MULTI-CRITERIA DECISION-MAKING IN THE EVALUATION OF SOFTWARE TESTING METHODS

Ana Bašić[1]*,
[0009-0009-4137-3296]

Dejan Viduka[2]
[0000-0001-9147-8103]

[1]Information Technology School - ITS, Belgrade, Serbia

[2]Alfa BK University, Faculty of Mathematics and Computer Sciences, Belgrade, Serbia

Abstract:

Development of software requires reliable and effective testing methods to ensure its functionality and quality. This research was conducted with the aim of identifying the most suitable testing method for different software development scenarios, considering key evaluation criteria. The paper analyzes and ranks four software testing methods – Black-Box Testing, White-Box Testing, Automated Testing and Agile Testing – using the PIPRECIA-S method. The analysis was conducted based on five key criteria: efficiency, testing coverage, ease of implementation, cost of implementation and reliability of results. The results showed that Automated Testing is the most effective method with the highest overall score, while Black-Box Testing ranks last due to limited reliability and effectiveness. Agile and White-Box testing proved to be moderately balanced methods, with different advantages depending on the specific requirements of the project. This paper provides an insight into the advantages and limitations of each method, as well as guidelines for their selection in real projects, contributing to a better understanding of the optimization of the software testing process.

Keywords:

Software Testing, Testing Methods, PIPRECIA-S Method, Quality Assurance, Decision-making.

INTRODUCTION

Developing high-quality software is a challenge in the modern software industry. Adequate software testing ensures the reliability, functionality and efficiency of the end product [1, 2]. However, the large number of software testing methods makes it difficult to properly choose a method that would meet specific project demands. The aim of this research is to provide an analysis and ranking of several software testing methods: Black-Box Testing, White-Box Testing, Automated Testing and Agile Testing. The methods are compared according to five key criteria: efficiency, test coverage, ease of application, cost of application and reliability. These criteria have been carefully selected because they reflect the most important aspects of quality and practical application of test methods in different contexts.

Correspondence:
Ana Bašić

e-mail:
ana.basic@its.edu.rs

CC BY

DOI: 10.15308/Sinteza-2025-150-157

To achieve an objective evaluation and enable decision-making, the PIPRECIA-S (Pivot Pairwise Relative Criteria Importance Assessment - Simplified) method was applied. This method is used because of its flexibility, ability to integrate the subjective judgments of experts, and precision in identifying the relative importance of criteria [3]. Through the application of the PIPRECIA-S method, it is possible to precisely determine the importance of each criterion, which enables reliable ranking and selection of the optimal testing method for a specific project.

2. SOFTWARE TESTING METHODS

Software testing is a key process in quality assurance, which enables error detection, performance evaluation and confirmation of the functionality of software systems [4]. Four methods were analysed in this research: Black-Box Testing, White-Box Testing, automated testing and agile testing. The selection of these test methods is based on their versatility and ability to cover different aspects of software development and validation.

Black-Box Testing is a method that focuses on the functionality of a software system, without insight into its internal structure or code [4, 5]. Using this method, testers evaluate software based on input data and expected outputs, according to system specifications. The goal is to verify that the system responds correctly to inputs and generates expected outputs according to specifications [5, 6]. The tester plays the role of the end user and focuses on testing the input and output data, the behaviour of the application and its functional requirements. The advantage of this method is that it does not require technical knowledge about the code, it allows validation of the end user experience and can be used at any stage of software development. On the other hand, this method does not provide insight into the internal structure of the code and may miss implementation-level errors [6].

White-Box Testing involves an in-depth analysis of the internal logic and code structure of a software system [7]. A tester has insight into system implementation and uses technical skills to analyse paths, control flows, loops, and functions. This testing enables validation of not only functionality, but also performance optimization and identification of hidden bugs, improves quality of code through detailed analysis, and helps identify security vulnerabilities [5, 6]. Applying this method requires deep technical knowledge and access to code, and the software testing process can be very time-consuming.

