UDC 004.05; 004.4

TOOLS FOR SELECTING A SOFTWARE DEVELOPMENT METHODOLOGY TAKING INTO ACCOUNT PROJECT CHARACTERISTICS

Seniv M. M. – PhD, Associate Professor of Software Department, Lviv Polytechnic National University, Lviv, Ukraine.

Kovtoniuk A. M. – Ms. Sc., Lviv Polytechnic National University, Lviv, Ukraine.

Yakovyna V. S. – Dr. Sc., Full Professor of Artificial Intelligence Department, Lviv Polytechnic National University, Lviv, Ukraine; Faculty of Mathematics and Computer Science, University of Warmia and Mazury in Olsztyn, Poland.

ABSTRACT

Context. In the software development process, the choice of a software development methodology is one of the important stages that significantly affects the success/failure of the project. The choice of the optimal development methodology depends on many factors and is a time-consuming and nontrivial task.

Objective. Therefore, there is a need to develop an effective and flexible software tool for selecting the best software development methodology that would automate this process and take into account the key characteristics of the project.

Method. This article presents an algorithm for selecting a software development methodology using methods of multi-criteria analysis and expert evaluation, which provides for gathering of the expert evaluation and implements the process of selecting the methodology using such methods as AHP, TOPSIS and Weighted Sum.

Results. Using the above-mentioned algorithm, a software system was developed for selecting the best software development methodology depending on the characteristics of the project, where the criteria weights provided by experts were taken into account and the AHP method was applied to determine user priorities regarding the criteria for the methodology comparison. The TOPSIS and Weighted Sum method were chosen to calculate the estimates of the methodology selection. The software tool provides for the output of useful details of the selection results, namely, an expert evaluation of the specified parameter values in relation to all methodologies, and it can be used to improve the efficiency of the software development process in terms of automating the provision of recommendations to IT project managers.

Conclusions. The algorithm for selecting a software development methodology was developed, which, unlike the existing ones, provided for gathering of expert evaluation, taking into account the values of the criteria set by a user independently, and implemented the process of selecting the methodologies using such methods of multi-criteria analysis as AHP, TOPSIS and weighted sum. Using the above algorithm, a software system was developed for selecting the best software development methodology, depending on the characteristics of the project, where the criteria weights provided by experts were taken into account, and the AHP method was applied to determine user priorities for methodology comparison criteria. TOPSIS and weighted sum methods and were chosen to calculate the scores of methodology choice. The software tool provides for the output of useful details about the selection results, namely, an expert evaluation of the set parameter values regarding all methodologies.

KEYWORDS: software, software development methodologies, software engineering.

ABBREVIATIONS

TOPSIS is a Technique for Order of Preference by Similarity to Ideal Solution;

AHP is a Analytic Hierarchy Process;

CASE is a computer-aided software engineering;

PAPRIKA is a Potentially All Pairwise Rankings of all possible Alternatives;

XP is a Extreme Programming;

DSDM is a Dynamic Systems Development Method;

RAD is a Rapid Application Development;

ROC is a Rank Order Centroid;

SDLC is a Software Development life cycle;

DBMS is a Database Management System;

SWEBOK is a Software Engineering Body of Knowledge;

PRINCE is a PRojects IN Controlled Environments;

PMBOK is a Project Management Body of Knowledge;

SMARTER is a Specific, Measurable, Achievable, Realistic, Time bound, Evaluate, and Reviewed;

PIS is a positive ideal solution;

NIS is a negative ideal solution.

NOMENCLATURE

p_a – priority evaluation of alternative a;

x_{ai} – evaluation of alternative a by criterion i;

w_i – weight of criterion i;

n – number of criteria;

x_{ij} – evaluation of alternative (methodology) i by criterion j;

m – number of alternatives;

n_{ij} – normalised value of evaluation of alternative i by criterion j;

Y – set of alternatives;

N – set of criteria.

