A Data-Driven Comparative Study of Web Application Automation Testing Frameworks via Analytical Hierarchy Process

Reza Wahyu Daknantya1, a) , Gita Indah Marthasari1, b) , Briansyah Setyo Wiyono1, c)

Author Affiliations

1Informatic, Faculty of Engineering, Muhammadiyah Malang University   
[Jalan Raya Tlogomas No 8](https://www.bing.com/ck/a?!&&p=ec7265a856919ac0cb4fb34e5d1f57601ce51c1a5963682b94d75fc89ed987e7JmltdHM9MTc0NzY5OTIwMA&ptn=3&ver=2&hsh=4&fclid=358c1a4e-848f-6052-1207-0bb3808f6e92&u=a1L21hcHM_Jm1lcGk9MTI3fn5Vbmtub3dufkFkZHJlc3NfTGluayZ0eT0xOCZxPVVuaXZlcnNpdGFzJTIwTXVoYW1tYWRpeWFoJTIwTWFsYW5nJnNzPXlwaWQuWU43OTk5eDExNTE2NzAyMjMwMDk2MjAxOTAwJnBwb2lzPS03LjkyMTQ0OTE4NDQxNzcyNV8xMTIuNTk2MDg0NTk0NzI2NTZfVW5pdmVyc2l0YXMlMjBNdWhhbW1hZGl5YWglMjBNYWxhbmdfWU43OTk5eDExNTE2NzAyMjMwMDk2MjAxOTAwfiZjcD0tNy45MjE0NDl-MTEyLjU5NjA4NSZ2PTImc1Y9MSZGT1JNPU1QU1JQTA&ntb=1), Malang – Jawa Timur, 65145.

Author Emails  
a) yudhareza3@webmail.umm.ac.id  
b) gita@umm.ac.id

b) brian@umm.ac.id

**Abstract.** The rapid proliferation of web applications necessitates the adoption of automated testing procedures to improve efficiency and reduce the risk of human error. However, the wide variety of available automation testing frameworks poses challenges in selecting the most suitable tool. This study aims to identify the optimal automation testing framework for web-based applications using the Analytical Hierarchy Process (AHP) method. Three widely used frameworks—Serenity BDD, Cypress, and Robot Framework are evaluated against seven criteria: covered test cases, time complexity, execution speed, element inspection, platform compatibility, script language, and parallel execution. Data collection is conducted through direct testing on a web application and expert questionnaires within the Software Quality Assurance domain. The results indicate that Cypress is the most optimal framework, achieving superior performance in multiple aspects, including an average execution speed of 6.073 seconds, a total computation time of 455.8 seconds, and comprehensive test coverage of 75 test cases. This research provides data-driven guidelines to assist practitioners in making objective, measurable, and efficient decisions when selecting automated testing frameworks.

# introduction

The objective of website development process testing is to ensure the effectiveness and quality of the system. Automation testing has gained widespread popularity due to its ability to expedite the testing process relative to manual testing methods and to mitigate human error [1], [2]. The automation testing framework itself is an architectural design intended to evaluate application features based on business logic and systematic code structure [2]. However, the abundance of available frameworks can complicate the decision-making process for testers [3]. Consequently, researchers have identified the necessity to conduct a comparative analysis to ascertain the most efficacious framework for automated testing on websites. The present study focuses on a comparison of three popular frameworks in the Indonesia Software Quality Assurance (ISQA) community: Serenity, Cypress, and Robot Framework. Senior software quality assurance professionals frequently recommend them for their proficiency in website testing. The functionality testing method was selected for this research due to its extensive scope, enabling a comprehensive evaluation of the performance of each framework [4].

To support the decision-making process in choosing the optimal framework, this research employs the Analytical Hierarchy Process (AHP) method. This approach facilitates the development of a hierarchical structure predicated on a number of relevant criteria, which are then compared to determine priority weights [4], [5]. This approach facilitates the structuring of the decision hierarchy and the weighting of the evaluation criteria of the automation testing framework [4]. In the context of website automated testing, the framework with the highest weight will be regarded as the optimal framework. The evaluation framework encompasses a range of criteria, including covered test cases, time complexity, execution speed, element inspection, platform compatibility, script language, and parallel execution [6].

Elis Pelivani and Betim Cico's (2021) study, "A comparison of automation testing tools on websites," indicates that Katalon Studio exhibits superiority in specific parameters when compared to other tools [2]. Concurrently, a technical analysis of Selenium and Cypress frameworks for automation testing, as conducted by Fatini Mobaraya and Shahid Ali in 2019, determined that Selenium exhibits superior performance compared to Cypress [7]. However, both studies did not employ quantitative comparison methods in their decision-making process. Consequently, this research proposes a more systematic and credible approach using the Analytical Hierarchy Process (AHP) method, expected to make a significant contribution to software testers in choosing the appropriate automation testing framework.