Automated testing uses tools and scripts to execute predefined test cases, eliminating manual steps and speeding up the testing process [8]. It is used for regression testing, performance testing, scalability and load testing. Automated tools enable test repeatability and reduce the risk of human error, making them ideal for testing large and complex systems. Using this method significantly reduces the time and costs of testing, and enables quick detection of errors. This testing method requires a high initial investment in tools and employee training. Maintaining scripts can be complex if the software changes frequently.

Agile testing is an integral part of iterative and incremental development according to the Agile methodology [9]. Testing is done continuously throughout the development process, often in collaboration with developers and other team members. The focus is on quick feedback and adaptation to changes in user requirements. The advantages of this method are that it is flexible and easily adaptable to changes in user requirements [9]. Using this method improves communication between testers, developers, and users, which reduces the risk of major errors late in software development. Impeding factors for effective application of the method may be unclear priorities in software development or unclear user requirements.

3. METHODOLOGY

Two separate groups of researchers participated in the research. The first group consisted of experts from the fields of software development, information technology, software testing, management and economics. They were in charge of defining the evaluation criteria. Five key criteria were identified: efficiency, test coverage, ease of application, cost of implementation and reliability of results. Based on the technical requirements of the testing methods, the specifics of the projects and the available resources, the priorities of the criteria are defined in order to enable an objective assessment and the selection of the optimal testing method for different cases. Another group of researchers applied defined criteria for evaluating selected testing methods. Ratings are assigned based on practical experiences, user feedback and analysis of specific project needs. The PIPRECIA-S method was used to accurately determine the importance of each criterion. This method allows reliable ranking and selection of the optimal test method for a specific project.

3.1. PIPRECIA-S

The PIPRECIA-S method was used to determine the weighting coefficients of the criteria. Application of the PIPRECIA-S method implies that the importance of each criterion is compared with the importance of the first criterion. The main advantage of this method is its simplicity. Also, this method is very easily applied in group decision-making processes. The procedure for determining weight coefficients consists of five steps [3, 10].

The first step involves defining the evaluation criteria C_j, j=1,…,n where n is the number of criteria taken into 83 account when solving the problem. To determine the criteria, one can use the literature or the opinion of experts 84 from the relevant field of research. The second step represents the determination of the relative importance of 85 criteria s_j. It is necessary to determine the criterion (C_1) used as a basis for comparison. After that, starting from the 86 second criterion, each criterion C_j is assigned the relative importance of criterion s_j based on Equation 1. Each 87 criterion C_j is compared with the reference criterion C_1.

$$s_j = \begin{cases} >1 \, , C_j > C_1 \\ =1 \, , C_j = C_1 \\ <1 \, , C_j < C_1 \end{cases}$$

Equation 1. Relative importance value s_j based on C_j and C_1

If criterion C_j is more important than criterion C_1, it is assigned a value s_j that is greater than 1 and vice versa. If the criteria are equally important, then both criteria have an importance value of 1. The values s_j belong to the interval [0.6, 1.4]. The value of $ss1$ is always 1 and represents an estimate of the importance of the reference criterion C_1.

In the third step, the value of the coefficient k_j is calculated based on Equation 2.

$$k_j = \begin{cases} 1, j{=}1 \\ 2 - s_j, j{>}1 \end{cases}$$

Equation 2. Calculation of the coefficient k_j

In the fourth step, the value of the coefficient q_j is calculated based on equation (3).

$$q_j = \begin{cases} 1, j{=}1 \\ \dfrac{q_{j-1}}{k_j}, j{>}1 \end{cases}$$

Equation 3. Calculation of the coefficient q_j

The last step involves the calculation of the relative weight of criteria w_j, where $0 \le w_j \le 1$ and $\sum_{k=1}^{n} w_k = 1$ based on Equation 4.

$$w_j = \frac{q_j}{\sum_{k=1}^{n} q_k}$$

Equation 4. Calculation of the relative weight of criteria w_j

The analysis was conducted on the basis of five key criteria: efficiency, test coverage, ease of application, cost of application and reliability of results.

