ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

UDC 004.42-519.816:005.8 doi: 10.32620/reks.2025.1.20

Alexander LYSENKO[1], Igor KONONENKO[1,2]

[1] *National Technical University "Kharkiv Polytechnic Institute", Ukraine*
[2] *Azerbaijan University of Architecture and Construction, Republic of Azerbaijan*

HOW TO SELECT THIRD-PARTY LIBRARY: HARNESSING VISUAL INSIGHTS AND SYSTEMATIC EVALUATION FOR INFORMED DECISIONS

The subject *of this study is a selection approach for third-party libraries in IT projects based on the collection and analysis of expert evaluations and solving optimization problems.* ***The aim*** *of this study is to minimize the key challenges associated with migrating third-party libraries, including high labor intensity and significant risks that could adversely affect project success. This study sets out several* ***objectives****: 1) to create a metric specification to guide evaluation data collection through a survey, focusing on technical metrics. Also, it needs to include recommendations for surveys development; 2) formalize a two-step method that enriches the outcomes and allows for their combined use, where the first step employs radial diagrams to visualize and simplify alternative assessments, and the second step applies the TOPSIS method to address the multi-criteria decision-making problem of selecting a third-party library; 3) the practical application of the proposed approach through illustrative samples.* ***Results.*** *The proposed metrics specification considers the essential technical metrics and offers a unified evaluation solution. We do not limit our approach to technical metrics or numeric data. It is both possible and recommended to include human and economic metrics and to use alternatives to numeric data. The combination of multi-criteria analysis and visual methodologies simplifies the assessment and comparison of libraries, making this approach valuable for developers in practical scenarios.* ***The conclusions*** *of this study emphasize that its findings may lay the groundwork for developing comprehensive decision-support systems focused on third-party library selection, thereby minimizing associated risks. Future developments will include the incorporation of other metrics for library selection, as well as consideration of uncertainty and data subjectivity.*

Keywords: *specification; method; third-party libraries; multi-criteria decision-making; TOPSIS; decision-support systems; radial diagrams; surveys; information technology; libraries migration.*

1. Introduction

This study focuses on third-party libraries, which allow us to reuse tested solutions but might pose challenges during replacement or migration processes, particularly in ensuring a systematic and objective evaluation. Third-party libraries are typically integrated into IT projects to facilitate reuse of ready-made and time-tested solutions.

1.1. Motivation

More than 20 papers have found that the process of migrating third-party libraries into IT projects is still under discussion [1]. Meaning that R&D related to the library migration process are ongoing, but the existing results remain fragmented and incomplete, requiring improvement and systematization.

It's known that this process may be prompted by various key circumstances: outdated technologies [2], cost reduction in using tools [3], adding new functionality [4, 5], insufficient documentation volume, and resolving existing product issues [1].

The criticality of selecting a third-party library and its subsequent integration into an IT project lies in the significant workload for the project development team [6, 7] and the potential risks that may impact the overall project's success [8]. The likelihood of risks generally depends on the competence level of the development team involved in the process [9]. This probability is also influenced by how well the team conducts a technical analysis of the selected third-party library and integrates it into the IT project.

In the meantime, the most challenging step in the migration process is selecting a third-party library to replace the existing one [10], which requires considering a set of metrics and ensuring that these metrics are objectively assessed [8, 9].

These drivers motivate the investigation and development of robust practical methods to address or minimize the challenges associated with third-party libraries' migration processes.

1.2. State of the art

Through research, scientists [1, 8] have identified several factors, each covering a specific area of impact on IT projects from a practical perspective. Each factor is represented by a set of metrics grouped by their area (or class) of influence on the IT project. Factors considered by development teams include technical, human, and economic aspects.

When selecting third-party libraries, considering a structured set of metrics is crucial for ensuring informed decision-making and minimizing risks [8]. These metrics provide a basis for evaluating libraries across multiple dimensions. By systematically assessing libraries based on these metrics, development teams can minimize workload [6, 7], reduce risks [8], optimize resource utilization for development team [3], and enhance team motivation [11].

Structured evaluation approaches that incorporates these metrics will allow development teams to make data-driven choices, balancing benefits and trade-offs. These systematic approaches will mitigate potential issues related to software deprecation [1], migration efforts [8], and unexpected technical debt [12], ultimately contributing to the success and longevity of the project. At the same time, clearly formalized approaches and methods specifically adapted for selecting third-party libraries are lacking [1].

The authors [13] proposed an approach that might fit the identification of library migrations by analyzing large datasets related to software development — specifically, code change histories. This approach searches for migration process patterns and then filters them based on their frequency or associated code changes. However, the usefulness of such approaches is limited because low accuracy results in significant human effort during inspection, and low coverage hinders developers from making optimal decisions because some migration opportunities may be overlooked.

To enhance the efficiency of the above approach, other researchers [14] proposed a new approach for the automated recommendation of target migration libraries based on historical software development data. Unlike filtering-based methods, the proposed approach focuses on ranking because relative ranking positions are more resilient to changes in metrics. The candidate libraries for migration were ranked using a combination of four carefully designed metrics: rule support, message support, distance support, and API support. These metrics are designed to extract different sources of evidence from the data and determine the most likely migration targets based on the evidence. Finally, the relevant target libraries, their corresponding metrics, and associated migration cases are returned to the human inspection.

The last but not least, the text search approach [15] provides a promising solution to the challenges of third-party library recommendation by leveraging the rich informational content of software product and development tool descriptions. The textual data are analyzed to establish meaningful relationships between libraries and development needs. By identifying relevant connections within large datasets, this enables a more informed and effective recommendation system, reducing reliance on prior developer knowledge and offering a broader, context-aware perspective on potential library choices.

The authors [16] suggested that establishing a knowledge base could provide an alternative solution at the corporate or organization level rather than on a global scale. This knowledge base can compile selection outcomes derived from systematically organized data collected from various surveys, as referenced in [1, 8], which form the foundation for library evaluations. It can serve as a valuable tool for offering both quick and detailed assistance during decision-making. Moreover, it becomes feasible to design and implement a rating system for experts who have experienced similar cases of migrating relevant third-party libraries. Incorporating expert opinions, validated by their ratings, in a weighted manner can streamline the selection process for the most effective and reliable solution to replace an existing library. However, establishing and implementing such an expert rating system presents unique challenges, requiring a comprehensive approach and regular evaluation criteria reviews.

