JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A COMPARATIVE STUDY OF AUTOMATION TESTING TOOLS FOR MOBILE, WEB AND DESKTOP APPLICATIONS

Pragya Sen
Department of Electronics and Communication, R.V College of Engineering, **Bangalore, India**

Savitri Tangirala
Department of Computer Science, R.V College of Engineering,
Bangalore, India

Dr. Sindhu D V
Faculty of Computer Science, R.V College of Engineering,
Bangalore, India

Dr. Saba Farheen. N. S
Faculty of Electronics and Communication, R.V College of Engineering,
Bangalore, India

Abstract - **Software testing is an essential part of the development process of Software tools and applications. It ensures that the designed software is free of defects and meets all of its functional requirements. Automation testing tools can make this process more efficient and increase test coverage as compared to manual testing. An extremely important part of this process is selecting the right testing tools and frameworks based on the requirement of the product being developed. Mobile, Web and Desktop applications have different requirements when it comes to testing leading to different testing tools for each type of application with some tools providing support for more than one type of application. This paper provides a comparative view of different test frameworks and tools for the three types of applications mentioned above.**

Keywords- automated testing, image-based testing, test development, scripting language

I. INTRODUCTION

High-quality software development is ensured by efficient software testing. Before a software release, testing is meant to find bugs and feature issues. The process of manual testing involves test engineers executing tests one at a time and independently. Automation testing involves the use of scripts and tools to enable quicker product testing. On analyzing various metrics of tests written manually and automated tests, it is seen that there is a trade-off between the amount of time and effort a test engineer invests in developing test cases as compared to the production of input assets needed for automation [1]. Automation testing can be used for the efficient conduction of different types of Functional Tests [2]. A broad classification of Software testing techniques include White Box, Black box and Grey box testing. White box testing requires the understanding of each aspect of the internal structure of the software application/tool under test. In-depth grasp of the source code is an essential part of it [3]. Black box testing requires the QA analyst to have only a surface-level understanding of the code and does not need to deal with the intricacies of the internal logic and implementation of the program being tested. It focuses on the external aspects of the application and on finding bugs that could hinder end-user experience [4]. Grey Box Testing is a technique intermediate to White Box and Black Box testing. Here, only limited information about the internal workings of the program is required. The tester will have access only to certain internal components of the source code that are necessary for generating test cases [5].

II. BACKGROUND

Different types of applications have different requirements based on their source, the platform they run on, etc. Thus, test tools for different types of applications also have different requirements.

Mobile Applications: Mobile devices have a limited interface design due to reduced screen size. Scalability and interactivity are some of the problems that developers run into when testing usability. Mobile devices come in a variety of forms such as phones, tablets, etc. The testing tool should be capable of handling variation in shapes and sizes of different mobile devices [6].

Web Applications: The web server hosts web-based applications, making them accessible to anybody with an internet connection. Web-based applications are dependent on different browsers, making consistent usability of the test tool crucial [7].

Desktop Applications: Desktop applications refer to softwares that run locally on a computer requiring installations for usability. Setups and teardowns are a tricky aspect of Desktop applications [8]. Unlike Web applications where setup and teardowns usually involve the opening and closing of a browser leading to no existing cookies, cache, etc. for the next test case, just restarting a Desktop application does not lead to a clean slate [9]. Numerous "residual" files could be produced by a desktop application. These could range from licenses and registration files to registry entries. The removal of such files involves some additional work that should not be neglected during the teardown of the tests.

III. LITERATURE REVIEW

Extensive research has been done for the testing of different types of applications over the years. Some applications support testing of a single type of application. However, over the decades, a lot of testing tools have been updated to support more than one type of application. In this paper we have conducted an extensive literature review to summarize the main functionalities, updates, supported platforms, etc. of commonly used test tools for Mobile, Web and Desktop Applications.