INTRODUCTION

With every passing year, the software development process becomes more complex, requiring deeper knowledge and experience from developers and project managers. The software creation is a series of processes result-

DOI 10.15588/1607-3274-2022-2-17

ing in the development of a software product. These processes are based mainly on software engineering technologies. The software development process can begin with the development of a software system from scratch, or new software is developed based on existing software systems by modifying them. The software development process, like any other intellectual activity, is based on such human factors as judgements and conclusions, i.e., is creative. As a result, the attempts to automate this process have met with only limited success. CASE tools can help in the implementation of some stages of the software development process, but they do not help much at those stages where the factor of a creative approach to development is essential [1]. The procedure for selecting a specific software development methodology also plays a significant role in the above-mentioned process. The success of the software product implementation depends on it, making this stage very important. However, due to a large number of existing methodologies, it really becomes a challenge for managers and developers to determine the one that would best suit the project task and development team. The reason is that different types of software projects require different approaches, since each category of projects has different priorities and goals; in addition, clear and standardized criteria for selecting a software development methodology have not yet been specified [2–4]. Therefore, the algorithmic support and software development for the selection of the most suitable software development methodology depending on the characteristics of the project and for various types of projects is an urgent scientific task.

Based on this, the **object of the research** is the process of selecting the methodology of software development, **the subject of the research** are algorithms, methods and tools for selecting the methodology of software development, taking into account the characteristics of the project; the **aim of the research** is to develop an effective and flexible tool for selecting the optimal methodology for software development, taking into account the characteristics of the project.

2 PROBLEM STATEMENT

Given: the set $Y = \{Y_1, Y_2, \ldots, Y_7\}$ of alternatives (software development methodologies) and the set of $N = \{N_1, N_2, \ldots, N_{23}\}$ criteria (project characteristics) with the weight of the i-th criterion w_i.

The task is to build a hierarchy in the form of a multitree and calculate the global priorities of alternatives – the priorities of alternatives for the whole hierarchy. The input data are the results of a survey of experts in the form of matrices of pairwise comparisons at all nodes of the hierarchy. Hierarchical synthesis is used to weigh the own vectors of matrices of pairwise comparisons, as well as to calculate the general priorities of alternatives. As a result of constructing a hierarchy and implementing paired comparisons, matrices of paired comparisons should be constructed for all vertices of the hierarchy except leaves. The pairwise comparison method to calculate the aggregate evaluation (global priority) of alternatives (development methodologies) should be applied.

3 REVIEW OF THE LITERATURE

Currently, various approaches are used to automate and optimize the choice of a software development methodology. One of them is rule-based expert systems [5]. Such criteria as application size, risks, project complexity, reliability, time, team size and expertise are taken into account, and a cascade model, spiral model, incremental model, XP, Scrum or RAD model can be proposed based on these characteristics. The expert system [5] uses a modular rule-based architecture. The questionnaire consists of different questions about the characteristics of the project: system type, system size, level of possible risks, complexity, reliability, etc. The experts can update or add any question from this repository. The “rule repository” is maintained as a set of “if...then” rules, it provides recommendations according to the characteristics of the project. The “set of facts” contains facts about recommendations for different possible values in rules. The answers provided by a user are placed in the relevant rules of the “rule repository”, which are used by the “rule engine” for comparing the “set of facts”, structuring and displaying recommendations to a user through the display module (“SDLC recommendation display module”) [5]. The main disadvantage of this type of system is the difficulty of filling a knowledge base. Upon the selection, as many existing software development methodologies as possible should be considered, and also many different criteria should be taken into consideration depending on the characteristics of the project. When trying to make the knowledge base as complete as possible, it is extremely difficult to predict all the details, especially considering that expert opinions often differ. Besides, users cannot change the priority of criteria in this type of system.

Another approach is described in the work [2], where an approach to solving the problem of choosing the agile methodology for small and medium-sized projects is proposed, using the multi-criteria method based on SMARTER. The proposed method for the methodology selection consists of the following stages [2]:

1. Determining a set of criteria: 13 criteria are proposed related to the setting up of work on the project, the complexity of the project and change management;
2. Developing alternative solutions: the choice is limited to four agile methodologies: DSDM, Scrum, XP and Crystal;
3. Creating an evaluation matrix: the evaluation of methodologies in relation to criteria is based on the number of scientific papers, which indicate that a certain value of the criterion is suitable for a certain software development methodology;
4. The relative importance of criteria is determined, and values of criteria weights are calculated using the ROC method;
5. The multi-attribute value of the function of each of the alternatives is set by the aggregation of functions;

DOI 10.15588/1607-3274-2022-2-17

As a result, the alternatives are ranked from best to worst [2].

