GUI Automation Testing Tools: a Comparative Study

Ali Abdalftah Fadul Mohammed [1] Prof. Khaled Ahmed Ibrahim[2]

1 Faculty of Computer Sciences & Info. Technology Nile Valley University

2Faculty of Computer Sciences & Info. Technology Karary University

Corresponding Author:

Email: Aliftah@nilevalley.edu.sd

Abstract

 Tools for automating GUI testing in web applications play a crucial role in the software development process by improving efficiency, ensuring quality, and reducing costs. In this study, twenty-five GUI automation testing tools were evaluated and compared according to fifteen criteria. The results reveal the excellence of some tools in specific areas, highlighting the importance of choosing the right tool based on project requirements. Selenium excels in cross-browser testing with its open-source framework, its robust community support and extensive integration options make it a top choice for developers seeking efficient and scalable testing solutions, while Applitools stands out with its AI-powered visual recognition capabilities. The study assists researchers, developers, and testers in making informed decisions when selecting the most suitable automation tools for their projects, simplifying the choice among many available options and ensuring the successful achievement of software project goals. The selection process, guided by the features of each tool, is critical to the success of the software project. The more we know about the characteristics of each GUI testing automation tool, the better we can choose the right tool for testing.

Keywords: Automation of GUI Testing, Automation Testing Tools, GUI Testing for Web Application.

1. Introduction

With the widespread use of software in the information era and with the increased dependability of people on the software, there was a need for the use of highly efficient software that meets the requirements of users, leading to the emergence of software testing science to ensure the quality of the software before delivery to the end user.

Automation testing of software came to facilitate the task of the software tester and to reduce the time and cost of the testing process.

The software graphical interface provides the best way to interact with the user, showing all the information and interactive elements such as buttons, text boxes, drop-down menus, radio buttons, and selection boxes. So it must be ensured that all its functions work well before delivery to the end user. The user interface is just as important as the code when developing software. A well-designed user interface can lead to customer satisfaction. For this reason, it is necessary to conduct comprehensive testing of the GUIs before releasing the software.

There are many software test automation tools with different purposes, most of which share the main goals of test automation, such as improving test coverage, the possibility of reusability and iteration of test scripts, detecting errors before the system reaches the end user, reducing the cost of the testing process, and reducing the need for a human factor during testing, which reduces the risks in Hazardous systems. Day after day the importance of automating functional and nonfunctional software testing is increasing, the use of test automation tools has improved significantly by testers, which indicates an increased demand for the use of test automation tools in the future, which motivated the researchers to compare, analyze and evaluate the test automation tools currently in the software market from different aspects due to the difficulty of choosing the appropriate automation test tool and framework. This scientific paper aims to provide a comparative analytical presentation of the most important graphical user interface test automation tools, as this paper contributes to creating knowledge in this modern trend in the software market for both academic researchers, developer, QA engineer, project manager and software testers.

The structure of this paper is organized as follows: Section 2 presents the previous studies related to the research topic. Section 3 describes the GUI automation testing tools selected for this research and provides a brief overview of each. Section 4 briefly introduces the criteria on which the automation tools were compared and gives a short description of them. Section 5 discusses the analysis of a comparison table of GUI automation testing tools, highlighting unique features and common features. Section 6 concludes this research.

2. LITERATURE REVIEW

A. Importance of Automation Tools in GUI Testing

(sen, savitri trangirala, sindhu D V, & saba farheen N S, 2023) Highlighted the importance and effectiveness of automation tools in software testing, emphasizing how these tools make the testing process more efficient and increase test coverage compared to manual testing. The study focuses on the significance of software testing in software development and ensuring its quality. The appropriate tools and frameworks are selected based on the specific requirements of the developed product. Mobile, web, and desktop applications have different needs, leading to the use of various testing tools for each type, with some tools supporting more than one application type. The study concludes that choosing the right tools for test automation depends on the type of application and its specific requirements. Popular tools for web application testing include Selenium and Cypress, while mobile application testing tools include Appium. It is observed that automation tools significantly enhance test efficiency and coverage, but they must be selected carefully to ensure they align with the specific application requirements.

(At Thooriqoh, Tiara Nur Annisa, & Umi Laili Yuhana, 2021) Present a systematic literature review on the use of the Selenium framework in web automation testing. Selenium is an open-source framework used for automating web application tests and can be used with several programming languages such as Python, Java, and Ruby. The study aims to summarize previous research related to test automation using Selenium to provide information that helps developers design and implement software tests more effectively. Twenty studies published between 2009 and 2021 were collected and analyzed using the systematic literature review (SLR) methodology. The results showed that Selenium can be used not only for testing application functionalities but also can be applied with technologies such as security and performance testing. The paper recommends focusing on improving the efficiency and maintainability of automation tests using Selenium in future research.

B. **Comparison of GUI Testing Automation Tools**

According to (SAMLI & Zeynep ORMAN, 2023), automation testing in the software testing process is generally required to obtain better test results and save time and money by trying as many test cases as possible.

Web-based software test automation tools differ from each other according to various criteria, the most important of which are cost, licensing, technical support, programming language support, user experience, documentation, and browser support. It is often difficult to select the right web-based automated tool for a particular testing process. Since many of these tools are licensed and expensive, comparing automated tools is an important issue that needs more careful research as each tool has different features and may not provide a useful test for every program.

The tester must decide which tool will be best for testing process. This selection process is done not only by the software features but also by many different criteria. These criteria can relate to the features of the application to be used, financial circumstances, technical support, or anything else. The more we know about tools, the more comfortable we will be with choosing the most appropriate tool.

(Yadu & Narain, 2022) Discuss a comparison between manual and automated testing in software development, with a focus on automated testing tools and techniques, especially in GUI (Graphical User Interface) application environments. Manual testing relies on human intervention to inspect software, which is beneficial when the process requires creativity or a deep understanding of user behavior. However, this type of testing is time-consuming and prone to human errors, making it unsuitable for repetitive tests. On the other hand, automated testing uses specialized tools to execute tests automatically, saving time and effort while reducing human errors. This type of testing is ideal for stable systems and repetitive tests such as regression tests.