The objective of this research is to provide a credible quantitative evaluation of three automation testing frameworks by systematically adopting the Analytical Hierarchy Process (AHP) method. The findings are anticipated to offer objective guidance for software testers in selecting a suitable framework for testing web-based applications.

# literature review

## Automation Testing

As stated in Min et al. 2020, automation testing, also known as automated testing, is a software testing method that utilizes specialized software to manage the execution of tests and subsequently compares the observed test results with the predicted or anticipated results. Automated testing can be executed with the assistance of tools to optimize the scope of testing [4]. Automated testing can minimize the occurrence of human errors that usually occur during manual testing and is more practical for running test cases repeatedly and can be re-run at any time based on the needs required [6]. Automated testing is also the best choice for several scenarios such as Regression Testing, Functionality Testing, Repeated Execution, and Performance Testing [6]. In regression testing, testing automation helps ensure that code changes do not break existing features. In performance testing, automated tools are used to test the speed, stability, and reliability of the system under various conditions. Meanwhile, in functional testing, automation testing ensures that application features work according to predefined specifications.

## Automation Testing Framework

Automation testing frameworks are a combination of tools and practices designed to simplify the software testing process through organized and reusable test structures [5]. The framework has been demonstrated to support test efficiency and effectiveness through script standardization and integration with test management systems and CI/CD tools [8]. Automation testing frameworks provide a standardized method for modifying, adding, and removing scripts and functions. The judicious application of automation testing frameworks has been demonstrated to enhance software quality and expedite development cycles. The implementation of an automation testing framework enables the testing team to allocate a greater proportion of their time to the validation and analysis of test results. This shift in focus enables the team to reduce the time spent on activities such as rewriting test scripts and manually testing applications. Therefore, the selection of a suitable framework is of paramount importance to ensure the success of test automation initiatives.

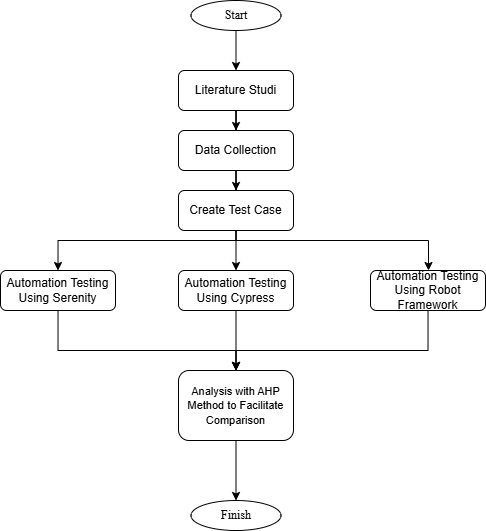
The author elected to ascertain the most efficacious option among Serenity, Cypress, and Robot Framework. Serenity BDD is a sophisticated, open-source automated testing framework designed to facilitate efficient software test development and reporting. Furthermore, Serenity utilizes test results to generate illustrations, narrate reports, and elucidate the functionality of the application [9]. Cypress is a contemporary automated testing framework that is characterized by its speed and efficiency. It was developed with a specific focus on the domain of web application development testing. Cypress offers a suite of sophisticated tools that facilitate the seamless automation of functionality testing, integration, and unit testing of web applications [10]. Robot Framework is an open-source test automation framework that is characterized by its ease of use. The objective of its design is to support programming language-based testing approaches, and it employs easy-to-read tabular syntax. Robot Framework facilitates the automation of functional testing, user acceptance testing, and continuity testing [11].

## Analytical Hierarchy Process

The Analytical Hierarchy Process (AHP) is a systematic approach utilized in multi-criteria-based decision-making. AHP is a valuable tool in situations where the weights of multiple criteria are not known with certainty. By employing assessment input in the form of a pairwise comparison matrix [12], AHP facilitates the determination of the relative importance of these criteria. This approach offers a systematic framework for decomposing intricate problems into more manageable and quantifiable components, thereby facilitating a more efficient and objective decision-making process. The Analytical Hierarchy Process (AHP) is a methodological framework that facilitates the comparison of elements through the utilization of an Eigenvalue approach. This approach is instrumental in determining the priority weight of each criterion, thereby enabling a systematic and objective assessment of relative importance. The assessment utilizes a numerical scale ranging from 1 to 9, which represents the relative importance of one element to another [12]. This scale facilitates the process of calibrating the assessment to both quantitative and qualitative aspects, rendering the method highly flexible in various evaluation contexts. The primary benefit of the Analytical Hierarchy Process (AHP) is its capacity to address issues involving numerous complex criteria. AHP combines qualitative and quantitative data, providing a mechanism to measure the consistency of decisions through the calculation of consistency ratios. Consequently, the validity of the obtained weights can be mathematically substantiated. This method has been widely adopted in various fields, including management, engineering, and business, due to its capacity to systematically address uncertainty and subjectivity in judgment [12].