Web Applications:

A. Selenium

Selenium is a suite of automation tools that includes the Selenium IDE, Selenium webdriver, Selenium RC, and Selenium grid [10]. Selenium IDE is a Firefox plug-in that allows the development of test cases. It is an integrated development environment that allows the recording and playback of actions performed on the web and the generation of tests. Selenium RC is another part of the Selenium test suite that supports multiple programming languages for automating UI tests for web applications against any HTTP website. Ajax applications are not supported by it which resulted in the development of the Selenium WebDriver which is another component of the Selenium test suite which is currently the most widely used tool for the automation of Web Applications. Along with providing support for Ajax applications and multiple web browsers, it is also more efficient in contrast with Selenium RC since it has direct communication with the browser [10]. It allows test execution even when the browser is minimized and supports multiple programming languages including Ruby, Python, Kotlin, JavaScript, C# and Java. It also provides multi-browser support including Chrome, Safari, Firefox and Internet Explorer. A major drawback of Selenium is that it cannot handle windows alerts and does not allow parallel execution of tests on the same hardware [11].

B. Sikuli

Sikuli is a test-tool for Web applications that allows image-based automation. The scripting system of Sikuli allows users to use images of graphical elements on screen and have access and control of these elements during the test-run. It supports JavaScript, Python (Jython) and Ruby. Jython is the Java implementation of Python that has a scripting-style like Python while providing full access to Java libraries. Some interfaces do not allow visual interaction with users thus forcing them to rely on non-visual alternatives such as XPaths and other locators [28]. Full XPaths happen to change within short periods of time with even minor updates in the Web Application. Thus identifying and accessing elements of the Web Application through images can help in making the developed test more robust and less prone to failures as long as the visual appearance of the identified elements do not change. But if there happens to be even slight changes in the appearance of the elements due to updates in the application, the test script would fail [27]. Along with Web applications, Sikuli also supports the automation of Windows Desktop Applications but it is most commonly used for Web Applications. As seen in the paper by Lathwal and Ashish, Sikuli can be integrated with Selenium for overcoming the drawbacks of both [25].

C. Cypress

Cypress is an end-to-end testing tool for modern web test automation and is JavaScript-based. Automated web tests can be written with the help of this tool. It is designed for developers and operates directly in the browser utilizing a DOM manipulation approach [30]. Automation developers have stated that a significant amount of time for error-handling was spent on the coordination of wait with the network speed, excessive load and other reasons that could lead to the slow loading of web-pages. Cypress was created and established in 2015 to tackle this issue. The declaration of implicit and explicit waits is not necessary with Cypress because the framework includes automated waiting. It automatically waits for DOM loading, animation, elements, and other events. Additionally, it operates within the web-browser having direct interaction with the application being tested [29]. Cypress can be used for different browsers and currently provides support for Firefox, WebKit and members of the Chrome family. Some of the limitations of Cypress are that it only supports JavaScript for test case scripts and it does not provide support for multiple tabs [26].

Mobile Applications:

A. Appium

It is the most commonly used automation tool for running tests on Mobile devices. Apart from native Mobile applications, it can also be used for Web and Hybrid applications that run on Mobile devices. It supports testing for applications on Android, iOS and Firefox OS platforms [12]. Unlike other third-party tools that require SDKs and HTTP Servers to be embedded in the application and use private APIs, Appium tracks activity on devices with the help of bootstrap.js. which results in the Appium server receiving the test execution results which then sends it to the Appium Client. Appium inspector is a tool that can be used for locating and tracking the elements of an Application, for simulating manual operations and recording test scripts of the App activity [25]. All programming languages supported by Selenium are also supported by Appium along with additional languages like PHP. Although it was first intended for mobile apps, new updates allow the testing of Windows Desktop Applications too [23]. The main disadvantage of Appium is that tests can run slightly slower than other testing tools because of its dependency on the remote web driver [13].

B. Robotium

This framework is made to offer Android applications black box tests. This implies that the test is for anticipated outcomes rather than particular techniques [15]. Robotium has a better execution time than Appium. The use of Robotium is not advised for testing tasks requiring the opening of other applications, changing connection types, or uninstalling software [14]. It provides support for Android API 1.6 and higher. One of its major disadvantages is that it provides support only for one programming language, which is Java, since Android Apps are developed using Java. Robotium Recorder can be used for recording tests while interacting with the application but it increases testing expenditure since it is a licensed software [16].