Automated tools like Selenium are used for automated GUI testing, where they can record user interactions with the interface and replay them to test software functionality. Other techniques include model-based testing and object-based testing, which rely on direct integration between testing tools and the software under test. The paper concludes that automated testing is more effective and efficient than manual testing in many cases, especially when it comes to reducing time and costs in large projects. However, manual testing cannot be completely eliminated, particularly in scenarios that require deep qualitative analysis. The paper also suggests that combining both approaches is the best way to achieve an optimal balance in the quality and speed of testing. (Kumari, Madhulika Mittal, Simranjeet Kour, & Vikash Sawan, 2022) Compared three tools: Selenium, Quick Test Professional (QTP), and TestComplete. Here are the criteria used for comparison along with key details:

Licensing Cost: Selenium is free, QTP is very expensive ($7260 for ten users), and TestComplete costs $2,000 per license. Application Support: Selenium supports web apps only, QTP supports client-server apps, and TestComplete supports both web and desktop apps.

Language Support: Selenium supports multiple languages (Java, .Net, Perl, PHP, Python, Ruby), QTP supports VBScript and JavaScript, and TestComplete supports VBScript, JScript, DelphiScript, C++Script, and C#Script.

Platform Support: Selenium supports Windows, PC, MAC, and UNIX, QTP supports Windows XP, and TestComplete supports Windows 7 and later.

Programming Skills: Selenium requires programming skills, QTP is easy to use, and TestComplete is suitable for both web and desktop apps.

Usage: Selenium requires expertise, QTP is easy to learn, and TestComplete supports all 32-bit and 64-bit Windows apps. Database Applications: Selenium requires effort for database apps, QTP works well with them, and TestComplete also works well with database apps.

Report Generation: Selenium lacks robust report generation, QTP generates comprehensive reports, and TestComplete provides easy report generation tools.

Plugins/Add-ons: Selenium requires additional plugins, QTP requires licensed add-ons, and TestComplete does not need additional plugins.

Overall: The choice depends on the application type, budget, and required efficiency. TestComplete is good for automation, while QTP is versatile for critical testing scenarios.

According to (Besimi & Betim Cico, 2022), it is difficult to choose a specific framework to use for testing automation. In this study, researchers compared Katalon Studio, Selenium, and UFT in web application testing in order to propose and create a new framework for testing and code optimization. UFT One is an AI feature used to run the same test on different platforms and versions using AI-based object identification where test scripts are more intuitive, technically neutral, and easier to maintain. This feature seems very useful and easy to use, but it is not practical. Needs deep knowledge in Visual Basic. Katalon's AI features can be used by installing two plugins, Auto Healing Smart Path and Appitools. Unlike Katalon Studio and UFT, Selenium does not have any AI-related features. AI can only be integrated via open source libraries. The researchers concluded that it is difficult to get a ranking in software testing tools based solely on comparison results. However, the following aspects must be considered when choosing a test tool, such as: test need, scope of application, required competence, and available budget.

In the modern era, websites and web applications have become essential for all types of businesses and commercial enterprises. Software testing plays a crucial role in the software development life cycle to ensure product quality. To achieve this efficiently, developers use automated testing tools to reduce time and cost compared to manual testing. This study compared automated testing tools for web applications, focusing on Katalon Studio and Selenium. (Pelivani & Betim Cico, 2021) Summarized the advantages and disadvantages of each tool. The researchers discussed the advantages of Selenium, noting that it is open-source and supports cross-browser testing but requires manual setup. Cypress offers fast and reliable tests but only supports certain browsers. Robot Framework is easy to use and supports keyword-driven testing but lacks features like nested loops. WebDriverIO is a stable and fast tool but is not fully customized. Gauge allows writing tests in a flexible language like Markdown but does not support all browsers. The researchers compared Selenium and Katalon Studio in terms of execution time, noting that Katalon Studio takes longer to execute due to using the Groovy language. However, Katalon Studio provides detailed and easy-to-read reports, whereas Selenium requires additional setups to generate reports. The researchers concluded that choosing the appropriate automated testing tool depends on the project's needs and testing requirements.

(Singh Gadwal & Prasad, 2020) Review automation testing tools for software, aiming to provide a comparative literature review of these tools, with a focus on web application testing tools. Various tools were analyzed based on characteristics such as platform support, programming language used, and browser compatibility. The tools discussed include open-source tools like Selenium and JMeter, and commercial tools like UFT and Ranorex. The researchers discuss the challenges associated with selecting the

appropriate testing tool, noting that no single tool can meet all automation testing needs. The process of selecting a tool depends on factors such as the project environment, application requirements, and the testers' knowledge. The researchers also review literature that highlights the importance of testing modern applications, as the complexity of modern software systems increases, leading to a greater need for automation in testing to reduce costs and increase productivity. The paper indicates that the most commonly used tools include Selenium WebDriver, UFT, and Ranorex, based on the project's budget and the environment in which it operates.

According to (Sabev & Katalina Grigorova, 2017), the solution in the current era to deliver high-quality software is to design test cases that are automatically created, executed, and maintained. The researchers concluded that there is no perfect tool for automating GUI testing. The top tool scored only 29 points out of 40. Each of these tools has its own drawbacks and advantages, indicating that one should take all factors into consideration when choosing GUI automation tools.