# Research Method

The stages of the research methodology applied in this study follow a systematic flow as shown in Figure 1.



**Figure 1.** Research Method

## Identification of The Tested System

In this research, the researcher will test the Medeasy.ph website as a case study. At this stage, the author will identify the website thoroughly by ensuring the flow of the features to be tested and understanding well the features that will be tested by the researcher. The features that will be assessed by researchers include the Medical Calculator feature, which encompasses sub-features such as BMI Calculator, Cardio Risk, Diabetes Risk, Medicine Reminder, Menstruation Calendar, and Pregnancy Due Date [13].

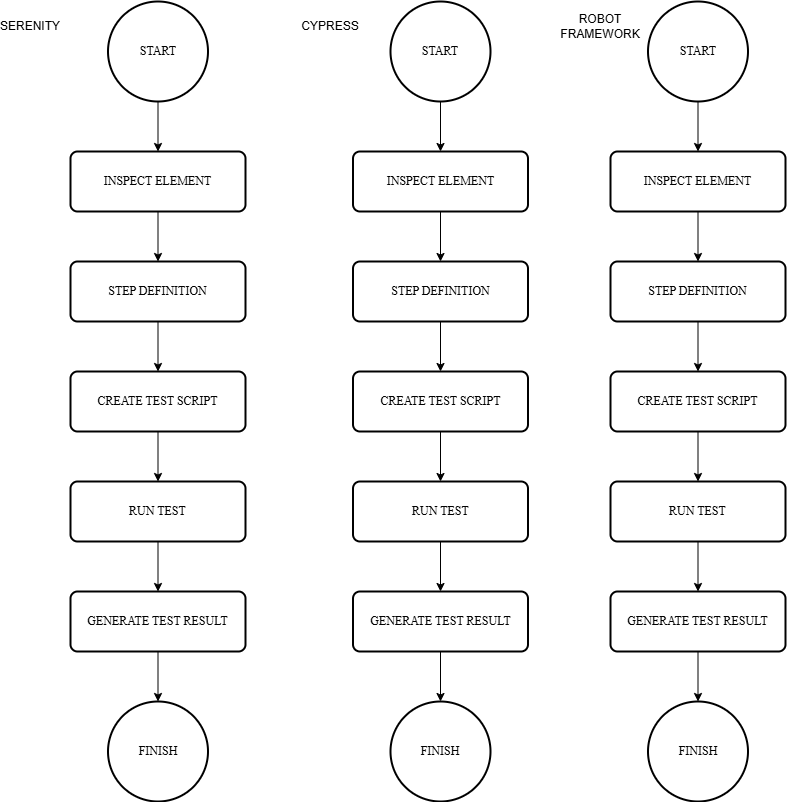
## Test Case Compilation

Test cases constitute a critical component of the software testing cycle, meticulously designed to ensure that the application or system functions in accordance with the specified requirements. Test cases are defined as scenarios or steps that are meticulously designed to evaluate a specific functionality or aspect of the software [14]. Each test case encompasses expected inputs, the steps to be taken, and the anticipated results.

In this study, a total of 75 test cases were meticulously compiled based on the functions in the features under evaluation. Each test case encompasses the identification of inputs, the delineation of test steps, the establishment of output expectations, and the classification of test categories as positive or negative. To access the complete set of test cases, please refer to the following [*link*](https://docs.google.com/spreadsheets/d/18M3q9zXzyLHJc1lM6xD2-7WM6uMMOjzLO2BIEMqeYSM/edit?usp=sharing)*.*

## Implementation of Automation Testing Framework

This research employs an experimental approach to compare the performance of three popular automation testing frameworks: Serenity BDD, Cypress, and Robot Framework. The selection of these tools was predicated on their extensive utilization by software testing practitioners and their endorsement by various Software Quality Assurance (SQA) communities. The testing process is executed directly on web-based applications by following standard steps in script development and automated test execution. These steps include identifying interface elements and analysing results in the form of test reports. Here are the steps of each framework in Figure 2.



**Figure 2.** Automated Testing Flow on Serenity, Cypress and Robot Framework

Serenity BDD is a Behaviour Driven Development (BDD) framework that facilitates scenario-based testing using Gherkin, a natural language formulation. The testing process with Serenity initiates with the inspect element process, wherein the elements of a website, such as id, name, class, and XPath, are identified for utilization in scripts. The subsequent step in the process is the creation of step definitions, which are achieved through the compilation of test logic mappings using annotations such as @Given, @When, @Then, and @And. This approach facilitates high readability, even for non-technical parties. Test scripts are constructed through the integration of object elements, scenario steps, anticipated outcomes, and assertion tests for result evaluation. Tests are executed via terminal commands, and the results are presented in the form of comprehensive HTML reports, complete with visualizations and summaries of automated test results [10].