Desktop Applications:

A. Katalon Studio

Along with Desktop Applications, Katalon Studio also provides support for Mobile and Web Applications. All Desktop applications written in the platforms Win32, WinForms, WPF, and UWP are fully supported by it [17]. Automatic testing of user interface elements, such as pop-ups, iFrames, and wait times, is possible with Katalon Studio. Linux, macOS, and Microsoft Windows are all supported by the tool. The key benefits of Katalon are its effortless setup and its ability to work well with an extensive range of automation tools. Users with varying degrees of programming skill can use Katalon without hassle due to its twin scripting interfaces which allow QA analysts with relatively lower technical skill to work with the more basic program that is free of coding. The option for advanced users provides various code-enhancing services. Native testing, as well as concurrent and successive executions, are supported by Katalon Studio. Groovy, a Java-like programming language, is used to run it [18]. A major disadvantage of Katalan Studio is that it does not support any other programming language which leads to only users comfortable with Java being able to make full use of it. It is not an open source tool [19].

B. Ranorex

Ranorex provides support for Desktop, Web and Mobile applications. It provides support for over 10 platforms including Winforms, WPF, etc. It provides GDI and GDI+ plugins for the challenging areas [20]. It has a very advanced image based processing and can support any primarily graphic based environment [18]. C# and vb.net are the two programming languages that Ranorex supports. Since it is written entirely in.net code, learning a scripting language is not required in order to use it [21]. Any change in the user interface can be automatically detected by the Ranorex smart object identification technique. Some of the drawbacks of Ranorex include that it does not provide support for MacOS and supports only two programming languages [22].

Test tool/ framework	a	b	c	d	e
Selenium	6	Apache License 2.0. (Open source)	no	hard	3
Sikuli	3	MIT Licence (Open Source)	yes	hard	2
Cypress	1	MIT Licence (Open Source)	yes	easy	1
Appium	6+	Apache License 2.0. (Open source)	yes	hard	3
Robotium	1	Apache License 2.0. (Open source)	no	easy	1
Katalon Studio	2	Freeware (Free license)	yes	easy	3
Ranorex	2	Node-locked and floating (paid licenses)	yes	easy	3

Table 1: Analysis of testing tools on various metrics, a: Test development platforms/ programming languages supported, b: License type, c: Image based testing, d: Ease of setup, e: No. of modes supported (Web, Mobile and Desktop)

IV. CONCLUSION

As can be seen in this paper, there is no one-shoe-fits-all solution for choosing the right testing tool since there is no tool with all performance metrics better than other tools. Different tools are suitable for different scenarios. This comparative study aims to summarize the pros and cons of each test tool to aid the process of choosing a test tool based on the requirement of the test engineer as this is an extremely crucial step of the Testing process of any product.

V. FUTURE SCOPE

As manual testing is an extremely time consuming part of the testing cycle, automated testing is the way of the future. It significantly increases test coverage as compared to manual testing which can be a very crucial factor for supporting new innovations while also keeping up with the pace of the market demand. Automated tests are currently in use to a certain extent. In a study conducted, 77% of the companies mentioned that they have used automated testing softwares yet only a little more than 24% of the companies had automated 50% or more of their test cases [24]. While automated testing tools are effective and can save a lot of time invested in software testing, the concept of automated testing is still relatively new and not as widely used as it can be. One of the major drawbacks of these tools being the frequent requirement changes in applications which can cause stability issues in test cases. We think that in the near future, most of these testing tools will be extremely robust and capable of handling regular changes to the applications under test.