This concept is also not dismissed at the global level. In our opinion, an effective strategy for selecting third-party libraries could leverage the collective experience of hundreds or even thousands of experts. To achieve this, it would be essential to create a platform similar to Gartner Peer Insights [17], where experts from various organizations could share their insights on specific third-party libraries and their application in different products or case studies. Carefully selecting experts who are permitted to submit opinions will help to prevent misinformation and malicious intent, ultimately enhancing the quality of recommendations.

Furthermore, the selection process could be enhanced with visual or algorithmic support, which is the primary focus of this study.

1.3. Objectives and approach

The aim of this study is to minimize the key challenges associated with migrating third-party libraries, including high labor intensity and significant risks that could adversely affect project success.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

This study establishes several objectives to enhance the library selection process:

1) a concise and informative metrics specification should be created in the form of a survey that serves as the foundation for gathering evaluation data. To streamline the process, the survey will focus exclusively on technical metrics based on existing systematization [1, 8], thereby enabling a clear assessment of the concept. In addition, it is worth including recommendations for constructing surveys that guarantee accurate and objective evaluations (see Section 2);

2) to formalize a two-step method that enriches the outcomes and allows for their combined use, where the first step employs radial diagrams to visualize and simplify alternative assessments, and the second step applies the TOPSIS method to address the multi-criteria decision-making problem of selecting a third-party library (Section 3);

3) demonstrate the practical application of the proposed metrics specification and method using illustrative samples (Section 3).

The approach to achieving the objectives is based on the following statements:

– before building a decision-support system, it initially requires defining metrics specifications for describing third-party libraries and exploring existing mathematical methods for solving multi-criteria problems;

– the task of selecting a third-party library is inherently multi-criteria because it involves simultaneously considering a large number of metrics that influence project success and, in some cases, may affect each other [8].

Then, the approach is considered the method that consists of two independent steps, each enriching the outcomes and allowing for their use in conjunction.

The first step employs radial diagrams to visualize evaluation results, thereby providing a quick and effective way to assess alternatives. This approach significantly simplifies the decision-making process.

The second step uses an adapted TOPSIS method, which addresses the multi-criteria decision-making problem of selecting a third-party library. The TOPSIS method ranks alternatives based on their proximity to an ideal solution via normalized and weighted metric evaluations.

2. Metrics specification

2.1. Technical factors influencing selection

We examine the technical factors with their metrics in detail within a practical context and use only these metrics to solve the problem of selecting a third-party library for an IT project. We limit the scope to technical metrics only to simplify the sample solution, ignoring human and economic factors because their processing will be the same.

Researchers [18, 19] have identified more than three technical factors influencing the selection of a third-party library. To simplify the material, we consider only three technical factors.

To obtain an accurate, significant, and consolidated quantitative evaluation of the suitability of a third-party library for each factor, we consider the existing quantitative metrics (Table 1). Researchers have systematically grouped and categorized these metrics [18, 19], but we proposed another visualization of them in a table format. In addition, we introduce metrics for the *Software system* factor.

Each technical factor and metric is briefly explained to provide clarity for subsequent evaluations.

Green- or brown-field projects of the software system. This factor is an exception to the other factors, as it directly considers the project in which the replacement of a third-party library will occur. It is important to evaluate how well a third-party library will fit into (or integrate with) the project ecosystem, whether it is newly designed or already established.

Table 1

Relationship between technical metrics and factors influencing the selection of a third-party library

Focus	Project	Third-party library	
Factors	Software system (green- or brown-field)	Functionality	Quality
Metrics	– Architecture presence, – Third-party dependencies, – Goals and objectives novelty, – Team proficiency in library usage, – Team competency.	– Size, – Complexity, – Suitability for project purposes.	– Alignment with architecture, – Usability, – Documentation, – Security, – Performance, – Testing.

A green-field project refers to one designed or built from scratch, with no existing infrastructure or dependencies. A brown-field project, on the other hand, refers to one that has already existed and has been maintained for some time. Integrating a new library into such a project often requires consideration of compatibility with the existing ecosystem.

Architecture presence. Greenfield projects typically exhibit minimal or no architectural presence because the architecture is either still in the design phase or only partially developed. As a result, third-party libraries are often integrated more easily into such projects [20].

Third-party dependencies. Greenfield projects are characterized by low coupling to existing dependencies [20] and a reduced likelihood of conflicts between dependencies. The priorities for dependencies in these projects often lean toward new and innovative solutions from a technological perspective.

Goals and objectives novelty. Greenfield projects often involve new ideas and concepts [18, 21] that have not yet been implemented by the company, development team, or market. For third-party libraries, it is important to evaluate the set of tasks they address or the ready-made solutions they provide as well as assess their degree of novelty for the project.

Team competency. This metric reflects the level of knowledge, skills, and experience of the project team members. This is a critical aspect because the team's expertise and experience significantly impact the success of project tasks, especially when using third-party libraries. These libraries often include tested and well-measured technical solutions that require a certain level of proficiency for effective use.

Library functionality. Third-party libraries provide limited functionalities that can be implemented by them. Libraries that are more likely to be selected are those with a strong alignment between their functionalities (or requirements) and the project's goals and objectives [19], ensuring suitability for the intended purpose.

Size. It is determined by the amount of code used to implement its functionality [22]. This can be assessed by examining the size of the compiled files or the number of modules, files, and lines of code.

Complexity. The complexity is characterized by the ease of integration or configuration into an IT project, as well as the number of dependencies contained within the third-party library.

Library quality. The success of integrating a third-party library and the overall functioning of an IT project ultimately depends on the technical quality metrics of the library.

Alignment with project architecture. This metric evaluates how well a third-party library integrates with the existing project architecture and adheres to established architectural principles [23]. Since this metric can be subjective, the following quantitative metrics can be considered: alignment with existing components in the IT project, adherence to architectural principles, and ease of integration into the IT project.

Usability. Usability is characterized by the following technical metrics [24]: ease of learning, ease of use, and ease of understanding by the development team.

Documentation. Refers to the collection of known knowledge that is used during prototyping and concept creation to help understand and effectively use a third-party library. The evaluation is based on the completeness and quality of the provided material [25].

Security. Awareness of vulnerabilities can be achieved by analyzing the presence of security audits, monitoring the frequency and severity of vulnerabilities, and assessing adherence to recommended security practices [26].

Performance. The efficiency is measured in terms of code execution speed, memory and resource usage, and scalability under load.

Testing. This is demonstrated through regular updates, visible project activity, and code test coverage of the third-party library [27].

2.2. Metrics specification

The task of selecting a third-party library is multi-criteria because it simultaneously considers several metrics that affect the project's success and may influence each other.