Also, in [6], for the selection of practices for organizing the software development process, it is proposed to use the PAPRIKA method. 31 practices are evaluated in pairs against 11 criteria. The tool interviews users and, based on the answers, forms a list of practices that it recommends using in project development. The PAPRIKA method is based on users expressing their preferences with respect to the relative importance of the criteria or attributes of interest for the made decision or choice, by pairwise comparison (ranking) of alternatives [6].

In [7], a method for selecting a project testing technique is described, using the AHP hierarchy analysis technique and TOPSIS method. TOPSIS is based on the concept that the ideal alternative has the shortest distance from the positive ideal solution and the longest distance from the negative ideal solution. AHP is used to calculate the criteria weights. AHP uses the relative consistency ratio to verify the consistency of the criteria weights.

In turn, a tool that uses the method of selecting a project management methodology based on fuzzy representations is described in the work [7]. The method uses a questionnaire with questions related to the number of people involved in the project, the customer's experience of working with the team, evaluation of the project team's competence by the project manager, project reporting and likelihood of risk events. For each situation specified in the questionnaire, using a survey of expert opinion, the membership functions of all project management methodologies considered are determined, i.e., their applicability to a particular situation. In accordance with the answers to the questions of the questionnaire for the project, the membership functions of the project evaluation for each of its parameters are formed. For all the methodologies considered, their total weighted distances from the project evaluation according to the questionnaire are calculated using the Euclidean and Hamming distances. The approach with the calculated minimum distances is selected [8].

M. Despa in his work [9] conducted a comparative analysis of software development methodologies with an emphasis on the features of project management. The author presented and compared the stages of the development process for such methodologies as waterfall, prototyping, iterative and incremental, spiral, rapid application development, extreme programming, V-model, scrum, cleanroom, dynamic systems development methodology, rational unified process, lean software development, test-driven development, behavior-driven development, feature-driven development, model-driven engineering, crystal methods, joint application development, adaptive software development, open source software development and Microsoft Solutions Framework. Such factors affecting the software development process as frequent software requirements changes, high dynamics of the technology stack and development standards, qualifications of the development team and the team globalization and dispersion were considered in the study [9]. The author describes in detail the characteristics, advantages, and disadvantages of each of the investigated methodologies. The advantages of traditional methodologies [9] include ease of understanding and implementation, availability of substantial documentation and ease of tracking, evaluation, and reporting. The agile methodologies, in turn, provide greater flexibility and can easily adapt to changes, contributing to earlier release of working code, better self-organization of teams and adaptive planning.

G.S. Matharu with co-authors [10] explore the issue of choosing between such agile software development methodologies as Scrum, Kanban, and extreme programming. The paper presents a detailed comparison of these methodologies in terms of such parameters as design approaches, customer cooperation, project complexity, team roles, team interaction, approach to workflow organization, requirements management, coding, and testing approaches, etc. The authors [10] indicate and analyse companies that use the above software development approaches. The authors showed that currently the most widespread in the industry are the approaches based on the Scrum methodology.

L. R. Vijayasarathy and C.W. Butler [11] study the factors influencing the selection of the best software development methodology. The authors investigated the problems of the influence of a software project organizational structure and characteristics of the team and the project itself on determining the best software development methodology. The study was conducted by interviewing project managers and members of the development team on the choice of methodologies. The results [11] show that although the agile methodologies such as the Agile Unified Process or Scrum have become increasingly popular in the last decade, traditional methodologies, including the waterfall model, are still popular in the software development industry. The companies also often adopt a hybrid approach using different methodologies in the same project. Besides, the choice of methodology is associated with certain organizational, project and team characteristics and remains an urgent task of software engineering [11].

The work [12] is dedicated to the issues of modelling the software development methodologies. The authors note that although modern modelling approaches must have a strong theoretical foundation, they do contain many vague concepts or even contradictions. C. Gonzalez-Perez and B. Henderson-Sellers present an approach that analyses the basic concepts of structural models and modelling in software engineering using representation theory. The authors investigated different types of interpretive reflections needed to track model entities with the entities they represent. The paper also explains the difference between forward- and backward-looking models and considers the need to integrate products and processes into methodologies.

The article [13] analyses the software development methodologies and their main stages. The authors compare international approaches, standards, and practices for software development with the standards and practices

DOI 10.15588/1607-3274-2022-2-17

used in Pakistan. The comparative analysis shows the gaps and shortcomings of the practices adopted in Pakistan and the ways to improve them.