The most important features that researchers focused on in the process of comparing GUI automation tools are the recording/replay feature, ease to use GUI, image recognition, support for more than one programming language and more than one operating system, reporting tools, maintainability of test case codes, and technical support for the tool. (Vahid Garousi, et al., 2017)Experimentally evaluated Visual GUI Testing (VGT) tools to determine the most suitable tool for corporate needs. VGT is a technique that uses image recognition to interact with the system under test and validate its behavior. This study, inspired by a real industrial need at a large Turkish software and systems company operating in defense and IT sectors, aimed to assess well-known VGT tools (Sikuli and JAutomate) to help the company choose the best tool for specific testing projects. The VGT project was planned and executed in an industrial context, evaluating two wellknown tools (Sikuli and JAutomate). The study analyzed the quality of "record and playback" features, the flexibility and repeatability of test execution, and the effort required for test development. The study findings:

- Playback Issues: Both tools encountered similar playback issues, such as the inability to locate small images.
- Test Execution Repeatability: JAutomate showed better repeatability for one of the tested systems, while the repeatability was comparable for the other system.
- Test Development Effort: There was a strong correlation between the development effort and the number of steps in test suites for both tools. However, the effort could be reduced by reusing test code.

The study provided tangible benefits to the company's engineers and test managers by enhancing their understanding of VGT and identifying challenges and solutions in using these tools.

(Abhishek V, Naveen A, Vidya G, & Jasmine K.S, 2014) Presented a comparative study of software testing tools used to test the graphical user interface, where they compared open source software testing tools and proprietary software testing tools due to the importance of choosing the appropriate testing tool in the success of the project.

The researchers found that open source software is the most reliable and has the best performance in testing the graphical user interface and verifying whether the application is functionally correct. In most cases, testing the graphical user interface includes executing a set of tasks and comparing their results with the expected outputs with the ability to replicate the same set of tasks multiple times with different data entered. Researchers have found that open source testing tools are fewer in comparison to proprietary software testing tools. Open source testing tools are faster in generating the test report when compared to proprietary testing tools.

(Ateşoğulları & Alok, AUTOMATION TESTING TOOLS: A COMPARATIVE VIEW, 2020) Discuss automation testing tools used in software development, reviewing 21 different tools such as Selenium, Appium, and Unified Functional Testing (UFT), and comparing them based on 20 different criteria. The study aims to help developers choose the most suitable tool based on their specific needs, achieving

efficiency and quality in software testing processes. The paper emphasizes the importance of selecting the appropriate framework for automation tests, reviewing different types of frameworks such as data-driven and keyword-driven frameworks. The study highlights that Selenium is one of the most popular and widely used tools, especially for testing web applications, as it is opensource and supports multiple programming languages like Java, Python, and Ruby. The paper also discusses other tools like Appium, which is used for testing mobile applications and supports both Android and iOS. UFT, on the other hand, is a paid tool that offers extensive support for web and desktop applications, providing features like test recording and detailed reporting. The study indicates that some tools excel in certain aspects, such as ease of use, technical support, and compatibility with various systems and applications. It also highlights the importance of considering cost and project budget when choosing an automation testing tool, given the availability of open-source tools like Selenium and paid tools like UFT.

The paper concludes that there is no single tool that fits all needs; the choice depends on the type of application, budget, and testing requirements. The study recommends continuing to develop and improve current automation tools and exploring new technologies like artificial intelligence and machine learning to enhance the efficiency and quality of software testing.

According to (Monika Sharma & Rigzin Angmo, 2014), there are several considerations that must be taken into account when choosing the best web application testing tool. The most important of these considerations are ease of integration, compatibility with application design, and comparison between cost and performance. The researcher discussed the top ten web automation testing tools: Selenium, HP –QTP, FitNesse, Watir, testComplete, LoadRunner,TestNg, TOSCA, SilkTest, WinRunner.

The first tool is Selenium that a portable and an open source automated testing suite that comprises of 4 tools: selenium IDE, RC, Webdrive, grid. The second tool is HP –QTP that provides regression and functional testing automation for major software environment and applications. The third tool is FitNesse: that a tool for enhancing collaboration in software development. Fitnesse enables customers, testers, and programmers to discover what their software should do, and to automatically compare that to what it actually does. The fourth tool is Watir: It is acronym for web application testing in ruby. Watir is an open source family that uses ruby libraries for automation web browsers. The fifth tool is TestComplete: that an automated testing tool which allows testers to create, manage and run tests for any Windows, Web or Rich Client software. The sixth tool is LoadRunner: that examines system performance and behavior. HP LoadRunner works by using the virtual users. It also simulates thousands of concurrent users to put the application through various real life user loads and analyses the results in detail to discover the particular behavior. The seventh tool is TestNg: that refers to "Testing, the Next Generation",

TestNg introduced some new functionality that makes it more powerful and easier for testers. The eighth tool is TOSCA: that is a test suite includes integrated test management, design, execution and data generation toolset for functional and regression tests. The ninth tool is SilkTest: that has the ability to test across multiple platforms, browsers and technologies. Silk Test offers test planning, validation, management. The last tool is WinRunner: that is an automation Functional GUI testing tool. This tool allowed User to capture, verified and replayed UI interaction as test script.

3. CRITERIA FOR SELECTING GUI TEST AUTOMATION
TOOLS IN THE STUDY

The researchers focused on selecting tools specifically designed for testing Graphical User Interfaces (GUI) for web applications. The testing tools were chosen based on several key criteria:

 i.Support for GUI testing.

 ii.Support for web application testing. iii.Availability of tool support.

Therefore, the criteria of GUI testing support and web application testing support were not included in the comparison between tools, as any tool that did not support these two criteria was excluded from the study. The researchers selected twenty-five tools out of fifty-one and excluded twenty-six tools for the aforementioned reasons. Below is a brief description of the selected tools to compare their advantages and disadvantages. The tools are listed in alphabetical order, not based on their popularity.

[1] Applitools

Applitools is designed to ensure that applications look and function correctly across all devices and browsers. It leverages AI-powered visual recognition to detect visual bugs and inconsistencies. Applitools integrates seamlessly with existing test automation frameworks like Selenium, Cypress, and WebDriverIO. It also provides a dashboard for detailed test analysis and collaboration. Applitools is highly regarded for its ability to enhance cross-browser testing and improve test maintenance.