The list of technical metrics can be considered a set of variables to be used further in the multi-criteria decision-making process. The variables are expected to contain numerical values ranging from 0 to 10.

To enhance the visualization and clarity of the variable definitions, we propose a structured way of describing each variable in a table format (Table 2). Each variable should include the metric name, its designation, human-readable labels mapped to a scale of 0 to 10, and other relevant details.

These variable definitions can serve as the foundation for building a comprehensive survey.

Inspired by the concept of a rating scale for surveys from the appendices to Agile practical guidelines [28], we propose levels with humanized labels corresponding to the scale from 0 to 10, which makes navigation through the scale easier for users (see Table 2). This can consist of three levels – two extremes (e.g., 0 and 10) and one median (e.g., 5).

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Table 2

How to describe the technical metrics in bulk in a survey

Name	Designation	Labels mapping to extremes and median values										
Project architecture presence	x_1	0	1	2	3	4	5	6	7	8	9	10
		significant				moderate				none		
Project third-party dependencies	x_2	0	1	2	3	4	5	6	7	8	9	10
		high				moderate				none		
Project goals and objectives novelty	x_3	0	1	2	3	4	5	6	7	8	9	10
		standard				somewhat new				significantly new		
Team proficiency in library usage	x_4	0	1	2	3	4	5	6	7	8	9	10
		none				moderate				high		
Team competency	x_5	0	1	2	3	4	5	6	7	8	9	10
		none				moderate				high		
Library suitability for project purposes	x_6	0	1	2	3	4	5	6	7	8	9	10
		none				average				high		
Library size	x_7	0	1	2	3	4	5	6	7	8	9	10
		large				medium				compact		
Library complexity	x_8	0	1	2	3	4	5	6	7	8	9	10
		high				moderate				low		
Library alignment with project architecture	x_9	0	1	2	3	4	5	6	7	8	9	10
		none				partial				full		

The lower the value of the variable and the closer it is to 0, the more likely the project belongs to the brownfield, which is considered a negative indicator when replacing a third-party library or indicates a negative impact of the replacement on the project. This is because there is a potential risk associated with the complexity of integrating the third-party library into the project, conflicts with other implemented third-party libraries, increased labor intensity, higher costs, and so on.

It is worth noting that metric evaluations are not always within the scale (from 0 to 10). In the case of technical metrics related to functionality, quantitative values may need to be normalized to align with the scale of variables. Any normalization method can be applied to address this issue. For example, linear transformation can be performed.

The selection of third-party libraries is based on the evaluation of the defined and outlined metrics to form a structured selection method.

3. Selection method

This method comprises two independent steps, each enhancing the evaluation outcomes and offering the flexibility to be used either individually or in combination.

Each step represents a quick or express way to select a third-party library.

3.1. First step based on radial diagrams (Visual)

The first step is more visual and user-friendly and is designed for simplicity and quick insights.

Typically, metrics are evaluated by multiple team members (Table 3). The number of team members can vary based on their expertise, mastery of the relevant third-party libraries, team leadership style, team capacity, and the importance of migration and its impact on the project.

Table 3

Sample of the development team evaluation of technical metrics (Table 2)

	x_1	x_2	x_3	x_4	x_5
Manager	5	8	8	5	6
Architect	5	5	4	6	8
Developer	7	7	7	4	4

In expert evaluations, the number of experts is less important than the knowledge of the third-party libraries being assessed. One individual with a deep understanding of the subject can be more valuable than a group that only has a superficial grasp of the subject. It is useful to include members of the project team who have experience with the third-party library in question, have studied it specifically, have been trained in it, or have taken advantage of other educational opportunities.

In such cases, the insights gained can be vital for decision-making. If team members lack adequate expertise in the subject area, it is wise to consult external specialists.

It is expected that the evaluations will be within a scale of 0 to 10 (see Table 2), where x_1 represents the project architecture presence; x_2 – the project third-party dependencies; x_3 – the project goals and objectives novelty; x_4 – the team proficiency in library usage; x_5 – represents the team competency.

To simplify things, we only selected a few metrics to demonstrate in the sample. In a real scenario, this would include a table with all the metrics.

The consolidated evaluations of the development team concerning metrics can be visualized using radial charts (Fig. 1, a). The idea of using radial charts was emphasized in the Agile practical guidelines [28] as a suitable dashboard tool. This visual clarity helps stakeholders and users quickly identify which metrics are most valuable, critical, or impactful for the given task or decision-making process. Radial charts simplify the prioritization of focus areas.

It is worth noting that obtaining consolidated evaluations is a pivotal task, particularly due to the vulnerability of arithmetic averages to significant deviations (Fig. 1, b). This means that outliers or extreme values in the data can disproportionately skew the results, leading to inaccurate or misleading evaluations.

Various alternative evaluation approaches can be considered, such as finding a compromise through discussions led by a manager, assessments provided by experts, or leveraging mathematical methods. For simplicity, we use the median of the evaluations as a simpler, commonly used, and acceptable method (Fig. 1, a).

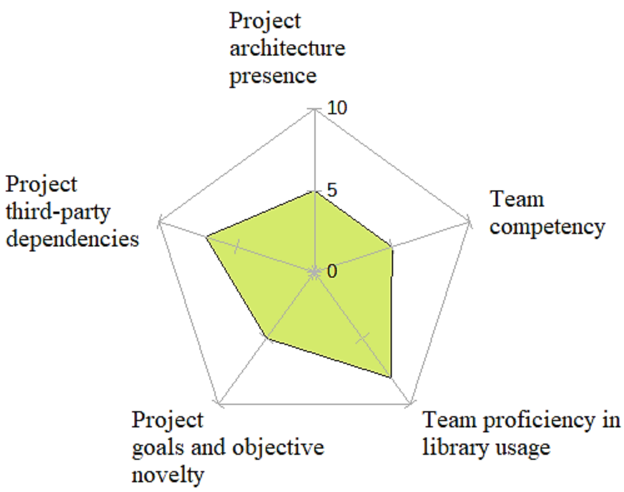
At the same time, representing all metrics from all factors on a single chart is not recommended because it may result in a loss of precision and focus (Fig. 2). Instead, it is more appropriate to visualize metrics within specific factors, as systematized by researchers [1, 8]. These factors contain a limited set of metrics, such as library affiliation to the green- or brown-field project (Fig. 1, a), functionality (Fig. 3), and quality (where should be used the primary metrics only). This approach ensures clarity and highlights the key evaluation aspects.