Another aspect of research in the field of software development methodologies is considered in the article [14], which examines the issue of ensuring that the skills and competencies of students of higher education institutions meet the requirements and expectations of the labour market. K. Saeedi and A. Visvizi emphasize the key role of teaching the software development processes and technologies for industry, economics, students, and universities. The paper points out the importance and relevance of agile development methodologies scrum, at the present stage. By analysing the problems and challenges of switching to agile software development methodologies in software projects, the article [14] concludes that software development and methodology for its development form the thrust of a multi-stakeholder ecosystem that defines today's digital economy and society.

Based on the foregoing, a conclusion can be made that high activity in the field of software development has led to the emergence of a large number of methodologies, and now the choice of a suitable approach remains a problem [15], because it usually requires quite extensive experience in software development. It is also worth noting that the problem of choosing a software development methodology is the reason for the studies, the purpose of which is to create a universal method for selecting the software development methodology. They can be divided into two types: rule-based expert systems and tools using multi-criteria analysis methods. The disadvantage of using classical expert systems is the complexity of filling them with a large amount of data and inability of users to influence the priority of criteria. In contrast to them, the existing approaches to the choice of software development methodologies, which use the methods of multi-criteria analysis, provide for the possibility of establishing criteria weights, but most of them still rely on the opinion of only one expert and a fixed set of criteria. Since there are many methodologies, the expert opinions may differ regarding the optimal values of criteria for a particular methodology. Besides, the criteria, possible values for which cannot be easily expressed in numbers, may also be considered. Therefore, there is a need to create a flexible tool that would be free of these limitations and allow automating the selection of the software development methodology, which is the most favourable for a certain project.

4 MATERIALS AND METHODS

It was decided to use the following methods of multi-criteria analysis for the process of selecting the best methodology: analytic hierarchy process, weighted sum method, TOPSIS and methods for expert evaluation. The AHP, developed by Thomas L. Saaty, is a well-known technique for multi-criteria decision making [16]. One of the distinguishing features of the AHP is the creation of a pairwise comparison matrix using a verbal scale. In the standard version of the method, the normalised eigenvector of this matrix allows calculating the score of each alternative and weight of each criterion.

The weighted sum method is the most popular method of multi-criteria analysis due to its simplicity. As the name suggests, this is simply the sum of the weighted scores:

$$p_a = \sum_{i=1}^{n} x_{ai} w_i , \quad (1)$$

We assume that the goal is to maximize all criteria.

The TOPSIS method is focused on evaluating the alternative in terms of the best and worst points [6].

1. The normalization of evaluation by criteria is carried out:

$$n_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} , \quad (2)$$

2. The weighted normalised decision matrix is calculated considering criteria weights:

$$u_{ij}=w_j n_{ij} \text{ where } i=1,\ldots,m;\ j=1,\ldots,n. \quad (3)$$

3. The PIS and NIS are determined:

$$A^+ = \frac{\max}{i u_{ij}}, A^- = \frac{\min}{i u_{ij}}.$$

4. The distance of alternatives to PIS and NIS is calculated:

$$d_i^+ = \sqrt{\sum_{i=1}^{n} (u_{ij} - A_j^+)^2} , j = 1,..., m. \quad (4)$$

$$d_i^- = \sqrt{\sum_{i=1}^{n} (u_{ij} - A_j^-)^2} , j = 1,..., m. \quad (5)$$

5. The integral index (proximity index) is determined for each compared alternative:

$$R_i = \frac{d_i^-}{d_i^- + d_i^+}. \quad (6)$$

The proximity index is between 0 and 1, where 1 is the best alternative.

5 EXPERIMENTS

Experts evaluate the extent to which it is permissible to use a certain methodology for each characteristic of the project, i.e., each possible value of the criterion.

The test data presented in the work [8] were used as the baseline, namely: 7 methodologies, a list of 23 criteria and their possible values, criteria weights, evaluation of values of criteria in relation to methodologies. A detailed list of project characteristics according to which the methodology is selected, is given in Table 1. Each of

DOI 10.15588/1607-3274-2022-2-17

them has four stages of gradation presented in the table. The weights of expert opinion may vary. Given these weights, the expert evaluation is aggregated.