[2] Cucumber

Cucumber is an open-source tool used for Behavior-Driven Development (BDD), allowing users to write test scenarios in plain language. These scenarios, facilitate better communication between technical and non-technical stakeholders. Cucumber supports multiple programming languages and integrates with various automation tools. It promotes collaboration and ensures that software behaves as expected by all parties. Cucumber is popular for its userfriendly approach to creating and maintaining test cases.

[3] Cypress

Cypress is a fast, reliable, and easy-to-setup testing framework specifically designed for web applications. It offers a powerful dashboard for debugging and running tests. Unlike other tools, Cypress operates directly in the browser, providing a more consistent testing environment. It supports JavaScript and integrates with CI/CD pipelines, making it ideal for developers who need quick feedback on their code changes. Cypress is known for its ease of use and modern web testing capabilities.

[4] FitNesse

FitNesse is a web-based testing framework that allows users to create and manage tests using a wiki interface. It supports both technical and non-technical users by providing a simple way to write tests in plain language. FitNesse integrates with various development tools and can execute tests written in different programming languages. It is particularly useful for fostering collaboration between developers, testers, and business stakeholders. FitNesse is valued for its flexibility and ease of integration.

[5] Functionize

Functionize is a cloud-based test automation platform that uses AI and machine learning to create and maintain tests. It enables users to automate complex workflows without requiring extensive coding knowledge. Functionize supports web, and mobile testing, providing detailed analytics and reporting features.

Its self-healing capabilities ensure that tests remain robust even when the application under test changes. Functionize is praised for its innovative approach to reducing test maintenance and improving test reliability.

[6] Imagium.io

Imagium.io is a visual testing tool that focuses on automating the visual validation of web applications. It uses advanced image recognition algorithms to detect visual discrepancies and ensures that the UI appears correctly across different devices and browsers. Imagium.io provides seamless integration with CI/CD pipelines and popular test automation frameworks. It is highly appreciated for its accuracy in visual comparisons and its ability to reduce manual visual inspection efforts.

[7] Katalon Studio

Katalon Studio is an all-in-one test automation solution for web, mobile, API, and desktop applications. It offers an easy-to-use interface with built-in keywords and drag-anddrop capabilities, making it accessible for users with varying levels of technical expertise. Katalon Studio supports multiple scripting languages and integrates with CI/CD tools. It provides comprehensive reporting and analytics features, enhancing test management and execution. Katalon Studio is known for its affordability and extensive functionality. **[8] Playwright**

Playwright is an open-source test automation tool developed by Microsoft, designed for end-to-end testing of web applications. It supports multiple browsers, including Chromium, Firefox, and WebKit, and can run tests in parallel for faster execution. Playwright allows for robust and reliable testing with features like auto-wait, network interception, and cross-browser testing. It is known for its powerful capabilities in handling modern web app complexities. **[9] Puppeteer**

Puppeteer is a Node.js library developed by Google that provides a high-level API to control Chrome or Chromium browsers. It is primarily used for automating web testing, scraping, and generating PDFs from web pages. Puppeteer offers powerful browser automation capabilities, including interaction with web elements, taking screenshots, and performance analysis.

[10] Protractor

Protractor is an end-to-end test framework for Angular and AngularJS applications. It runs tests against the application in a real browser, interacting with it as a user would. Protractor integrates seamlessly with Angular, offering features like automatic waiting for elements and synchronization with the Angular framework. It is designed to work with Jasmine and other test frameworks, providing a complete testing solution for Angular applications.

[11] Ranorex

Ranorex is a commercial test automation tool for desktop, web, and mobile applications. It provides a user-friendly interface with a codeless option for creating tests, as well as support for scripting in C# and VB.NET. Ranorex offers robust object recognition and integrates with various CI/CD tools. It is known for its comprehensive reporting and debugging features, making it suitable for both novice and experienced testers.

[12] Rapise

Rapise is a test automation tool designed for testing web, mobile, and desktop applications. It supports a wide range of technologies and platforms, offering both codeless and script-based test creation. Rapise provides powerful object recognition and reusability of test scripts. It integrates with popular test management and CI/CD tools, making it a versatile choice for comprehensive test automation.

[13] RFT (Rational Functional Tester)

Rational Functional Tester (RFT) is an IBM tool for automated functional and regression testing. It supports a
variety of applications, including web, Java, and .NET. RFT provides both script-based and scriptless test creation options, leveraging technologies like data-driven testing and keyword-driven testing. It integrates with IBM's Rational Test Workbench and other IBM products, offering a robust testing solution for enterprise environments.

[14] Sahi

Sahi is an open-source tool designed for automated testing of web applications. It supports both browser-based and server-side testing, providing a simple scripting language for writing tests. Sahi is known for its ability to handle complex web applications with dynamic content. It offers features like in-browser controls, record and playback, and extensive reporting, making it a powerful tool for web application testing. **[15] Selenium**

Selenium is a widely-used open-source framework for web application testing. It supports multiple browsers and platforms, allowing tests to be written in various programming languages such as Java, C#, and Python. Selenium provides tools like Selenium WebDriver for browser automation, Selenium Grid for parallel testing, and Selenium IDE for record and playback. It is known for its flexibility and extensive community support.

[16]Sikuli

Sikuli is a visual testing tool that uses image recognition to automate GUI interactions. It allows users to create scripts by capturing screenshots of UI elements and defining actions based on these images. Sikuli is useful for automating tasks that involve visual elements, regardless of the underlying technology. It is particularly effective for applications where traditional object recognition fails.