Note that we should adhere to the defined set of metrics rather than replacing them with a more detailed subset, as doing so would increase the complexity of the chart. In this study, quality metrics were replaced with a subset of nested metrics (Table 4) rather than using the primary metrics to simulate a real-world practical scenario. The motivation for such a replacement may include increasing the level of detail, accuracy, or other specific requirements.

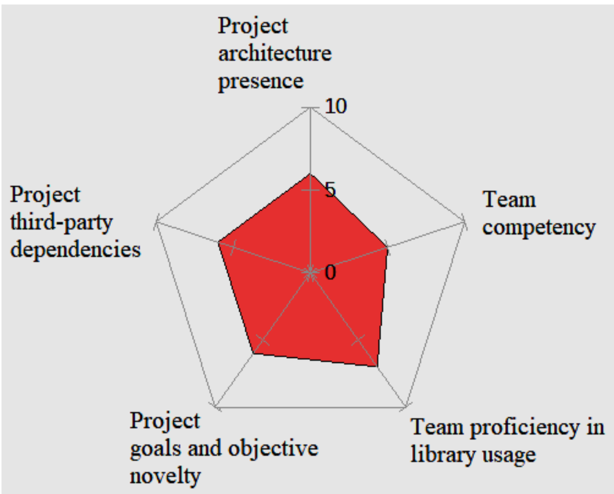
To address this situation and minimize unnecessary rework, we proposed and demonstrated a practical workaround. This solution is particularly useful when time constraints prevent adherence to the recommended visualization approach.

The workaround (see Fig. 2) involves enhancing clarity and readability by introducing nested labels for metric names and establishing relationships between primary and nested metrics (Table 4).

The nested label approach maintains a balance between providing deeper insights and ensuring that the chart remains reasonably easy to interpret, thus improving its overall usefulness and insightfulness.



a



b

Fig. 1. Sample of a library affiliation to the field project (green- or brown-):
a – constructed using the median of evaluations from Table 3;
b – constructed using the arithmetic mean of evaluations from Table 3

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

In our opinion, Fig. 2 is too information-dense. It would be better to construct it based on the primary metrics listed in Table 1. Nested metrics can be used to enhance the precision or detail of metric evaluation.

Table 4

Mapping library's quality metrics to labels

#	Primary metric	##	Nested metric
A	Alignment with project architecture	A1	Alignment with existing components
		A2	Compliance with architectural principles
		A3	Ease of integration
B	Usability	B1	Easy of use
		B2	Ease of configuration and setup
		B3	Clarity and usefulness of error messages
C	Docs	C1	Availability of comprehensive API guides
		C2	Clarity and accuracy of documentation
		C3	Availability of samples and educational materials
D	Security	D1	Frequency and severity of vulnerabilities
		D2	Adherence to best security practices
		D3	Availability of security audits
E	Performance	E1	Execution speed
		E2	Memory and resource usage
		E3	Scalability under load
F	Testing	F1	Coverage with modular tests
		F2	Quality and coverage with integration tests
		F3	Response time and issue resolution with community

For example, Fig. 3 provides a summary overview of the proposed library based on the functionality metrics. However, if we choose to explore this perspective further, we can select each primary metric to evaluate it in depth and gain insights through nested metrics.

3.2. Visual selection step sample

We consider selecting a third-party library among three alternatives A_i that are proposed. For simplicity of demonstration, we use only the technical metrics x_j for the third-party library, which are relevant to the green- or brown-project factor and within the project where the selected third-party library will be integrated (see Table 1):

– project architecture presence x_1;
– project third-party dependencies x_2;
– project goals and objectives novelty x_3;
– team proficiency in library usage x_4;
– team competency x_5.

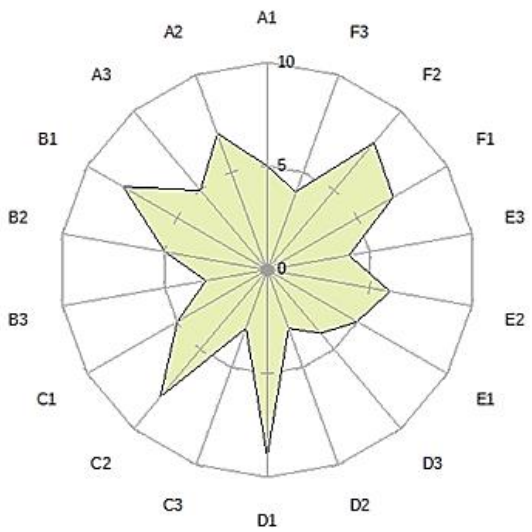


Fig. 2. Visualization sample of library's quality metrics

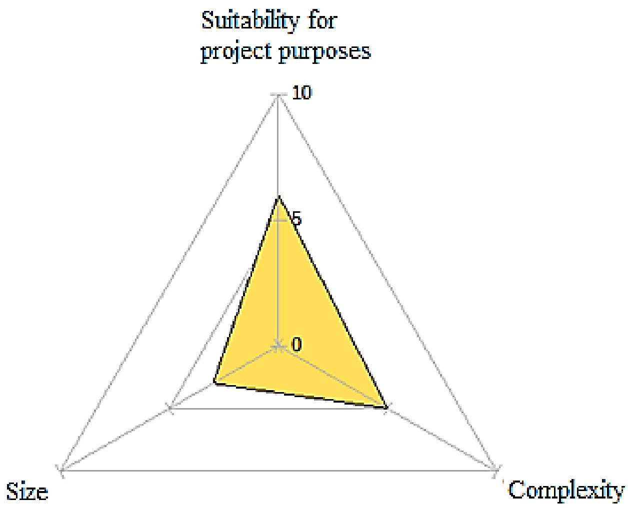


Fig. 3. Visualization sample of library's functionality metrics

The consolidated metrics evaluation from a development team for all three alternatives has the following form (Table 5).

The consolidation step can be omitted from our sample to maintain focus. However, when needed, this step can be implemented by using the median of the initial evaluations provided by the development team.

Table 5
Sample consolidated evaluation of technical metrics by the development team for the green- or brown-field project factor, as outlined in Table 1

	x_1	x_2	x_3	x_4	x_5
A_1	5	4	7	4	7
A_2	7	5	7	7	4
A_3	6	8	7	8	4

In addition, it is known that metrics such as project architecture presence x_1 and project third-party dependencies x_2 are particularly critical for the development team during the evaluation process, with x_1 carrying greater importance than x_2.