It was decided to use the AHP to calculate the weights of criteria used to evaluate alternatives. The user makes a pairwise comparison of the criteria, and the absolute weights of criteria are calculated using the AHP. The pairwise comparison is made on a scale from 1 to 9. The AHP uses a consistency ratio as a measure to check the consistency of the weights obtained. This ensures that the weights are consistent.

Based on the user-defined values of criteria, their weights and expert evaluation, the system calculates the score for each methodology using the weighted sum and TOPSIS methods. The higher the score, the better the applicability of the methodology to the project.

Table 1 – Parameter values

N	**Parameter**	**Possible values**			
1.	Project cost	< 100,000	100,000–300,000	300,000 – 1,000,000	> 1,000,000
2.	Requirements change percent/month	< 7%	7%–25%	25%–45%	> 45%
3.	Number of people involved in the project	< 10 per	10–30 per	30–100 per	> 100 per
4.	Consequences in case of unsatisfactory project outcome	loss of comfort in work	Loss of insignificant sum of money	Loss of irreplaceable sum of money	Loss of life
5.	Work experience in the given field	No work experience	Experience of working in the field for less than 2 years	Experience of working in the field from 2 to 5 years	Experience of working in the field from 2 to 5 years
6.	Requirements to the realization period of the project	The period is unlimited	Not very urgent	Urgent	Very urgent
7.	Teams ability to work effectively in freedom or order	Able to work effectively in full order	Able to work effectively in middle order	Able to work effectively in partial order	Able to work effectively in full freedom
8.	Understanding of requirements, adapting ability, initiative	Almost do not understand the requirements; require frequent explanations and constant control	Understand the requirements, can follow them, but require regular control	Understand the requirements, can follow them, do not require regular control	Have good understanding of the requirements; can follow them without regular control; can suggest better alternatives
9.	Probability of occurrence of managerial risks (inefficient planning, controlling, communication problems, etc.)	Risk is not likely to occur (10%)	Probability of risk occurrence is equal (50%)	Risk is highly likely to occur (75%)	Risk will most probably occur (>95%)
10.	Knowledge of applied tools and methods	Tools and methods, applied in the given project, have never been used before and are unknown to the team	Tools and methods, applied in the project, are known to the team but have never been used before	Tools and methods, used in the project, are known to the team but are rarely used	Tools and methods, are known to the team and have been widely used before
11.	Means of communication	Written reports. Formal record-keeping	Voice communication	Online text communication	Direct communication
12.	Frequency of reporting to the Customer	Reports on every operation	Reports on completing the blocks of work	Reports on the readiness of a component of projects product	Reports about project finish
13.	Understanding the scope of works	There is a full list of works; further alternation is impossible	There is a detailed list of works, further alternation is possible	There is an approximate list of project works	The team understands the project goal and several ways for its achievement
14.	Requirements to the project quality	Highest international requirements	International requirements	National requirements	Local requirements
15.	Probability of occurrence of technical, manufacturing or qualitative risks	Risk is not likely to occur (10%)	Probability of risk occurrence is equal (50%)	Risk is highly likely to occur (75%)	Risk will most probably occur >95%)
16.	Probability of occurrence of external risks (disruption of work by contractors, unfavourable political, etc.)	Risk is not likely to occur (10%)	Probability of risk occurrence is equal (50%)	Risk is highly likely to occur (75%)	Risk will most probably occur >95%)
17.	Probability of occurrence of organizational risks (disruption of funding, delivery of resources, inaccurate prioritizing, etc.)	Risk is not likely to occur (10%)	Probability of risk occurrence is equal (50%)	Risk is highly likely to occur (75%)	Risk will most probably occur >95%)

DOI 10.15588/1607-3274-2022-2-17

Table 1 – Parameter values (continuation)