Meanwhile, Cypress is recognized as a contemporary JavaScript-based framework that has been specifically designed for the purpose of web application testing. The process commences with the identification of elements, employing the same method, and is subsequently followed by the creation of test scripts that utilize cy objects as the foundation of syntax for each action or validation. Cypress scripts are characterized by their incorporation of interface interaction, expected result validation, and a range of assertions that are executed asynchronously. Tests can be executed through a variety of commands, including npx cypress open for visual interfaces or npx cypress run for headless execution. Upon completion of the execution, Cypress furnishes an interactive test runner that presents logs, snapshots of the test process, and the results of each line of script in detail, thereby enabling efficient analysis and debugging [7].

The Robot Framework is a keyword-driven framework that is flexible and supports various types of testing, including web UI, mobile, and web services. The framework employs a modular approach, with a file structure divided into several sections, including variables, test cases, and keywords. The initial identification of interface elements to be tested is conducted through the utilization of common attributes such as id and XPath. In the process of developing test scripts, users can employ data parameterization to facilitate data-driven testing and leverage openly available libraries. The execution of tests is facilitated by employing the robot -d output/name\_file.robot command, which generates an execution report that encompasses a summary, an activity log of each step, and the status of each scenario. This report is instrumental in identifying failures and assessing the efficacy of the testing process [11].

## Comparative Analysis Using Analytical Hierarchy Process

The Analytical Hierarchy Process (AHP) method conceptualizes the decision-making process as a hierarchical structure comprising three primary components: objectives, criteria, and alternatives [15]. Criteria are evaluative factors that influence the decision, while alternatives are the options considered in achieving the goal. In the context of this research, the objective is to ascertain the most optimal automation testing framework. The evaluation encompasses seven testing parameters, and the contending alternatives include Serenity BDD, Cypress, and Robot Framework. The evaluation framework encompasses a range of criteria, including covered test cases, time complexity, execution speed, element inspection, platform compatibility, script language, and parallel execution [6].

The initial stage in applying AHP is the weighting process for each criterion. This weighting method is intended to reflect the relative importance of each criterion in supporting the achievement of goals. The determination of weights for qualitative criteria is achieved through the implementation of interviews with subject matter experts and relevant stakeholders [16]. The assessment is administered using an intensity of importance scale ranging from 1 to 9, as outlined in the standard AHP fundamental scale developed by Saaty [16]. The subsequent stage involves the formulation of a pairwise comparison matrix, wherein each criterion is meticulously evaluated in pairs to ascertain its relative importance [15]. The matrix is symmetrical, with a diagonal value of 1, indicating that each element has the same value when compared to itself.

The construction of a priority value matrix or a criteria value matrix is achieved through the normalization of the pairwise comparison matrix. Normalization is achieved through the division of each element in the matrix column by the total number of values in that column. Subsequently, the average of each row is calculated to obtain the final weight of each criterion [15]. This process can be expressed in equation (1).

The value of a represents the element in the i row and j column in the pairwise comparison matrix. The value is the total of all elements in the jth column, while is the result of normalizing the element by dividing it by the value. Equation (2) which states the calculation process of normalization and priority weights.

Normalized values in the i row and j column are used to calculate the final weight. Sum of all normalized values in the i row is divided by the total criteria , resulting in the final weight or priority for the i criterion .

Subsequently, the summation of each row in the comparison matrix is conducted to ascertain the aggregate relative influence of a specific criterion on all other criteria, as delineated in formula (3).

is the total number of elements in the i-th row, is the comparison value between the i criterion and the j criterion, and is the total number of criteria. These three terms are used to systematically explain the relationships and calculations in the analysis of specific criteria.

To ensure the validity of the weighting results, consistency testing was conducted through the calculation of the Consistency Index (CI) and the Consistency Ratio (CR). The critical index (CI) is determined by the difference between the maximum eigenvalue (λmax) and the number of criteria (n), as delineated in equation 4.

The Consistency Index (CI) is a metric that quantifies the consistency of the priority weight λ\_(max))in an assessment. λ\_(max) is derived from the pairwise comparison matrix and represents the priority weight that is calculated. Conversely, n denotes the quantity of criteria employed to ascertain the degree of consistency in decision evaluation. Subsequent to acquiring the CI value, the subsequent step is to ascertain the value of the consistency ratio (CR). The calculation of the CR value can be performed using equation 5.