Using the data from each row in Table 5, corresponding radial charts were constructed to visually represent the evaluations – Fig. 4, Fig. 5 and Fig. 6.

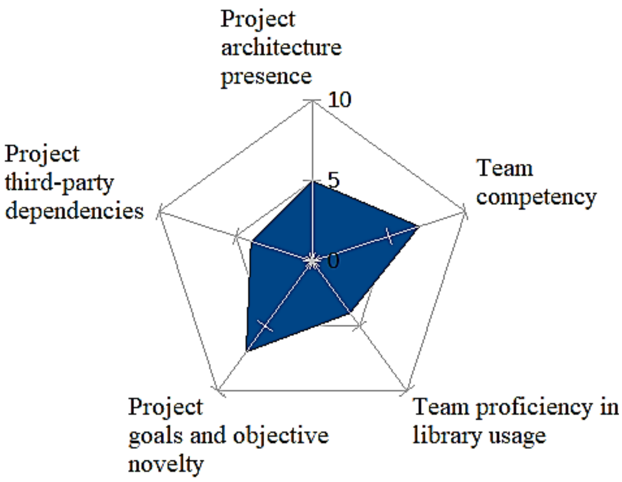


Fig. 4. Evaluations of the alternative A_1

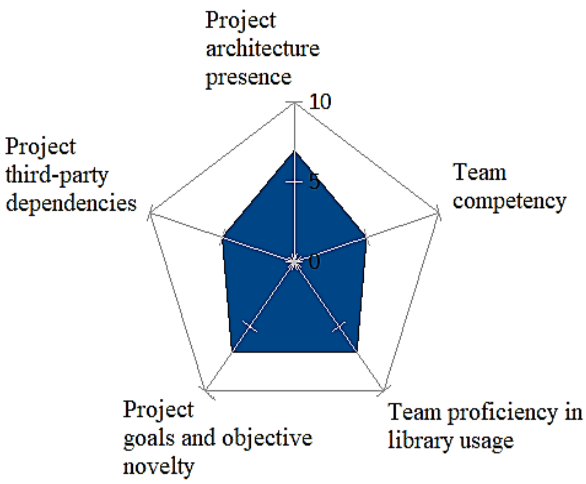


Fig. 5. Evaluations of the alternative A_2

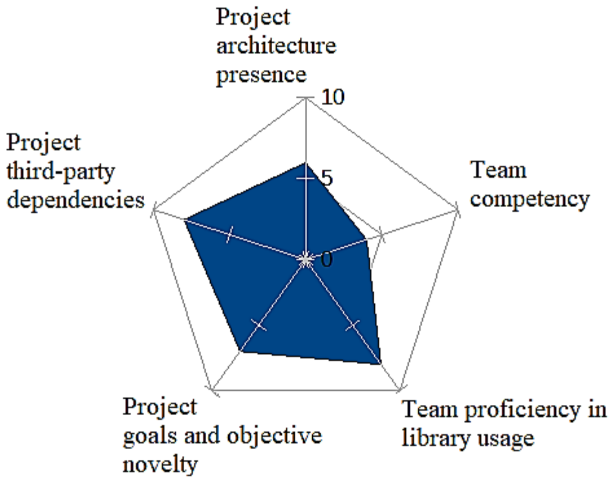


Fig. 6. Evaluations of the alternative A_3

Conclusion. Considering that project architecture presence x_1 and project third-party dependencies x_2 are particularly critical metrics for the development team, with x_1 carrying greater importance than x_2, the ideal alternative is A_2. A_2 has the highest score for x_1, which is the most critical metric, indicating a less stale architectural foundation. A_2 also performs well in other metrics, such as team proficiency in library usage x_4.

3.3. Second step using crisp numbers (Algorithmic)

The second step leverages an established algorithmic approach to deliver a more detailed and systematic evaluation.

Given the list of technical metrics j, understanding how to evaluate them (see Table 3), a multi-criteria analysis method can be used to identify the most effective third-party development tool i for selection.

The method can be based on crisp or fuzzy numerical data. Initially, within the framework of a composite study, it can be based on crisp data to simplify hypothesis testing.

The selection step using crisp numbers is based on TOPSIS [29] (Table 6), which ranks alternatives by selecting alternatives closest to the ideal solution.

The TOPSIS method is suitable for determining the ideal tool without considering the labor costs for training, the implementation effort of an individual, the wages during implementation, and the costs of acquiring usage rights.

Since the metrics specification presents technical metrics on a scale from 0 to 10, and all metrics must be maximized, the set of criteria to be minimized becomes null (i.e., $J^{'}=\emptyset$). Therefore, it makes sense to exclude it from the calculations.

Table 6

Summary of the TOPSIS method [29]

Steps	Formalization
Step 1. Normalize the decision matrix of metric evaluations and transition to the weighted normalized matrix through the importance weights	$v_{ij}=w_j\times\frac{x_{ij}}{\sqrt{\sum_i x_{ij}^2}}$, (1) where i represents a library as an alternative, $i=\overline{1,m}$; j represents a technical metric as an evaluation criterion, $j=\overline{1,n}$; x_{ij} is an evaluation of metric j for library i; w_j is a weight of importance for metric j, $w_j=\overline{0,1}$, $\sum_{j=1}^{n} w_j=1$; v_{ij} is a weighted evaluation of metric j for library i.
Step 2. Determine the best and worst libraries as selection alternatives	$I^+=\{v_1^*,\ldots,v_n^*\}$, $v_j^*=\{\max_i(v_{ij})\ \text{ if } j\in J;\ \min_i(v_{ij})\ \text{if } j\in J'\}$, (2) $I^-=\{v_1',\ldots,v_n'\}$, $v_j'=\{\min_i(v_{ij})\ \text{ if } j\in J;\ \max_i(v_{ij})\ \text{if } j\in J'\}$, (3) where J represents the set of criteria to maximize; J' represents the set of criteria to minimize.
Step 3. Calculate the distance of each alternative from the ideal solution	$S_i^+=\sqrt{\sum_j (v_j^*-v_{ij})^2}$, $S_i^-=\sqrt{\sum_j (v_j'-v_{ij})^2}$, (4)
Step 4. Select the alternative	$C_i^*=\frac{S_i^-}{S_i^-+S_i^+}$, $0 < C_i^* < 1$. (5)

The adapted TOPSIS method modifies formulas 2 and 3:

$$v_j^*=\{\max_i(v_{ij})\ \text{ if } j\in J\}, \tag{6}$$

$$v_j'=\{\min_i(v_{ij})\ \text{ if } j\in J\}, \tag{7}$$

3.4. Algorithmic selection step sample

We consider the same task as it was in the section *Visual selection step sample*. The initial decision matrix for the consolidated metrics evaluation has the following form:

$$\begin{matrix} \\ A_1 \\ A_2 \\ A_3 \end{matrix}\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ 5 & 4 & 7 & 4 & 7 \\ 7 & 5 & 7 & 7 & 4 \\ 6 & 8 & 7 & 8 & 4 \end{pmatrix}.$$

The first step involves normalizing the metric evaluations using the calculated coefficients listed in Table 7.