Efficiency refers to the speed and ability of a method to detect and isolate software errors. Methods that enable rapid identification of critical issues are essential for projects with tight deadlines. The indicators used to evaluate this criterion are: the average error detection time and the number of detected errors in a certain period of time.

Test coverage measures how much system functionality or lines of code a method covers. High-coverage methods reduce the chance of undetected errors and enable detailed system analysis. The indicators used to evaluate this criterion are: percentage of functionality covered and percentage of code covered by the tests.

The criterion related to the simplicity of application evaluates how simple the method is to implement and use in real conditions. This includes the time required to train the team, the complexity of the tool, and the resources required for testing. The indicators used to evaluate this criterion are: the time required to learn the method and the required level of technical knowledge.

The cost of implementation includes the cost of tools, staff training, time spent on testing, as well as additional resources required for implementation. The indicators used to evaluate this criterion are: total costs per test cycle and maintenance costs of methods or tools.

Reliability of results refers to the accuracy and consistency of detected errors, as well as the method's ability to minimize false positive or negative results. The indicators used to evaluate this criterion are: the percentage of false positive and false negative results and the consistency of the results in different conditions.

3.2. RANKING SCALE

For each of the selected criteria, a ranking will be used based on the defined scale presented in Table 1. This scale follows the principles of the PIPRECIA-S method. The PIPRECIA-S method uses a specific range of values, typically 0.6 to 1.4, to determine the relative importance of criteria. Values less than 1.0 indicate reduced significance compared to the reference criterion, while values greater than 1.0 indicate increased significance. In order to enable the use of the known scale from 1 to 5 together with the PIPRECIA-S principles, a conversion was made between these scales.

3.3. PRIORITIZATION OF CRITERIA

The ranking of criteria according to their importance in the evaluation process of software testing methods is shown in Table 2. At the beginning of the research, the criteria were ranked on a scale from 1 to 5. Grade 1 indicates the lowest priority, and grade 5 the highest. The obtained scores were aggregated, and the average values were used as input data for the PIPRECIA-S method. Through an iterative process, participants adjusted their ratings. The criteria are ranked for the purposes of this research, but these results may differ depending on the software development context, requirements and goals of the specific project. Table 2 also shows the evaluation results of software testing methods according to defined criteria.

4. RESULTS

The relative importance of software testing methods in relation to the Efficiency criterion is shown in Table 3. The evaluations of software testing methods based on this criterion were made based on the following facts:

- Black Box achieves a score of 5 because it enables quick and efficient detection of errors from the perspective of the end user, without the need to know the internal structure of the system;
- White-Box gets a score of 4 because it provides detailed code analysis and a high level of test coverage, but requires significant time and technical knowledge;
- Automated testing achieves a score of 5 because it significantly speeds up the testing process, enables repeatability and covers a large number of scenarios in a short time and

Table 1. Ranking scale

Description	Rank	PIPRECIA-S scale
Very bad	1	0.60
Bad	2	0.80
Satisfactory	3	1.00
Good	4	1.20
Excellent	5	1.40

Table 2. Evaluation results of software testing methods according to defined criteria

Specification	Notation	Rank	Black Box	White Box	Automated	Agile
Efficiency	C_1	5	Excellent (5)	Good (4)	Excellent (5)	Good (4)
Testing coverage	C_2	4	Satisfactory (3)	Good (4)	Excellent (5)	Satisfactory (3)
Ease of application	C_3	3	Good (4)	Satisfactory (3)	Satisfactory (3)	Excellent (5)
Application price	C_4	2	Excellent (5)	Satisfactory (3)	Satisfactory (3)	Good (4)
Reliability of results	C_5	5	Good (4)	Excellent (5)	Excellent (5)	Good (4)

Table 3. Relative importance of the considered software testing methods in terms of criterion C_1- Efficiency

		s_j	k_j	q_j	w_j
A_1	Black Box Testing		1	1	0,19
A_2	White Box Testing	1,2	0,80	1,25	0,24
A_3	Automated Testing	1,4	0,60	1,67	0,32
A_4	Agile Testing	1,2	0,80	1,25	0,24
				5,17	1,00

- Agile testing gets a score of 4 because it effectively supports iterative development and enables rapid adaptation to changes, but may be less focused on deeper analysis of specific problems.