CR is the abbreviation for Consistency Ratio, CI is the abbreviation for Consistency Index, and RI is the abbreviation for Random Index. The utilization of these three approaches is predicated on their capacity to assess the consistency inherent in the decision-making process, particularly within the framework of the AHP (Analytic Hierarchy Process) method. This methodological approach is employed to ensure the attainment of precise and dependable outcomes during the evaluation or assessment of a given alternative. Additionally, CR is derived by dividing CI value is determined by the Random Index (RI), whose value depends on the number of criteria as listed in Table 1. CR value is less than or equal to. The value of 0.1 indicates that the comparison matrix is considered consistent and can be used to support subsequent analyses.

Table 1. The Random Index Value

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Criterion | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Random Index | 0 | 0 | 0,58 | 0,9 | 1,12 | 1,24 | 1,32 | 1,41 | 1,45 | 1,49 | 1,51 | 1,48 | 1,56 | 1,57 | 1,59 |

The culminating step in the AHP method is the decision-making process, which is informed by the priority weights that have been calculated for each alternative. The framework that has been assigned the highest weight value is thereby designated as the optimal alternative in view of fulfilling all the stipulated criteria. The AHP method's primary strength lies in its capacity to systematically, accurately, and flexibly organize the evaluation process, encompassing both quantitative and qualitative data. Therefore, AHP can be considered a reliable approach in supporting multi-criteria-based decision making in the context of selecting an automated software testing framework.

# research result

The evaluation of Serenity BDD, Cypress, and Robot Framework was carried out using 75 functional test cases covering five sub-features of the Medeasy.ph Medical Calculator: BMI Calculator, Cardio Risk, Diabetes Risk, Menstruation Calendar, and Pregnancy Due Date. These features were selected due to their diverse data input requirements, multi-step calculation logic, and the need for accurate UI interaction making them representative of typical web application testing challenges.

## Empirical Analysis of Automation Testing Result

This study conducted automated testing on the **medeasy.ph** website using three automation frameworks Serenity BDD, Cypress, and Robot Framework to evaluate test case coverage, total computation time, and execution speed. The results of the testing are summarized in Table 2.

**Table 2.** Automation Testing Result Data

|  |  |  |  |
| --- | --- | --- | --- |
| **Framework Name** | ***Covered Test Case*** | ***Time Complexity*** | ***Execution Speed*** |
| Serenity | 74 | 1525,1 | 20,334 |
| Cypress | 75 | 455,8 | 6,073 |
| Robot Framework | 74 | 1279,6 | 17,061 |

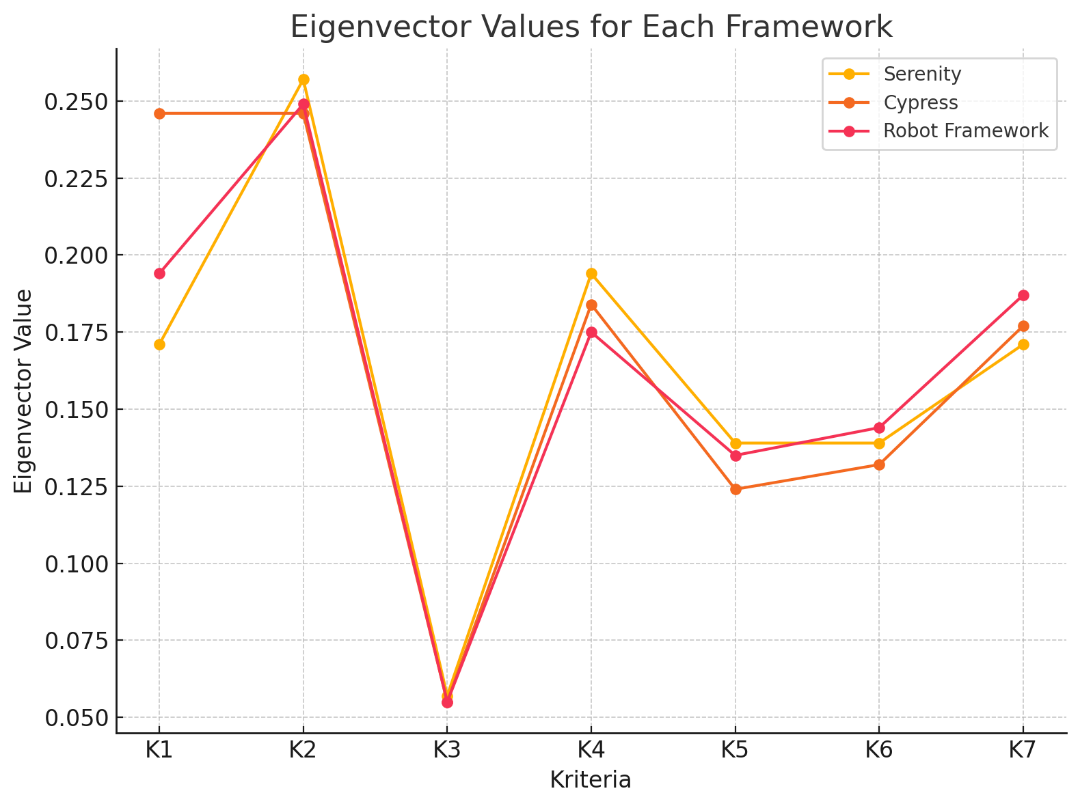
The execution performance data revealed that Cypress achieved the fastest average execution time of 6.073 seconds per test case, outperforming Robot Framework at 17.061 seconds and Serenity BDD at 20.334 seconds. Similarly, Cypress demonstrated the lowest total computation time of 455.8 seconds for all test cases, compared with 1279.6 seconds for Robot Framework and 1525.1 seconds for Serenity BDD. The performance advantage of Cypress can be attributed to its architecture, which executes directly in the browser environment and leverages asynchronous event handling without requiring external drivers such as Selenium WebDriver.