Table 7

Results of $\sqrt{\sum_i x_{ij}^2}$ for each metric x_{ij}

	x_1	x_2	x_3	x_4	x_5
$\sqrt{\sum_i x_{ij}^2}$	10.49	10.25	12.12	11.36	9.0

$$\begin{matrix} \\ A_1 \\ A_2 \\ A_3 \end{matrix}\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ 0.48 & 0.39 & 0.58 & 0.35 & 0.78 \\ 0.66 & 0.49 & 0.58 & 0.62 & 0.44 \\ 0.57 & 0.78 & 0.58 & 0.70 & 0.44 \end{pmatrix}$$

Then, we weighed the normalized decision matrix of the metric evaluations using the weights (Table 8).

Table 8

Weights of metrics x_{ij}

	x_1	x_2	x_3	x_4	x_5
w_j	0.3	0.25	0.15	0.15	0.15

Thus, the weighted normalized matrix of metrics is as follows:

$$\begin{matrix} \\ A_1 \\ A_2 \\ A_3 \end{matrix}\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ 0.114 & 0.098 & 0.087 & 0.053 & 0.117 \\ 0.198 & 0.123 & 0.087 & 0.093 & 0.066 \\ 0.171 & 0.195 & 0.087 & 0.105 & 0.066 \end{pmatrix}$$

The second step determines the best and worst alternatives for a third-party library based on the maximization and minimization functions, respectively.

The best alternatives I^+ are

$$\begin{matrix} \\ A_1 \\ A_2 \\ A_3 \end{matrix}\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ 0.114 & 0.098 & \mathbf{0.087} & 0.053 & \mathbf{0.117} \\ \mathbf{0.198} & 0.123 & \mathbf{0.087} & 0.093 & 0.066 \\ 0.171 & \mathbf{0.195} & \mathbf{0.087} & \mathbf{0.105} & 0.066 \end{pmatrix}$$

The worst alternatives I^- are

$$\begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ \mathbf{0.114} & \mathbf{0.098} & \mathbf{0.087} & \mathbf{0.053} & 0.117 \\ 0.198 & 0.123 & \mathbf{0.087} & 0.093 & \mathbf{0.066} \\ 0.171 & 0.195 & \mathbf{0.087} & 0.105 & \mathbf{0.066} \end{pmatrix}$$

The third step calculates the distance between each alternative and the ideal solution (Table 9).

Table 9

Distances of each alternative from the ideal solution

S_i^+	S_i^-
$\begin{array}{c} A_1 \\ A_2 \\ A_3 \end{array} \begin{pmatrix} 0.138 \\ 0.089 \\ 0.058 \end{pmatrix}$	$\begin{array}{c} A_1 \\ A_2 \\ A_3 \end{array} \begin{pmatrix} 0.051 \\ 0.218 \\ 0.124 \end{pmatrix}$

Conclusion. Finally, third-party library selection can be performed based on outcomes from the adapted TOPSIS method. The best choice is A_2.

$$\begin{array}{c} A_1 \\ A_2 \\ A_3 \end{array} \begin{pmatrix} 0.27 \\ \mathbf{0.71} \\ 0.68 \end{pmatrix}$$

4. Discussion

The results obtained in the given practical scenario were consistent across both the visual and algorithmic selection steps. In contrast, the algorithmic selection step enhances the evaluation by providing a more structured, precise, and quantitative approach. While the visual selection step offers an intuitive and quick overview, the algorithmic step adds depth and rigor by incorporating weighted metrics and mathematical calculations, ensuring a more comprehensive and objective analysis of the alternatives. Together, these steps provide a well-rounded decision-making method.

An analysis conducted before this study identified a significant gap: currently, there are no clearly defined methods specifically designed for selecting third-party libraries. Furthermore, such methods should minimize workload, reduce risks, optimize resource utilization for development teams, and enhance team motivation.

We designed our metrics specification to be as concisely as possible, focusing exclusively on technical metrics. A significant addition is the incorporation of metrics related to green-field and brown-field projects within the software system. This factor is unique because it addresses scenarios in which a third-party library is replaced.

Additionally, we have re-evaluated how we visualize and group the factors and metrics proposed by other authors, consolidating them for our study. Essentially, we have simplified the presentation of the relationship between technical metrics and the factors influencing the selection of a third-party library.

Lastly, we drew inspiration from the Agile practical guidelines [28] to present the information required to evaluate these metrics. We adapted this information to fit our case study and created a clear and concise guideline on how to prepare and construct a survey for evaluation.

In addition to metrics, it is believed that having initial information about both the third-party libraries being replaced and those considered as replacements positively influences the development team's morale because it eliminates the need to spend resources on collecting and comparing metrics (or indicators) [16].

Such information should be formed and maintained by a decision-support system to ensure effective selection of third-party libraries. The use of such a system is proposed because it computerizes the regular actions of the development team during the replacement process, even if the replacement cases are irregular. In addition, the metrics used and processed are systematized, which simplifies further work.

5. Conclusions

The multi-criteria decision-making problem of selecting a third-party library was addressed using the metrics specification and method based on crisp numerical data. The proposed solution forms the initial pillar component of a broader framework to support decision-making in third-party library selection. This provides preliminary results to guide decision-making processes and serves as a foundational anchor for subsequent approaches explored throughout the study.

This study proposed a metrics specification for third-party library selection grounded in a set of technical metrics corresponding to key technical factors influencing the selection process. The concept of a survey for metrics was effectively adapted to ensure a scale of values from 0 to 10 for each metric, thereby facilitating uniform data collection and evaluation. In addition, the specification offers flexibility for enhancement by transitioning from subjective metric evaluations to heuristic or quantitative evaluations, which are then normalized for consistent analysis.

The specification enables the construction of radial charts, which serve as a powerful visualization tool for decision-making. These charts act as an express visual selection step, providing quick insights and results within a relatively short timeframe. The simplicity and clarity of these methods make them particularly valuable for initial assessments and comparative analysis of alternatives.