The relative importance of software testing methods in relation to the Test Coverage criterion is shown in Table 4. The evaluations of software testing methods based on this criterion were made based on the following facts:

- Black-Box gets a score of 3 because it covers functionality from the user's perspective, but does not include internal logic errors or system structure;
- White-Box achieves a score of 4 because it enables deeper testing of internal system components and detailed code analysis;
- automated testing gets a score of 5 because it enables wide coverage by testing different scenarios and a large volume of tests with minimal human effort and
- Agile testing gets a score of 3 because it focuses on continuous adaptation and iteration, but does not provide comprehensive coverage of all scenarios.

Table 5 presents the relative importance of software testing methods in relation to the Ease-of-Use criterion, with the following conclusions:

- Black-Box gets a score of 4 because it is easy to implement and does not require technical knowledge about the system structure;
- White-Box achieves a score of 3 because it requires detailed knowledge of the code and more technical skills, which makes it difficult to apply in some situations;
- Automated testing gets a score of 3 because it requires initial setup of scripts and tools, which can be technically demanding and
- Agile testing achieves a score of 5 because it enables quick adaptation and easy integration into iterative software development processes.

The relative importance of software testing methods in relation to the Cost of Implementation criterion is shown in Table 6. The evaluations of software testing methods based on this criterion were made on the basis of the following conclusions:

- Black-Box achieves a score of 5 because it is cost-effective and requires minimal resources compared to other methods;
- White-Box gets a score of 3 because it requires more time, professional staff and resources, which increases the cost of implementation;
- Automated testing gets a score of 3 because the initial setup of tools and scripts can be expensive, but the costs are reduced in the long run and
- Agile testing gets a score of 4 because the costs are adjusted to the needs of the project and are usually lower compared to methods that require complex tools.

Table 4. Relative importance of the considered software testing methods in terms of criterion C_2- Testing coverage

		s_j	k_j	q_j	w_j
A_1	Black Box Testing		1	1	0,20
A_2	White Box Testing	1,2	0,80	1,25	0,25
A_3	Automated Testing	1,4	0,60	1,67	0,34
A_4	Agile Testing	1,0	1,00	1,00	0,20
				4,92	1,00

Table 5. Relative importance of the considered software testing methods in terms of criterion C_3 - Simplicity of application

		s_j	k_j	q_j	w_j
A_1	Black Box Testing		1	1	0,21
A_2	White Box Testing	1,0	1,00	1,00	0,21
A_3	Automated Testing	1,0	1,00	1,00	0,21
A_4	Agile Testing	1,4	0,60	1,67	0,36
				4,67	1,00

Table 7 shows the relative importance of software testing methods in relation to the criterion Reliability of results, with the following conclusions:

- Black-Box receives a score of 4 because the results reliably reflect functional errors, but do not include internal system problems;
- White-Box achieves a score of 5 because it provides detailed and precise information about errors within the system;
- Automated testing receives a score of 5 because it provides consistent and repeatable results with minimal human error and
- Agile testing gets a score of 4 because it provides reliable results in the context of iterative cycles but may miss details due to the fast pace of development.

Figure 1 shows the final ranking order of the analysed software testing methods based on the importance of the given criteria.

5. DISCUSSION

The evaluation results of the four selected software testing methods based on the five analysed criteria show significant differences in the performance of the analysed methods.