In terms of test coverage, Cypress successfully executed 100% of the 75 test cases, while Serenity BDD and Robot Framework exhibited occasional failures. These failures were primarily due to dynamic DOM changes and synchronization delays, which required additional wait-time configuration in Serenity BDD and more complex keyword customization in Robot Framework. This observation aligns with prior studies indicating that frameworks with native DOM access and automatic wait handling generally outperform driver-based frameworks in UI stability.

Element inspection capabilities, essential for locating and interacting with UI elements, were also strongest in Cypress due to its real-time reactivity and snapshotting features. Serenity BDD provided comprehensive element handling through Selenium but required more extensive scripting to achieve similar responsiveness. Robot Framework, while capable, struggled with dynamic element updates a limitation also noted by Pelivani and Cico (2021) in multi-page web applications[2].

## Multi-Criteria Decision Analysis Using AHP

The Analytical Hierarchy Process was applied to combine quantitative metrics with expert evaluations from three experienced SQA practitioners[17]. The pairwise comparison matrix identified execution speed as the most critical criterion (weight = 0.238), followed by covered test cases (0.197) and time complexity (0.164). Element inspection, platform compatibility, scripting language, and parallel execution received relatively lower weights but remained relevant to overall framework effectiveness. The consistency ratio (CR) for all comparisons was below the 0.1 threshold, ensuring reliability in expert judgments[17].



**Figure 3.** Eigenvector Values for Each Framework

The eigenvector values derived from the Analytical Hierarchy Process (AHP) analysis identify the highest-performing framework for each evaluation criterion. In **K1 (Covered Test Case)**, Cypress achieved the highest score 0.246, indicating slightly superior test coverage compared to its counterparts. For **K2 (Total Execution Time)**, Serenity recorded the highest value 0.257, reflecting greater priority assigned to this criterion in its overall performance profile. In **K3 (Execution Speed)**, all three frameworks demonstrated similar values 0.055, with negligible differences, indicating that no single framework distinctly dominated this parameter.

In **K4 (Element Inspection)**, Serenity attained the top score 0.194, showcasing its strength in accurately identifying and interacting with interface elements. For **K5 (Platform Compatibility)**, Serenity and Robot Framework shared similarly high values 0.139 and 0.135, respectively, suggesting comparable adaptability across multiple platforms. In **K6 (Script Language)**, Robot Framework led with a score of approximately 0.144, highlighting its flexibility in supporting various scripting languages. Finally, in **K7 (Parallel Execution)**, Robot Framework achieved the highest score 0.187, demonstrating superior capability in executing tests concurrently.

The distribution of eigenvector values provides a clear indication of each framework’s relative strengths across the evaluation criteria. When these priority weights are aggregated in accordance with the AHP methodology, they yield the overall performance scores for each framework. This integration of weighted criteria ultimately determines the final ranking, identifying Cypress as the most optimal framework among the three evaluated alternatives.

**Table 3**. The Final Ranking Result

|  |  |
| --- | --- |
| **Framework Name** | **Normalization Amount** |
| Cypress | 1,141 |
| Robot Framework | 1,085 |
| Serenity | 1,045 |

Weighted scoring results ranked Cypress highest with a composite score of 0.412, ahead of Serenity BDD (0.356) and Robot Framework (0.232). This finding reinforces the view that multi-criteria approach such as AHP provide a more comprehensive assessment than single-metric evaluations, particularly in balancing speed, coverage, and maintainability.