[17] SilkTest

SilkTest is a test automation tool developed by Micro Focus for functional and regression testing of desktop, web, and mobile applications. It provides robust test creation and management features, supporting multiple scripting languages. SilkTest is known for its high performance and ability to handle complex testing scenarios. It integrates with other Micro Focus products, offering a comprehensive testing solution.
[18] Squish

Squish is a cross-platform test automation tool designed for GUI applications. It supports various technologies, including Qt, Java, Windows, and web applications. Squish provides powerful object recognition and scripting capabilities, allowing tests to be written in multiple languages such as JavaScript, Python, and Perl. It is known for its flexibility and extensive support for different GUI toolkits. **[19] Telerik Test Studio**

Telerik Test Studio is a comprehensive test automation tool for web, desktop, and mobile applications. It offers a userfriendly interface with codeless test creation and supports multiple browsers and platforms. Test Studio provides robust reporting and integrates well with CI/CD pipelines, making it suitable for end-to-end testing scenarios.

[20] Test Architect

Test Architect is an automation tool that supports testing for web, mobile, and desktop applications. It provides a visual interface for creating and managing tests and integrates with various development and testing tools. Test Architect focuses on improving test maintenance and scalability, making it suitable for large-scale testing projects.

[21] TestCafe

TestCafe is an open-source, end-to-end testing framework for web applications. It does not require browser plugins and supports modern JavaScript features like ES6/7. TestCafe is known for its ease of use, fast setup, and integration with CI/CD systems, making it a popular choice for front-end testing.

[22] TestComplete

TestComplete is a comprehensive test automation tool that supports web, mobile, and desktop applications. It offers script and scriptless test creation options, robust object recognition, and integration with various CI/CD tools. TestComplete is known for its ease of use and extensive features, making it suitable for a wide range of testing needs.

[23] Tricentis Tosca

Tricentis Tosca is an enterprise-level test automation tool that supports continuous testing for web, mobile, and desktop applications. It uses model-based test automation to create and maintain test cases efficiently. Tosca integrates with various development and testing tools, providing robust reporting and analytics features.

[24] UFT (Unified Functional Testing)

Unified Functional Testing (UFT), formerly known as HP QuickTest Professional (QTP), is an automated functional and regression testing tool. It supports a wide range of applications and offers a graphical interface for creating test scripts. UFT integrates well with various development environments and provides powerful features for comprehensive test automation.

[25] Watir

Watir (Web Application Testing in Ruby) is an open-source tool for automating web browsers. It is written in Ruby and uses Ruby's expressive syntax to write test scripts. Watir supports multiple

browsers, multiple programming languages and integrates well with other testing tools, making it a flexible and powerful option for web application testing.

4. CRITERIA USED FOR EVALUATION AND COMPARISON

In the context of research aimed at analyzing and evaluating (GUI) automation testing tools, fifteen criteria were selected after reviewing previous studies to ensure comprehensive coverage of various aspects affecting the efficiency and effectiveness of these tools. These criteria not only help determine the tool's compatibility with project requirements but also ensure the achievement of the research objectives outlined in the introduction.

To compare the tools used in GUI automation testing, it is essential to evaluate them based on the same criteria. The researchers have chosen fifteen criteria to assess and compare these tools. Below is a brief description of why the researchers selected these criteria. **[1] Artificial Intelligence Support**

Object recognition using artificial intelligence is a crucial feature for improving the accuracy and speed of object detection in the user interface. It helps reduce manual intervention and enhances test automation, aligning with the goal of improving process efficiency. AI can assist in automatically generating test cases, identifying anomalies, and improving testing efficiency through intelligent automation.

[2] Desktop Application Support:

This criterion evaluates the tool's ability to test desktop applications. Effective desktop application support ensures that the tool can handle a wide range of software testing scenarios beyond web and mobile applications. This is important for organizations that develop and maintain desktop software. **[3] Ease of Integration:**

Ease of integration refers to the extent to which a tool can work seamlessly with other tools and systems, which helps reduce complexity and increases process efficiency, which is in line with the goal of reducing cost and time, thus enhancing overall productivity.

[4] Free:

Free tools offer a significant economic advantage, especially for projects with limited budgets, which is in line with the goal of reducing testing costs. Free tools may come with some limitations, but they often provide basic functionality to get started with GUI testing, so it is important to consider the potential need for additional resources to overcome any limitations. **[5] Mobile Application Support:**

With the proliferation of mobile applications, the support of these tools for mobile application testing is essential to ensure comprehensive testing coverage. Mobile support refers to the tool's ability to test mobile applications on different platforms such as iOS and Android. Effective mobile support is crucial to ensure a seamless user experience across mobile devices.

[6] Multi Operating System Support:

Multiple operating system support enhances the tool's ability to test software across different environments, which contributes to improving the overall coverage of tests. This evaluates the tool's compatibility with different operating systems, such as Windows, macOS, and Linux. **[7] Multi Programming Language Support:**

Multiple programming language support ensures that the tool can integrate with projects that rely on various programming languages, expanding its scope of use. This flexibility allows teams to use their preferred languages and integrate the tool into various development environments.

[8] Multiple Browsers Support:

This criterion evaluates the tool's ability to test applications across different web browsers to ensure that the application works correctly on all popular browsers, enhancing the ultimate user experience. **[9] Open source:**

An open source tool is one whose source code is freely available for modification and distribution. Open source tools offer greater flexibility, transparency, and customization options. Open source tools often have strong community support and frequent updates. However, users may need a higher level of technical expertise to get the most out of these tools and integrate them effectively into their testing environments.

[10] Record and Reply:

The Record and Replay feature enables testers to record their interactions with the GUI and automatically replay them. This feature simplifies the test creation process and helps in reproducing and diagnosing issues quickly by comparing the recorded actions with the expected results. **[11] Screen Shots support:**

Screen Shots support helps document test cases and identify errors easily, which contributes to improving the quality of tests and their reports. **[12] Virtual users Support:**

Virtual user support enables stress and performance testing to be performed by simulating a large number of users, which helps identify potential performance issues. **[13] User Experience:**

User experience (UX) measures how easy it is for users to interact with a testing tool. This includes user interface design, clarity of instructions, ease of navigation, and overall user satisfaction. A tool with a superior user experience can significantly reduce the learning curve, increase productivity, and improve the efficiency of the testing process by making it more accessible to both technical and non-technical users. **[14] Technical support:**

Technical support refers to the assistance provided by the tool's developers or vendors to help users solve problems and improve their use of the tool. Good technical support can include a variety of resources such as documentation, forums, email support, live chat, and phone support. Reliable technical support is critical to quickly address issues, implement best practices, and ensure smooth running of testing processes, which reduces downtime and increases the effectiveness of the tool. **[15] Release Date:**

The release date of a GUI testing tool indicates how long it has been on the market. This information helps users understand how mature and sophisticated the tool is. Older tools may have more extensive documentation and community support, while newer tools may include the latest technologies and methodologies.