N	Parameter	Possible values			
18.	Requirements to the precise compliance with a deadline	The deadline should be strictly met	Insignificant deviation from the deadline is allowed	Considerable deviation from the deadline is allowed	Compliance with the deadline is now strictly required
19.	Ability to admit mistakes	Do not admit makings mistakes and cannot learn from them	Rarely admit their mistakes but try t never make them again	Openly admit making mistakes and try to never make them again	Openly admin making mistakes and always learn from them
20.	Learning ability	It is hard for the team to learn new knowledge and technologies, and to adjust to changes	For some members of the team, it is hard to learns new information and technologies, but the team can adjust to changes	Easily absorb new knowledge, can adjust to changes	The team can easily absorb information, always tries to learn something new; can well adjust to the changes
21.	Experience of cooperation	Have never worked together	Worked together on the creation of a product but in the different field	Worked together on the creation of one product in a field of interest	Worked together on the creation of several projects in the field of interest
22.	Teams ability to clearly formulate and openly express ideas	Cannot clearly formulate ideas and rarely express them	Can clearly formulate their ideas but rarely express them	Can clearly formulate their ideas and openly express them	Can clearly formulate, openly express and justify their ideas
23.	Customers experience of working with this project team	Has never worked with this team	Worked with some members of the team	Worked with the project team leader	One or more common projects with the whole project team

6 RESULTS

To create a tool to automate the selection of the best software development methodology for the project, an appropriate algorithm was developed, which provided for the gathering of expert evaluation and implemented the process of selecting methodologies using such multi-criteria analysis methods as AHP and weighted sum. It consists of 11 steps; its block diagram is shown in Fig. 1.

1. Filling the database with description of software development methodologies.

2. Filling the database with a set of necessary criteria, by which the characteristics of projects will be determined, with the relevant setting of initial values.

3. Setting the default weights for the criteria and, if required, the weights for individual possible values of criteria.

4. Gathering the expert evaluation of all possible values of criteria in relation to all methodologies available in the database.

5. A user must set the values of criteria in accordance with the characteristics of the project; if required, a user can omit some of the criteria.

6. If required, a user can determine the weights of criteria independently, using the AHP method. If a user refuses, then the weight of criteria takes on the default values.

7. If a user agrees to determine the weights of criteria independently:

a) A user must compare in pairs the importance of all specified criteria with each other.

b) The relative consistency of the weights is determined, if it is > 0.2, then the weights are not consistent, and a user should start the process of comparison from the beginning or allow the default values of the weights to be set.

8. The decision matrix with $m \times n$ dimension is determined, where m is the number of methodologies, n is the number of criteria, the values of which are set by a user. The matrix consists of evaluation of the established values of criteria in relation to methodologies.

9. The scores for methodologies are determined using the weighted sum method.

10. The scores for the methodology are determined using the TOPSIS method:

a) A weighted normalised matrix is determined.

b) The positive and negative ideal solution is determined.

c) The Euclidean distance and relative proximity of each of the alternatives (methodologies) to ideal solutions are calculated.

11. The methodologies are sorted from the best (with the highest scores) to the worst (with the lowest scores), and details on the scores of the established values of criteria are provided.

For the purpose of the software implementation of the above algorithm, a software system was developed in the form of a web application with a client-server architecture, therefore, any modern web browser with the Internet access can be its operating environment. For technical implementation, the Ruby programming language version 2.6.5 was chosen with the Ruby on Rails framework version 6.0.3.3. PostgreSQL version 13.1 was used as a DBMS.

The main features of the software product are the introduction by experts of membership functions for each known criterion regarding each methodology in the system; adding new methodologies and criteria; determining the criteria weights by default; determining the criteria weights based on comparison of criteria by a user; input of criteria values by a user; selection and output of the results of the methodology selection. The form for creating a new project is presented in Fig. 2.

DOI 10.15588/1607-3274-2022-2-17

By clicking on a specific project, a user will be redirected to the stage corresponding to the status of the project. This can be:

– filling out a questionnaire about the project (Fig. 3);

– comparison of the importance of parameters (Fig. 4);

– page with results (Fig. 5).

The questionnaire for setting the parameter values is displayed as shown in Fig. 3. The name of the project is indicated at the top of the page, below it there is a progress bar displaying the percentage of questions (parameters) answered by a user, below it there is the name of a parameter and available answer choices, as well as the submit and skip buttons.

Figure 4 shows the interface for the pairwise comparison of parameters. It contains the names of parameters and their set values, as well as a slider to estimate the degree of importance of one parameter with respect to the other one.

Figure 5 shows the results page. The left pane displays a list of methodologies, sorted from best to worst. After clicking on one of them, the right panel displays their values and scores of the set parameter values in relation to this methodology.

The questionnaire for establishing expert evaluation is shown in Fig. 6.