## Comparative Insights and Practical Implications

The results corroborate previous work founded Cypress superior for UI test automation in web environments due to its low-latency execution and minimal flakiness under repeated runs. However, the present study advances the discussion by incorporating AHP to account for multiple performance and usability factors simultaneously, thereby offering a more replicable and transparent selection process[8].

Interestingly, earlier studies reported that Selenium-based frameworks could surpass JavaScript-based tools in scalability and integration flexibility. While Serenity BDD, built on Selenium, indeed provided richer reporting and better integration with Behavior-Driven Development (BDD) pipelines, these advantages were insufficient to offset its slower execution speed and greater configuration overhead in the tested context. This highlights the contextual nature of framework selection, where the “best” choice depends on the weight assigned to specific criteria such as speed versus integration features[18].

Robot Framework, although ranking lowest in this study, remains relevant in environments requiring cross-technology test automation, such as hybrid web–desktop or IoT testing. Its keyword-driven approach and broad library support make it appealing for non-programmer testers, but its performance limitations in handling dynamic web content restrict its competitiveness for time-critical functional testing[19].

From a practical standpoint, the results suggest that teams prioritizing rapid feedback cycles and minimal configuration overhead should consider Cypress as their primary tool for web UI functional testing. Conversely, projects emphasizing integration with BDD workflows or requiring multi-environment compatibility might still favor Serenity BDD despite its slower execution. For organizations seeking a generalized testing platform across multiple domains, Robot Framework’s flexibility could outweigh its slower runtime in certain use cases.

By integrating empirical measurements with expert judgments via AHP, this study addresses a critical gap in prior literature, which often relied solely on empirical benchmarks or qualitative reviews. The methodological approach presented here offers both rigor and adaptability, enabling its application to other software quality assurance decision-making contexts beyond web automation testing[2], [20].

# conclusion

This study presented a data-driven comparative evaluation of three widely adopted automation testing frameworks Serenity BDD, Cypress, and Robot Framework applied to the Medeasy.ph Medical Calculator web application. Using a hybrid methodology that combined functionality testing results with the Analytical Hierarchy Process (AHP), the evaluation incorporated seven performance and usability criteria: covered test cases, time complexity, execution speed, element inspection, platform compatibility, scripting language, and parallel execution.

Empirical results demonstrated that Cypress outperformed the other frameworks across most criteria, achieving the fastest execution speed (6.073 seconds per test case), the shortest total computation time (455.8 seconds), and 100% coverage of 75 functional test cases. The AHP-based analysis confirmed Cypress as the optimal choice with the highest composite score, followed by Serenity BDD and Robot Framework. These findings underscore Cypress’s suitability for projects that prioritize rapid execution, stability, and minimal configuration overhead in web UI functional testing.

From a practical perspective, the methodology applied in this study offers a replicable framework selection process that balances objective performance data with subjective expert judgment. The results are particularly relevant for software quality assurance teams operating under time-sensitive conditions, where tool efficiency directly impacts delivery cycles. From a theoretical standpoint, this research contributes to the body of knowledge by demonstrating the value of multi-criteria decision-making techniques such as AHP in software engineering tool selection, thereby addressing limitations in prior work that relied solely on empirical or qualitative comparisons.

Future research could extend this evaluation to a broader range of automation frameworks, incorporate additional criteria such as maintainability and learning curve, or apply the methodology to other domains such as mobile application testing and continuous integration pipelines. By adapting the AHP-based approach to evolving technological contexts, practitioners and researchers can make more informed, transparent, and context-sensitive decisions in selecting optimal testing frameworks.

# References

[1] F. A. K. P. G. Sutapa, S. Kusumawardani, and A. E. Permanasari, “Automated Test Suite for Regression Testing Based on Serenity Framework: A Case Study,” *Proceeding - 2019 International Conference of Artificial Intelligence and Information Technology, ICAIIT 2019*, pp. 138–144, Mar. 2019, doi: 10.1109/ICAIIT.2019.8834609.

[2] E. Pelivani and B. Cico, “A comparative study of automation testing tools for web applications,” *2021 10th Mediterranean Conference on Embedded Computing, MECO 2021*, Jun. 2021, doi: 10.1109/MECO52532.2021.9460242.

[3] C. Merina, “Analisis perbandingan kinerja test automation framework untuk functional testing pada aplikasi berbasis android dengan metode the distance to the ideal alternative,” Nov. 2017, Accessed: Jun. 04, 2025. [Online]. Available: https://repository.uinjkt.ac.id/dspace/handle/123456789/53151

[4] J. L. Min, A. Istiqomah, and A. Rahmani, “Evaluasi Penggunaan Manual Dan Automated Software Testing Pada Pelaksanaan End-To-End Testing,” *JTT (Jurnal Teknol. Ter., vol. 6, no. 1, p. 18, 2020, doi: 10.31884/jtt. v6i1. 256*, 2020.