ID	Name of the Tool	AI	Desk. supp	Integr.	free	Mob. supp	Multi. OS	Multi. Prog.	Multi. Browser	Op. Source	Rec & Replay	Sc. Shot	Virtual User	UE	Tech Supp.	Release Date
1	Applitools	T	T	T	F	T	T	T	T	F	T	T	T	T	T	2013
2	Cucumber	T	T	T	F	T	T	T	T	F	T	T	T	T	T	2008
3	Cypress	F	F	T	F	F	T	F	T	T	F	T	T	T	T	2014
4	FitNesse	F	T	T	T	T	T	T	T	T	F	F	T	F	T	2003
5	Functionize	T	F	T	F	T	T	T	T	F	T	T	T	T	T	2016
6	imagium.io	T	F	T	F	F	T	T	T	F	F	T	T	T	T	2019
7	Katalon Studio	T	T	T	F	T	T	T	T	F	T	F	T	T	T	2015
8	Playwright	F	T	T	T	T	T	T	T	T	T	T	F	T	T	2020
9	Puppeteer	F	F	T	T	F	T	F	T	T	T	T	T	T	T	2017
10	Protractor	F	F	T	T	F	T	T	T	T	T	F	T	T	T	2013
11	Ranorex	T	T	T	F	T	F	T	T	F	T	T	T	T	T	2007
12	Rapise	F	T	T	F	T	T	T	T	F	T	T	T	T	T	2011
13	RFT	F	T	T	F	T	T	T	T	F	T	T	T	T	T	2002
14	Sahi	F	F	T	F	F	T	T	T	T	T	F	F	T	T	2005
15	Selenium	F	F	T	T	F	T	T	T	T	T	T	T	F	T	2004
16	Sikuli	F	T	T	T	F	T	T	T	T	F	F	T	T	T	2009
17	SilkTest	T	T	T	F	T	T	T	T	F	T	F	T	T	T	1993
18	Squish	F	T	T	F	T	T	T	T	F	T	F	F	T	T	2003
19	Telerik Test Studio	T	T	T	F	T	T	T	T	F	T	T	T	T	T	2009
20	Test Architect	F	T	T	F	T	T	T	T	T	T	F	F	T	T	2006
21	TestCafe	F	F	T	T	F	T	F	T	T	F	F	T	T	T	2016
22	Test Complete	T	T	T	F	T	F	T	T	F	T	T	T	T	T	1999
23	Tricentis Tosca	T	T	T	F	T	T	T	T	F	T	T	T	T	T	2008
24	UFT	F	T	T	F	T	T	T	T	F	T	F	T	T	T	2002
25	Watir	F	F	T	T	F	T	T	T	T	F	F	T	T	T	2004

Table No (1) Comparison of GUI Automation Testing Tools

5. ANALYSIS OF A COMPARISON TABLE OF GUI

AUTOMATION TESTING TOOLS

This study evaluates 25 GUI automation testing tools based on 15 criteria to provide a comprehensive comparison, aimed at assisting researchers, developers, and testers in selecting the most suitable tools for their software projects. The study's main objective is to highlight key strengths and weaknesses in each tool to align the selection process with project requirements, ultimately enhancing software quality, reducing costs, and increasing efficiency.

[1] AI Capabilities

AI-driven features are present in 10 tools, with varying levels of sophistication. Applitools and Functionize are noted for their advanced AI capabilities, which enhance their ability to provide intelligent insights and recommendations. While others like Katalon Studio and Test Complete offer average AI features, others like Selenium and Cypress rely on traditional automation without advanced AI features, which may be sufficient for projects with less emphasis on AI-driven testing.

[2] Desktop Application Testing Support

Support for desktop application testing is provided by 16 tools. This feature remains important for organizations that develop both web and desktop applications, ensuring comprehensive test coverage.

[3] Integration Capabilities

All tools mentioned support various integrations with CI/CD tools and test management tools, but some tools like FitNesse and Sikuli may have limited integrations and rely heavily on custom integrations.

[4] Free Version Availability

Free versions are available in 8 of the tools, making them accessible to smaller teams or those with limited budgets. This is especially important for startups and individual developers who need to manage costs while still leveraging powerful testing tools.

[5] Mobile Application Testing Support

16 tools offer support for mobile application testing, reflecting the growing importance of mobile platforms. This feature is crucial for teams focused on delivering seamless experiences across both web and mobile applications.

[6] Multi Operating System Support

The majority of tools (23 out of 25) support multiple operating systems, ensuring broad applicability across different development environments. This is particularly important for teams working in diverse ecosystems or targeting multiple platforms.

[7] Multi Programming Language Support

Supporting multiple programming languages is vital for flexibility in test automation. 22 tools provide this support, allowing teams to write scripts in their preferred languages. However, tools like Cypress, Puppeteer, and TestCafe do not offer this flexibility, which could be a limiting factor for teams using diverse tech stacks.

[8] Multiple Browsers Support

All 25 tools support multiple browsers, which is essential for cross-browser testing. This feature ensures that web applications function correctly across different browsers and platforms, a critical requirement in today's diverse web environment.

[9] Open Source

Open-source tools are favored for their flexibility and effectiveness. 11 of the evaluated tools, including Selenium, Cypress, and TestCafe, are open source. These tools are popular in the developer community due to their extensibility and the ability to tailor them to specific needs, although they may require more effort in setup and maintenance compared to commercial tools.