Complementing the radial charts, the TOPSIS

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

method is another express selection step that leverages crisp numerical data. This step ranks alternatives (or third-party library alternatives) by identifying alternatives closest to the ideal solution based on normalized and weighted metric evaluations. The outcomes derived from TOPSIS not only support immediate decision-making and serve as a validation mechanism for subsequent approaches based on alternative principles, such as fuzzy numbers. Both steps form the entire method.

The study proposes further development in this domain by exploring and developing specifications, models, and methods that incorporate fuzzy numbers. This extension enhances the accuracy and robustness of the analysis, addressing the inherent uncertainty and subjectivity in decision-making processes for third-party library selection.

Contributions of authors: conceptualization, methodology – **Igor Kononenko, Alexander Lysenko;** formulation of tasks, analysis **– Alexander Lysenko;** development of metrics specification, method, verification **– Alexander Lysenko, Igor Kononenko;** analysis of results, visualization **– Alexander Lysenko;** writing, original draft preparation – **Alexander Lysenko**; review **– Igor Kononenko**; editing **– Alexander Lysenko.**

Conflict of Interest

The authors declare that they have no conflict of interest concerning this research, whether financial, personal, authorship, or otherwise, that could affect the research and its results presented in this paper.

Financing

This study was conducted without financial support.

Data Availability

The manuscript has no associated data.

Use of Artificial Intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

All the authors have read and agreed to the published version of this manuscript.

References

1. He, H., He, R., Gu, H., & Zhou, M. A large-scale empirical study on Java library migrations: prevalence, trends, and rationales. *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2021)*, Athens, Greece, ACM, 2021, pp. 478–490. DOI: 10.1145/3468264.3468571.

2. Teyton, C., Falleri, J.-R., & Blanc, X. Mining library migration graphs. *Proceedings of the 19th Working Conference on Reverse Engineering (WCRE 2012)*, Kingston, ON, Canada, IEEE, 2012, pp. 289–298. DOI: 10.1109/WCRE.2012.38.

3. Mojica, I. J., Adams, B., Nagappan, M., Dienst, S., Berger, T., & Hassan, A. E. A Large-Scale Empirical Study on Software Reuse in Mobile Apps. *IEEE Software*, 2014, vol. 31, iss. 2, pp. 78–86. DOI: 10.1109/MS.2013.143.

4. Li, M., Wang, W., Huo, W., Wang, P., Wang, S., Wu, D., Liu, J., Xue, R., & Huo, W. LibD: Scalable and Precise Third-Party Library Detection in Android Markets. *Proceedings of the 39th International Conference on Software Engineering (ICSE 2017)*, Buenos Aires, Argentina, IEEE, 2017, pp. 335–346. DOI: 10.1109/ICSE.2017.38.

5. Nguyen, P. T., Di Rocco, J., Di Ruscio, D., & Di Penta, M. CrossRec: Supporting software developers by recommending third-party libraries. *Journal of Systems and Software*, 2019, vol. 161, article no. 110460. DOI: 10.1016/j.jss.2019.110460.

6. Alrubaye, H., Mkaouer, M. W., & Ouni, A. On the use of information retrieval to automate the detection of third-party Java library migration at the method level. *Proceedings of the 27th IEEE/ACM International Conference on Program Comprehension (ICPC 2019)*, Montreal, QC, Canada, IEEE, 2019, pp. 347–357. DOI: 10.1109/ICPC.2019.00053.

7. Chen, C., Xing, Z., Liu, Y., & Ong, K. L. X. Mining likely analogical APIs across third-party libraries via large-scale unsupervised API semantics embedding. *IEEE Transactions on Software Engineering*, 2021, vol. 47, iss. 3, pp. 432–447. DOI: 10.1109/TSE.2019.2896123.

8. Larios Vargas, E., Aniche, M., Treude, C., Bruntink, M., & Gousios, G. Selecting third-party libraries: The practitioners' perspective. *Proceedings of the 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2020)*, Virtual Event, ACM, 2020, pp. 245–256. DOI: 10.1145/3368089.3409711.

9. Pano, A., Graziotin, D., & Abrahamsson, P. Factors and actors leading to the adoption of a JavaScript framework. *Empirical Software Engineering*, 2018, vol. 23, iss. 8, pp. 3503–3539. DOI: 10.1007/s10664-018-9613-x.

10. Cox, R. Surviving software dependencies. *Communications of the ACM*, 2019, vol. 62, iss. 9, pp. 36–43. DOI: 10.1145/3347446.

11. He, H., Xu, Y., Ma, Y., Xu, Y., Liang, G., & Zhou, M. A multi-metric ranking approach for library migration recommendations. *Proceedings of the 28th IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER 2021)*, Honolulu, HI, USA, IEEE, 2021, pp. 72–83. DOI: 10.1109/SANER50967.2021.00016.

12. Zimmermann, M., Staicu, C.-A., Tenny, C., & Pradel, M. Small world with high risks: a study of security threats in the npm ecosystem. *Proceedings of the 28th USENIX Security Symposium (USENIX Security 2019)*, Santa Clara, CA, USA, USENIX Association, 2019, pp. 995–1010. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/zimmermann. (accessed 10.11.2024).

13. Koren, Y. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data*, 2010, vol. 4, iss. 1, article no. 1. DOI: 10.1145/1644873.1644874.

14. Zhao, X., Li, S., Yu, H., Wang, Y., & Qiu, W. Accurate library recommendation using combining collaborative filtering and topic model for mobile development. *IEICE Transactions on Information and Systems*, 2019, vol. E102.D, iss. 3, pp. 522–536. DOI: 10.1587/transinf.2018EDP7227.

15. Thung, F., Lo, D., & Jiang, L. Automated library recommendation. *Proceedings of the 20th Working Conference on Reverse Engineering (WCRE 2013)*, Koblenz, Germany, IEEE, 2013, pp. 182–191. DOI: 10.1109/WCRE.2013.6671293.

16. Lysenko, O. O., & Kononenko, I. V. Spetsyfika ta skladnist′ mihratsiyi storonnikh instrumentiv v IT proyektakh [Specifics and complexity of third-party tool migration in IT projects]. *Visnyk Natsional'noho tekhnichnoho universytetu «KHPI». Seriya: Stratehichne upravlinnya, upravlinnya portfelyamy, prohramamy ta proektamy – Bulletin of the National Technical University "KhPI". Series: Strategic management, portfolio, program and project management*, 2024, no. 1(8), pp. 26–32. Kharkiv, Ukraine. DOI: 10.20998/2413-3000.2024.8.4. (In Ukrainian).