[5] D. S. N, S. D. S, D. Vijayasree, N. S. Roopa, and A. Arun, “A Review on the Process of Automated Software Testing,” Sep. 2022, Accessed: Jun. 04, 2025. [Online]. Available: https://arxiv.org/pdf/2209.03069

[6] B. Renaldy, . B., D. A. Agusti, and A. D. P.-U. N. Mandiri, “PEMILIHAN FRAMEWORK UNTUK PEMBUATAN APLIKASI MOBILE CROSS-PLATFORM DENGAN MENGGUNAKAN METODE ANALYTICAL HIERARCHY PROCESS (AHP),” *Indonesian Journal of Networking and Security (IJNS)*, vol. 11, no. 1, Jun. 2022, doi: 10.55181/IJNS.V11I1.1764.

[7] F. Mobaraya, S. Ali, and others, “Technical Analysis of Selenium and Cypress as functional automation framework for modern web application testing,” in *9th international conference on computer science*, 2019.

[8] G. A. Prasetyo and N. Setiani, “Perbandingan Kinerja Framework Automation UI Testing Menggunakan The Distance to The Ideal Alternative,” *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 10, no. 1, pp. 224–237, Jan. 2025, doi: 10.29100/JIPI.V10I1.5760.

[9] U. Sa’adah *et al.*, “AUTOMATIC TESTING FRAMEWORK BASED ON SERENITY AND JENKINS AUTOMATED BUILD,” *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 19, no. 2, pp. 102–110, Jul. 2021, doi: 10.12962/J24068535.V19I2.A1017.

[10] W. Mwaura, *End-to-End Web Testing with Cypress: Explore techniques for automated frontend web testing with Cypress and JavaScript*. Packt Publishing Ltd, 2021.

[11] “Writing Your First Code | ROBOT FRAMEWORK.” Accessed: Jun. 04, 2025. [Online]. Available: https://docs.robotframework.org/docs/getting\_started/how\_to\_write\_rf

[12] D. N. Alfiana, V. R. S. Nastiti, and C. S. K. Aditya, “Sistem Pendukung Keputusan Pemilihan Lokasi Tanah Strategis di Kota Mataram Menggunakan Metode AHP-TOPSIS,” *Jurnal Repositor*, vol. 5, no. 1, pp. 591–602, 2023.

[13] “Medeasy.” Accessed: Jun. 04, 2025. [Online]. Available: https://medeasy.ph/

[14] N. Doorn, T. E. J. Vos, and B. Marín, “Towards understanding students’ sensemaking of test case design,” *Data Knowl Eng*, vol. 146, p. 102199, Jul. 2023, doi: 10.1016/J.DATAK.2023.102199.

[15] A. R. Harahap, N. H. M. Simbolon, R. A. Agata, and S. Sunarsih, “Metode Fuzzy AHP (Analytical Hierarchy Process) untuk Pemilihan Metode Pembelajaran Demi Menunjang Pembelajaran Matematika,” *Jurnal Sains dan Edukasi Sains*, vol. 5, no. 1, pp. 9–17, Feb. 2022, doi: 10.24246/JUSES.V5I1P9-17.

[16] H. Irawan, I. Ismiyati, and B. Pudjianto, “Penentuan Skala Prioritas Penanganan Jalan Kabupaten di Kabupaten Kudus Dengan Metode Analytical Hierarchy Process,” *TEKNIK*, vol. 37, no. 2, pp. 72–77, Dec. 2016, doi: 10.14710/TEKNIK.V37I2.8411.

[17] T. L. Saaty, “The analytic hierarchy process (AHP),” *J Oper Res Soc*, vol. 41, no. 11, pp. 1073–1076, 1980.

[18] F. Mobaraya and S. Ali, “TECHNICAL ANALYSIS OF SELENIUM AND CYPRESS AS FUNCTIONAL AUTOMATION FRAMEWORK FOR MODERN WEB APPLICATION TESTING,” pp. 27–46, 2019, doi: 10.5121/csit.2019.91803.

[19] H. U. Rahman, A. Alzayed, M. I. Mohmand, A. M. Albarrak, and S. N. Qasem, “Application Maintenance Offshoring Using HCI Based Framework and Simple Multi Attribute Rating Technique (SMART),” *IEEE Access*, vol. 11, pp. 107068–107084, 2023, doi: 10.1109/ACCESS.2023.3320941.

[20] S. Carpitella, V. Kratochvíl, and M. Pištěk, “Multi-criteria decision making beyond consistency: An alternative to AHP for real-world industrial problems,” *Comput Ind Eng*, vol. 198, p. 110661, Dec. 2024, doi: 10.1016/J.CIE.2024.110661.

# 