[10] Record & Replay

Record & Replay functionality is present in 19 of the tools, providing ease in test case creation and execution. This feature is particularly useful for teams that need to quickly generate test cases without writing extensive scripts.

[11] Screen Shots Support

Only 14 tools support screenshot capturing, which can be a limitation for tools that focus on detailed visual validation. Screenshot support is valuable for verifying UI elements and detecting visual regressions, making it a key feature in tools like Applitools, which specialize in visual testing.

[12] Virtual User Support

The ability to simulate real-user scenarios is supported by 21 of the tools. This feature is particularly valuable for load testing and performance validation, where simulating multiple users interacting with the application simultaneously is critical. The widespread support for virtual users reflects its significance in modern testing strategies, especially for web applications that need to handle high traffic volumes.

[13] User Experience (UE)

User experience is another key criterion, with 22 tools offering a positive user experience. Tools that excel in this area often feature intuitive interfaces, clear documentation, and user-friendly workflows.

[14] Technical Support

A critical aspect for many organizations, technical support is provided by 24 out of the 25 tools evaluated. The only exception is FitNesse, which offers partial support. The widespread availability of technical support underscores its importance in ensuring smooth adoption and effective use of these tools, particularly in complex testing environments where troubleshooting and guidance are crucial.

[15] Release Date

The release dates of the tools span from 1993 (SilkTest) to 2020 (Playwright). A notable trend is the surge in the development and release of GUI automation tools post2000, particularly between 2008 and 2017, reflecting a significant rise in demand for advanced testing solutions during this period. This trend corresponds to the growing complexity of web applications and the need for robust testing tools to ensure software reliability. The emergence of newer tools like Playwright highlights advancements in features such as cross-browser testing and developerfriendly environments, which were less emphasized in older tools.

The analysis reveals that while many tools offer robust support for technical requirements like multi-browser and multi-OS support, features such as AI capabilities, open-source availability, and mobile testing support are less universally available. The distinction between tools is becoming increasingly pronounced as newer tools incorporate AI and other advanced features to enhance testing accuracy and efficiency. The evolution of GUI automation tools reflects broader industry trends, such as the growing importance of AI, cross-platform compatibility, and user-friendly interfaces. For researchers, developers, and testers, this comparative study underscores the importance of aligning tool selection with project-specific needs. While Selenium remains a top choice for its comprehensive features and community support, newer tools like Applitools and Playwright offer specialized capabilities that can significantly enhance testing outcomes in specific contexts. The choice of tool should be guided by the unique requirements of the software project, whether it be AI-driven visual testing, cross-browser compatibility, or support for mobile platforms.
This expanded analysis emphasizes that a one-size-fits-all approach is insufficient; instead, a strategic selection based on the detailed features and strengths of each tool is crucial for achieving software project goals efficiently and effectively.

6. CONCLUSION

In this study, we conducted a comprehensive comparison of twenty-five GUI automation testing //tools for web applications, evaluating each tool based on fifteen distinct criteria. The primary objective was to aid researchers, developers, and testers in selecting the most suitable tool for their specific software testing needs. The comparative analysis revealed that each tool possesses unique advantages and potential drawbacks, underlining the importance of aligning tool selection with project requirements.
Our findings highlight that no single tool excels in all aspects; rather, the choice of an automation testing tool should be guided by factors such as the nature of the application, budget constraints, required technical support, and the specific testing needs. For instance, tools like Selenium are highly regarded for their open-source nature and cross-browser testing capabilities, while tools like Applitools stand out for their AI-powered visual recognition features.
The study underscores the critical role of automation in enhancing testing efficiency, coverage, and reliability, especially in the context of modern web applications. Automation tools significantly reduce the time and effort required for testing, thereby lowering costs and minimizing the risk of human error. However, the decision to adopt a particular tool should consider the overall testing strategy, including the need for integration with existing frameworks, ease of use, and the level of technical expertise available.
Future research could expand on this study by exploring the performance of these tools in different real-world scenarios and by examining emerging trends in test automation, such as the integration of AI and machine learning. As the landscape of web application development continues to evolve, the continuous evaluation and adaptation of testing tools will remain crucial for maintaining software quality and meeting user expectations.
Ultimately, this comparative study serves as a valuable resource for making informed decisions about GUI automation testing tools, facilitating the development of robust and user-friendly web applications.

References

[1] At Thooriqoh, H., Tiara Nur Annisa, & Umi Laili Yuhana. (2021). SELENIUM FRAMEWORK FOR WEB AUTOMATION TESTING: A SYSTEMATIC LITERATURE REVIEW. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 65-76.

[2] Abhishek V, Naveen A, Vidya G, & Jasmine K.S. (2014). Comparative study on Software testing tools used for GUI testing. *International Journal of Engineering Research & Technology (IJERT)*, 70-74.

[3] Ateşoğulları, D., & A. M. (2020). AUTOMATION TESTING TOOLS: A COMPARATIVE VIEW. *International Journal of Information and Computer Security*, 62-76.

[4] Ateşoğulları, D., & Mishra, A. (2020). AUTOMATION TESTING TOOLS: A COMPARATIVE VIEW. *International Journal of Information and Computer Security*, 63-76.

[5] Kumari, S., Madhulika Mittal, Simranjeet Kour, & Vikash Sawan. (2022). Comparison of Selenium, Quick Test Professional, and Test Complete as Automated Testing Tools. *International Journal of All Research Education and Scientific Methods (IJARESM)*, 1490-1495.

[6] Pelivani, E., & Betim Cico. (2021). A comparative study of automation testing tools for web applications. *10th MEDITERRANEAN CONFERENCE ON EMBEDDED COMPUTING* . BUDVA, MONTENEGRO.

[7] Sabev, P., & Katalina Grigorova. (2017). A Comparative Study of GUI Automated Tools for Software Testing. *The Third International Conference on Advances and Trends in Software Engineering*, 715.