Automated testing received the highest overall value and was ranked first. This suggests that automated testing provides the best results compared to other methods, primarily due to its high efficiency, test coverage and reliability of results.

Table 6. Relative importance of the considered software testing methods in terms of criterion C_4 - Cost of application

		s_j	k_j	q_j	w_j
A_1	Black Box Testing		1	1	0,24
A_2	White Box Testing	1,0	1,00	1,00	0,24
A_3	Automated Testing	1,0	1,00	1,00	0,24
A_4	Agile Testing	1,2	0,80	1,25	0,29
				4,25	1,00

Table 7. Relative importance of the considered software testing methods in terms of criterion C_5 - Reliability of results

		s_j	k_j	q_j	w_j
A_1	Black Box Testing		1	1	0,18
A_2	White Box Testing	1,4	0,60	1,67	0,30
A_3	Automated Testing	1,4	0,60	1,67	0,30
A_4	Agile Testing	1,2	0,80	1,25	0,22
				5,58	1,00

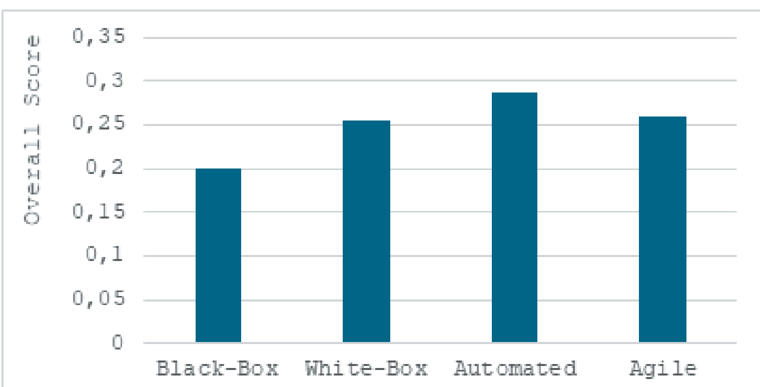

Figure 1. Final ranking of software testing methods

These factors make this method particularly suitable for situations that require high precision and coverage in testing, although the cost of implementation is relatively lower compared to the advantages that this method provides.

Agile testing, which ranked second, shows balanced performance across all criteria. Its strongest point is its ease of deployment, making it an ideal choice in agile environments where flexibility and speed are key. However, its reliability of results lags behind automated testing, which can be a challenge in complex projects.

White-Box Testing took third place with overall value. Its advantages are the efficiency and reliability of the results, but the relatively lower score in test coverage and ease of application indicates that this method is more suitable for specific cases where a detailed understanding of the internal logic of the system is required.

Black-Box Testing ranked last. Although this method has the advantage of simplicity of application, its results in criteria such as reliability of results and efficiency show that it is less suitable for projects where reliability and coverage are crucial. This indicates that this method is best suited for the initial stages of testing or for less complex systems.

Although agile testing and white-box testing have achieved similar results, the choice between the two methods depends on the specific requirements of the project. Automated testing stands out as the most effective option, while Black-Box Testing remains the least effective choice for complex projects.

6. CONCLUSION

The analysis carried out using the PIPRECIA-S method showed that different software testing methods have specific advantages and limitations depending on the analysed criteria. Automated testing has been identified as the most reliable and efficient method, thanks to its ability to provide high test coverage and reliable results, with relatively low implementation costs. On the other hand, Black-Box Testing, while simple to implement, falls short in key categories, making it more suitable for less complex projects. Agile testing has proven to be a method that balances flexibility and reliability, while White-Box Testing provides high reliability in specific cases, but requires greater technical knowledge. Based on the obtained results, the choice of the optimal testing method should be adapted to the specific requirements of the project, taking into account the criteria of efficiency, costs and complexity of implementation.