REFERENCES

[1] S. Karlsson, A. Čaušević, D. Sundmark and M. Larsson, "Model-based Automated Testing of Mobile Applications: An Industrial Case Study," 2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Porto de Galinhas, Brazil, 2021, pp. 130-137, doi: 10.1109/ICSTW52544.2021.00033.
[2] Ateşoğulları, Dilara & Mishra, Alok. (2020). AUTOMATION TESTING TOOLS: A COMPARATIVE VIEW. International Journal of Information and Computer Security. 12. 63-76.
[3] Ehmer, Mohd & Khan, Farmeena. (2012). A Comparative Study of White Box, Black Box and Grey Box Testing Techniques. International Journal of Advanced Computer Science and Applications. 3. 10.14569/IJACSA.2012.030603.
[4] GeeksforGeeks. (2023, April 19). *Differences between Black Box Testing vs White Box Testing*. https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/ [5] GeeksforGeeks. (2022, July 19). *Gray Box Testing | Software Testing*.
https://www.geeksforgeeks.org/gray-box-testing-software-testing/
[6] Anureet Kaur, "Review of Mobile Applications Testing with Automated Techniques," 2015 International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)
[7] BrowserStack. (2021, Feb 24). *Difference between Mobile and Web Application Testing*
https://www.browserstack.com/guide/differences-between-mobile-application-testing-and-web-application-testing
[8] Katalon. (n.d.). *Desktop Testing* https://katalon.com/desktop-testing
[9] Leapwork. *Differences in Automating Web and Desktop Application Testing*. https://www.leapwork.com/blog/differences-automating-web-and-desktop-application-testing[10] Satish Gojare, Rahul Joshi, Dhanashree Gaigaware, Analysis and Design of Selenium WebDriver Automation Testing Framework, Procedia Computer Science, Volume 50, 2015,
Pages 341-346, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.04.038.
[11] Lathwal, Ashish. (2019). A Literature Review on Automation Testing Using Selenium+Sikuli. International Journal of Distributed Artificial Intelligence. 11. 35-40. 10.4018/IJDAI.2019070104.
[12] Appium. (n.d.). *Appium Documentation*. http://appium.io/docs/en/2.0/
[13] Tutorials link. (2022, February 21). *Advantages and Disadvantages of Appium*. https://tutorialslink.com/Articles/What-are-the-advantages-and-disadvantages-of-Appium-and-its-working-Software-Testing-Tool/3255
[14] A. M. Sinaga, P. A. Wibowo, A. Silalahi and N. Yolanda, "Performance of Automation Testing Tools for Android Applications," 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), Bali, Indonesia, 2018, pp. 534-539, doi: 10.1109/ICITEED.2018.8534756.
[15] CodePath. (n.d.). *Testing with Robotium*. https://guides.codepath.com/android/ui-testing-with-robotium
[16] SOFTWARETESTER.NET. (2022, February 7). *Robotium vs. Appium*. https://softwaretester.net/robotium-vs-appium-2/
[17] Katalan (n.d.). *Introduction to Desktop app testing in Katalon Studio*. https://docs.katalon.com/docs/create-tests/introduction-to-test-creation/introduction-to-desktop-app-testing-in-katalon-studio
[18] N. Srivastava, U. Kumar and P. Singh (2021) Software and Performance Testing Tools. Journal of Informatics Electrical and Electronics Engineering, Vol. 02, Iss. 01, S. No. 001, pp. 1-12, 2021.
[19] AltexSoft. (2021, June 15). *Pros and Cons of Katalon Studio* https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-katalon-studio-automation-testing-tool/
[20] Ranorex (n.d.). *Windows Desktop Test Automation*. https://www.ranorex.com/windows-desktop-test-automation/
[21] V.Sangeetha and T.Ramasundaram, "Optimizing Whole Test Suite Generation," 2016 International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)
[22] AltexSoft. (2018, Nov 12). *The Good and the Bad of Ranorex GUI Test Automation Tool* https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-ranorex-gui-test-automation-tool/
[23] Digital.ai. (2020, May 5). Comparing the Top 4 Android Testing Tools https://digital.ai/catalyst-blog/comparing-the-top-4-android-testing-tools/
[24] DogQ. (2022, Sept 27). Test Automation Statistics for Making the Right Decisions. https://dogq.io/blog/test-automation-statistics-for-making-the-right-decisions/

[25] J. Wang and J. Wu, "Research on Mobile Application Automation Testing Technology Based on Appium," 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Jishou, China, 2019, pp. 247-250, doi: 10.1109/ICVRIS.2019.00068.
[26] Browser Stack. (2023, February 14). *Cypress vs. Selenium*. https://www.browserstack.com/guide/cypress-vs-selenium
[27] Sun, Jin-lei & Zhang, Shi-wen & Huang, Song & Hui, Zhanwei. (2018). Design and Application of a Sikuli Based Capture-Replay Tool. 42-44. 10.1109/QRS-C.2018.00021.
[28] Tom Yeh, Tsung-Hsiang Chang, and Robert C. Miller. 2009. Sikuli: using GUI screenshots for search and automation. In Proceedings of the 22nd annual ACM symposium on User interface software and technology (UIST '09). ACM, New York, NY, USA, 183-192.
[29] Mobaraya, Fatini & Ali, Shahid. (2019). Technical Analysis of Selenium and Cypress as Functional Automation Framework for Modern Web Application Testing. 27-46. 10.5121/csit.2019.91803.
[30] Cypress. (n.d.). *Cypress Documentation*. https://docs.cypress.io/guides/overview/why-cypress