[8] SAMLI, R., & Zeynep ORMAN. (2023). A Comprehensive Overview of Web-Based Automated Testing Tools. *İleri Mühendislik Çalışmaları Ve Teknolojileri Dergisi 4, no. 1*, 13-28.

[9] sen, r., savitri trangirala, sindhu D V, & saba farheen N S. (2023). A COMPARATIVE STUDY OF AUTOMATION TESTING TOOLS FOR MOBILE, WEB AND DESKTOP APPLICATIONS. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 332-337.

[10] Singh Gadwal, A., & Prasad, L. (2020). Comparative review of the literature of automated testing tools. *International Journal of Information and Computer Security*, 63-76.

[11] V, A., Naveen A, Vidya G, & Dr.Jasmine K.S. (2018). Comparative study on Software testing tools used for GUI testing. *International Journal of Engineering Research & Technology (IJERT)*, 70 - 74.

[12] Vahid Garousi, Wasif Afzal, Adem Çağlar, İhsan Berk Işık, Berker Baydan, Seçkin Çaylak, . . . Kadir Herkiloğlu. (2017). Comparing Automated Visual GUI Testing Tools: An Industrial Case. *proceedings of the 8th ACM SIGSOFT International Workshop on Automated Software Testing*, (pp. 21-28). Paderborn, Germany.

[13] Yadu, P., & Narain, D. (2022). SOFTWARE TESTING TOOLS AND FRAMEWORKS WITH VISUAL GUI TECHNIQUES IN INDUSTRIAL PRACTICE. *International Journal of Mechanical Engineering*, 1100-1112.

[14] Applitools. *Applitools: Automated Visual Testing*. Available at: https://applitools.com [Accessed 5 August 2024, 9:15 AM].

[15]   Cucumber. *Cucumber: BDD for All*. Available at: https://cucumber.io [Accessed 8 August 2024, 10:30 AM].

[16]   Cypress. *Cypress: Fast, Easy and Reliable Testing for Anything That Runs in a Browser*. Available at: https://www.cypress.io [Accessed 3 August 2024, 11:00 AM].

[17]   FitNesse. *FitNesse: A Fully Integrated Standalone Acceptance Testing Framework*. Available at: http://www.fitnesse.org [Accessed 12 August 2024, 10:45 AM].

[18]   Functionize. *Functionize: The Future of Test Automation*. Available at: https://www.functionize.com [Accessed 6 August 2024, 11:30 AM].

[19]   imagium.io. *Imagium: AI-Powered Visual Testing Platform*. Available at: https://imagium.io [Accessed 9 August 2024, 8:50 AM].

[20]   Katalon Studio *Katalon Studio: Simplify API, Web, Mobile Automation Tests*. Available at: https://www.katalon.com [Accessed 2 August 2024, 9:45 AM].

[21]   Playwright. *Playwright: Reliable End-to-End Testing for Modern Web Apps*. Available at: https://playwright.dev [Accessed 14 August 2024, 10:00 AM].

[22]   Puppeteer. *Puppeteer: Headless Chrome Node API*. Available at: https://pptr.dev [Accessed 4 August 2024, 11:15 AM].

[23]   Protractor. *Protractor: E2E Test Framework for Angular and AngularJS Applications*. Available at: https://www.protractortest.org [Accessed 1 August 2024, 9:30 AM].

[24]   Ranorex. *Ranorex: Test Automation for Any Desktop, Web, and Mobile Application*. Available at: https://www.ranorex.com [Accessed 7 August 2024, 9:00 AM].

[25]   Rapise. *Rapise: Next Generation Test Automation*. Available at: https://www.inflectra.com/Rapise/ [Accessed 13 August 2024, 11:45 AM].

[26]   RFT. *Rational Functional Tester (RFT): IBM Test Automation Tool*. Available at: https://www.ibm.com/products/rational-functionaltester [Accessed 3 August 2024, 8:15 AM].

[27]   Sahi. *Sahi Pro: Automated Web Testing*. Available at: https://sahipro.com [Accessed 11 August 2024, 8:30 AM].

[28]   Selenium. *Selenium: Web Browser Automation*. Available at: https://www.selenium.dev [Accessed 10 August 2024, 10:15 AM].

[29]   Sikuli. *SikuliX: Automate Anything You See on Screen*. Available at: http://www.sikulix.com [Accessed 2 August 2024, 11:00 AM].

[30]   SilkTest. *Silk Test: Automated Functional and Regression Testing*. Available at: https://www.microfocus.com/en-us/products/silktest/overview [Accessed 5 August 2024, 8:30 AM].

[31]   Squish. *Squish: Automated GUI Testing*. Available at: https://www.froglogic.com/squish/ [Accessed 14 August 2024, 9:45 AM].

[32]   Telerik Test Studio. *Test Studio: Automated Testing Tool for Web and Desktop*. Available at: https://www.telerik.com/teststudio [Accessed 15 August 2024, 9:00 AM].

[33]   Test Architect. *TestArchitect: Enterprise Test Automation*. Available at: https://www.testarchitect.com [Accessed 9 August 2024, 11:30 AM].

[34]   TestCafe. *TestCafe: Node.js End-to-End Testing Framework*. Available at: https://testcafe.io [Accessed 6 August 2024, 10:00 AM].

[35]   Test Complete. *TestComplete: Automated UI Testing for All*. Available at: https://smartbear.com/product/testcomplete/overview / [Accessed 12 August 2024, 9:15 AM].

[36]   Tricentis Tosca. *Tricentis Tosca: Continuous Testing for DevOps*. Available at: https://www.tricentis.com/products/tosca-testautomation [Accessed 8 August 2024, 9:30 AM].

[37]   UFT. *UFT One: Automated Functional Testing*. Available at: https://www.microfocus.com/enus/products/uft-one/overview [Accessed 7 August 2024, 10:45 AM].

[38]   Watir. *Watir: Web Application Testing in Ruby*. Available at: http://watir.com [Accessed 4 August 2024, 9:15 AM].