

[17] Vos, T., Marin, B., Panach, I., Baars, A., Ayala, C., & Franch, X. (2011). Evaluating software testing techniques and tools. *Actas de Las 16th Jornadas de Ingenieria Del Software y Bases de Datos, JISBD 2011*, pp. 531–536.

[18] Kannan, S. & Pushparaj, T. (2014). A Study on Variations of Bottlenecks in Software Testing, *International Journal of Computer Sciences and Engineering*, 2(5), 8 - 14.

[19] Malik, V. & Gahlan, M. (2016). Comparative Study of Automated Web Testing Tools, *Int. J. Latest Trends Eng. Technol.*, 6(3), 1- 8.

[20] Top 10 Testing Automation Tools for Software Testing, pp. 1–4.

[21] Mohammad, D. R., Al-Moman, S., Tashtoush, Y. M., & Alsmira, M. (2019). A Comparative Analysis of Quality Assurance Automated Testing Tools for Windows Mobile Applications, 2019 IEEE 9th Annu. Comput. Commun. Work. Conf., 414–419.

[22] Chauhan, R. K. & Singh, I. (2014). Latest Research and Development on Software Testing Techniques and Tools, *International Journal of Current Engineering and Technology*, 4(4), 2368–2372.

[23] Babar, M. I., Jehanzeb, M., Khan, M., & Iqbal, S. (2017). A Comparative Analysis of Quality Assurance of Mobile Applications using Automated Testing Tools, *International Journal of Advanced Computer Science and Applications*, 8(7), 249-255.
[24] Shah, G., Shah, P. & Muchhala, R. (2014). Software Testing Automation using Appium, *International Journal of Current Engineering and Technology*, 4(5), 3528–3531.

[25] S. P, what is openscript, *Testing Tools*, (2019). [Online]. Available: http://www.testingtools.co/oats/what-is-openscript. [Accessed: 24- May- 2019].