Conducting this research has practical and scientific significance. On the one hand, the results can help software teams and managers to make better decisions when choosing testing methods. On the other hand, the research contributes to the theoretical understanding of the application of multicriteria analysis in the domain of software testing. Future research can extend this analysis by including additional methods and criteria, to enable even more precise selection of test strategies in different industrial settings.

REFERENCES

[1] M. Bajjouk, M. E. Rana, C. Ramachandiran and S. Chelliah, "Software testing for reliability and quality improvement", *Journal of Applied Technology and Innovation*, vol. 5, no. 2, 2021, pp. 40-46. [Online]. Available: https://www.researchgate.net/publication/359617048_Software_testing_for_reliability_and_quality_improvement [Accessed: February 10, 2025].

[2] U. Azeem & A. Abhineet, "Importance of Software Testing in the Process of Software Development", *International Journal for Scientific Research & Development,* vol.6, no.12, 2019, pp. 141-145. [Online]. Available: https://www.researchgate.net/publication/331223692_Importance_of_Software_Testing_in_the_Process_of_Software_Development [Accessed: January 22, 2025].

[3] S. Popović, D. Viduka, A. Bašić, V. Dimić, D. Djukic, V. Nikolić & A. Stokić, "Optimization of Artificial Intelligence Algorithm Selection: PIPRECIA-S Model and Multi-Criteria Analysis", *Electronics*, vol. 14, no. 3, 2025, pp. 562. [Online]. Available: https://doi.org/10.3390/electronics14030562 [Accessed: March 11, 2025].

[4] U. Mubarak Albarka, "A Study of Software Testing: Categories, Levels, Techniques, and Types",*International Journal of Advance Research, Ideas and Innovations in Technology,* vol. 5, no. 6, 2019, pp. 32-40. [Online]. Available: http://dx.doi.org/10.36227/techrxiv.12578714 [Accessed: March 10, 2025].

[5] S. Nidhra & J. Dondeti, "Black Box and White Box Testing Techniques - A Literature Review", *International Journal of Embedded Systems and Applications,* vol. 2, no. 2, 2012, pp. 29-50. [Online]. Available: http://dx.doi.org/10.5121/ijesa.2012.2204 [Accessed: March 10, 2025].

[6] T. Capote, "A comparative study of Balck box and White box testing techniques in modern software develeopment", *Frontiers In Engineering And Technology (FET)*, vol. 5, no. 1, 2024, pp.1-7. [Online]. Available: https://iaeme.com/Home/issue/FET?Volume=5&Issue=1 [Accessed: March 5, 2025].

[7] M. Ehmer & F. Khan, "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques", International Journal of Advanced Computer Science and Applications, vol. 3, no. 6, 2012, pp.12-15. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2012.030603 [Accessed: March 10, 2025].

[8] P. Mahajan, H. Shedge, & U. Patkar, "Automation Testing In Software Organization", International Journal of Computer Applications Technology and Research, vol. 5, no. 4, 2016, pp. 198-201. [Online]. Available: http://dx.doi.org/10.7753/IJCATR0504.1004 [Accessed: March 15, 2025].

[9] W. Gopalkrishna, J. Shah & P. Nanavati, "Agile software dewelopment", International Journal of Engineering Applied Sciences and Technology, vol. 5, no. 12, 2021, pp.73-78. [Online]. Available: http://dx.doi.org/10.33564/IJEAST.2021.v05i12.011 [Accessed: March 18, 2025].

[10] D. Stanujkic, D. Karabasevic, G. Popovic & C. Sava, "Simplified Pivot Pairwise Relative Criteria Importance Assessment (Piprecia-S) Method" , Romanian Journal of Economic Forecasting, vol. 24, no. 4, 2021, pp. 141–154. [Online]. Available: https://www.researchgate.net/publication/357736315_SIMPLIFIED_PIVOT_PAIRWISE_RELATIVE_CRITERIA_IMPORTANCE_ASSESSMENT_PIPRECIA-S_METHOD [Accessed: December 20, 2024].