The developed algorithm for selecting a software development methodology uses the weighted sum and TOPSIS methods to find the best alternative, i.e., methodology. To determine the weights of criteria by a user, the AHP method is used. The decision matrix is formed of the estimates of the criteria values in relation to the methodologies determined with the help of experts.

To check the accuracy of the recommendations provided by the software tool, the extent to which it meets the expectations of users - managers and project developers, and its reaction to data changes, the test data presented in the work [8] were used, namely: 7 methodologies, a list of 23 criteria and their possible values, criteria weights, evaluation of criteria values in relation to methodologies.

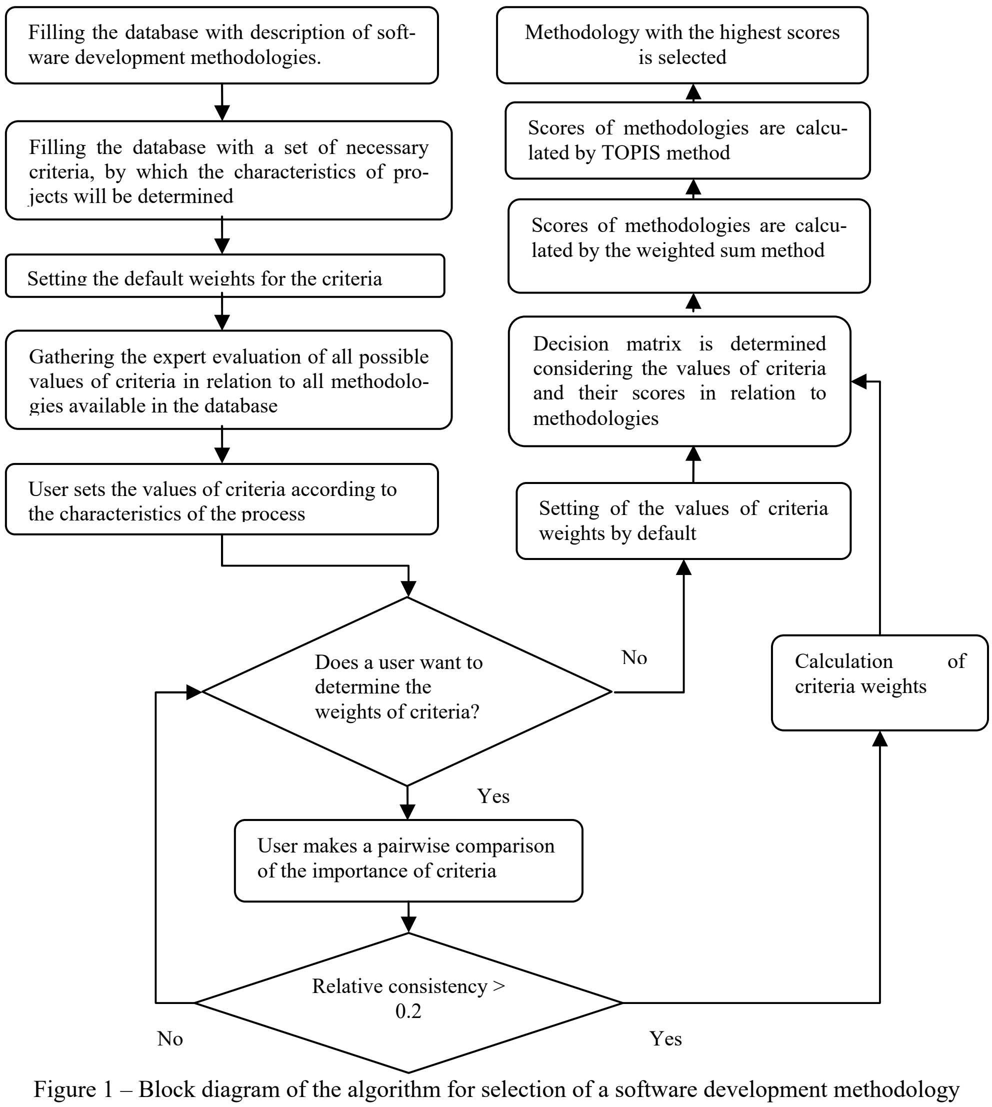

Figure 1 – Block diagram of the algorithm for selection of a software development methodology