17. *Gartner Peer Insights.* Available at: https://www.gartner.com/peer-insights/home (accessed 22.10. 2024).

18. Abebe, M. T., & Megento, T. L. Urban green space development using GIS-based multi-criteria analysis in Addis Ababa metropolis. *Applied Geomatics*, 2017, vol. 9, iss. 4, pp. 247–261. DOI: 10.1007/s12518-017-0198-7.

19. Alshboul, O., Shehadeh, A., Almasabha, G., Mamlook, R. E. A., & Almuflih, A. S. Evaluating the impact of external support on green building construction cost: A hybrid mathematical and machine learning prediction approach. *Buildings*, 2022, vol. 12, iss. 8, article no. 1256. DOI: 10.3390/buildings12081256.

20. Collie, B., Ginsbach, P., Woodruff, J., Rajan, A., & O'Boyle, M. F. P. M3: Semantic API Migrations. *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (ASE 2020)*, Melbourne, Australia, ACM, 2020, pp. 90–102. DOI: 10.1145/3324884.3416618.

21. Cao, Y., Kamaruzzaman, S. N., & Aziz, N. M. Green building construction: A systematic review of BIM utilization. *Buildings*, 2022, vol. 12, iss. 8, article no. 1205. DOI: 10.3390/buildings12081205.

22. Saaty, T. L. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation.* New York, USA: McGraw-Hill, 1980. 287 p.

23. Bass, L., Clements, P., & Kazman, R. *Software Architecture in Practice.* 2nd ed., Addison-Wesley, 2003. 560 p.

24. Nielsen, J. *Usability Engineering.* San Francisco, USA: Morgan Kaufmann Publishers, 1994. 362 p.

25. Robillard, M. P., & DeLine, R. A field study of API learning obstacles. *Empirical Software Engineering*, 2011, vol. 16, iss. 6, pp. 703–732. DOI: 10.1007/s10664-010-9150-8.

26. Viega, J., & McGraw, G. *Building Secure Software: How to Avoid Security Problems the Right Way*. Boston, USA: Addison-Wesley Professional, 2001. 528 p.

27. Myers, G. J., Badgett, T., Thomas, T. M., & Sandler, C. *The Art of Software Testing. 2nd ed.*, Hoboken, NJ, USA: John Wiley & Sons, 2004. 234 p.

28. Project Management Institute. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide), Sixth Edition.* Newtown Square, PA, USA: Project Management Institute, 2017. 756 p.

29. Alaoui, L. Y. Introduction to Multi-Criteria Decision Making: TOPSIS Method. *Proceedings of the Islamic Financial Engineering Laboratory (IFE-Lab) Seminar*, Mohammadia School of Engineering, Rabat, Morocco, July 2019. DOI: 10.13140/RG.2.2.36465.22882.

Received 17.12.2024, Accepted 17.02.2025

Radioelectronic and Computer Systems, 2025, no. 1(113)
ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

ЯК ВИБРАТИ СТОРОННІЙ ІНСТРУМЕНТ РОЗРОБКИ: ВИКОРИСТАННЯ ВІЗУАЛЬНИХ ІНСАЙТІВ ТА СИСТЕМАТИЧНОЇ ОЦІНКИ ДЛЯ ОБҐРУНТОВАНИХ РІШЕНЬ

О. О. Лисенко, І. В. Кононенко

Предметом дослідження є підхід до вибору сторонніх інструментів розробки для ІТ-проєктів на основі на основі збору та аналізу експертних оцінок та розв’язання оптимізаційної задачі. **Метою** дослідження є мінімізація ключових викликів, пов’язаних із міграцією сторонніх інструментів, включаючи високу трудомісткість та значні ризики, які можуть негативно вплинути на успішність проєкту. **Завдання** дослідження полягають у наступному: 1) створити специфікацію метрик, щоб керувати збором оцінок через опитування, зосереджуючись на технічних метриках. Також необхідно надати рекомендації щодо розробки опитувань; 2) формалізувати двоетапний метод, який збагачує результати та дозволяє їх комбіноване використання, де на першому етапі застосовуються радіальні діаграми для візуалізації та спрощення оцінки альтернатив, а на другому етапі використовується метод TOPSIS для вирішення багатокритеріальної задачі прийняття рішень щодо вибору стороннього інструменту; 3) продемонструвати практичне застосування запропонованого підходу на прикладах. **Результати**. Запропонована специфікація метрик враховує основні технічні метрики та пропонує єдиний підхід до оцінювання. Поєднання багатокритеріального аналізу та візуального методів спрощує оцінку та порівняння сторонніх інструментів, що робить його цінним для розробників у практичних сценаріях. **Висновки**. Дослідження підкреслює, що отримані результати створюють основу для розробки комплексних систем підтримки прийняття рішень, орієнтованих на вибір сторонніх інструментів. Подальшим розвитком буде додавання людських та економічних метрик для вибору сторонніх інструментів та врахування невизначеності і суб’єктивності даних.

Ключові слова: специфікація; метод; сторонні інструменти розробки; багатокритеріальне прийняття рішень; TOPSIS; системи підтримки прийняття рішень; радіальні діаграми; оцінювання на основі опитування; інформаційні технології; міграція сторонніх інструментів розробки.

Лисенко Олександр Олександрович – асп. каф. управління проєктами в інформаційних технологіях, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна.

Кононенко Ігор Володимирович – д-р техн. наук, проф., проф. каф. управління проєктами в інформаційних технологіях, Національний технічний університет "Харківський політехнічний інститут", Харків, Україна;
проф. каф. управління проєктами, Азербайджанський університет архітектури та будівництва, Баку, Республіка Азербайджан.

Alexander Lysenko – PhD Student of the Department of Project Management in Information Technology, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine,
e-mail: alexander.lysenko.dev@gmail.com; ORCID: 0009-0009-4962-5881, Researcher ID: MCK-6299-2025, https://scholar.google.com/citations?user=Ku_tnv0AAAAJ&hl=en.

Igor Kononenko – Doctor of Technical Sciences, Professor, Professor at the Department of Project Management in Information Technology, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;
Professor of the Project Management Department, Azerbaijan University of Architecture and Construction, Baku, Republic of Azerbaijan,
e-mail: igorvkononenko@gmail.com, ORCID: 0000-0002-1218-2791, Scopus Author ID: 57188536276, Researcher ID: O-2252-2016.