DOI 10.15588/1607-3274-2022-2-17

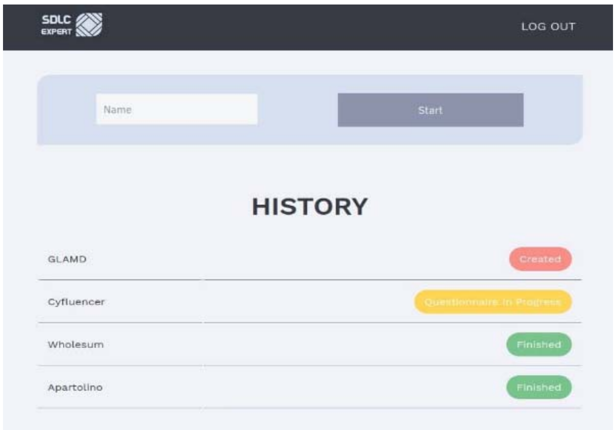

Figure 2 – List of projects

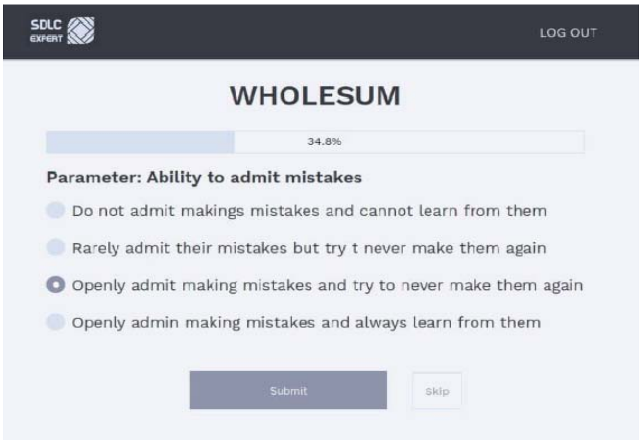

Figure 3 – Setting of parameter values

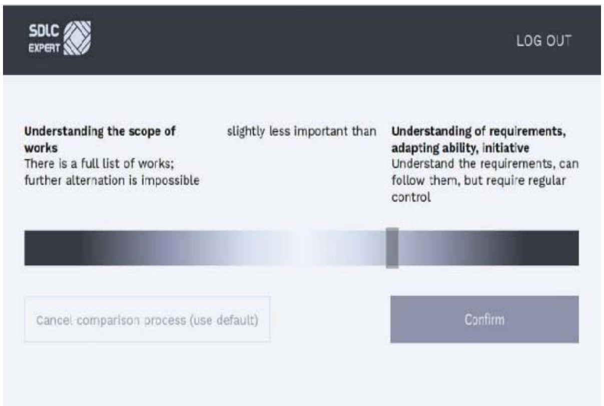

Figure 4 – Comparison of parameter importance

Figure 5 – Page of the methodology selection results

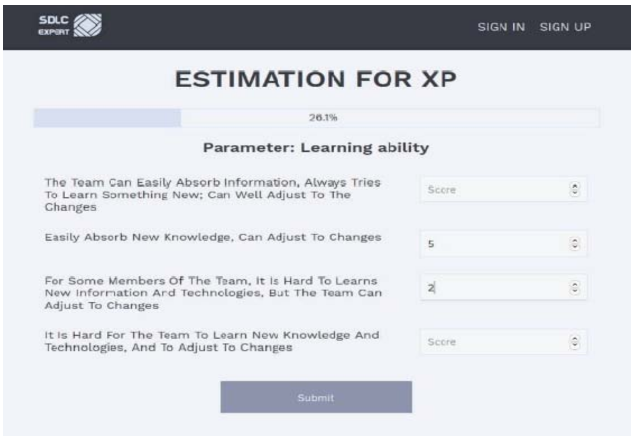

Figure 6 – Filling in the scores by an expert

Table 2 shows the results of calculating the scores using the approach described in [8] and a tool developed by us using the weighted sum and TOPSIS methods.

Table – 2 Table of comparison of methodology scores

Methodology	Results in the work [9]	A developed software tool		
		Weighted Sum	TOPSIS	Average
PMBOK	0.341	0.165	0.347	0.256
ISO21500	0.341	0.165	0.347	0.256
PRINCE2	0.276	0.143	0.314	0.228
SWEBOK	0.361	0.193	0.376	0.285
Scrum	0.900	0.371	0.764	0.567
XP	0.732	0.190	0.404	0.297
Kanban	0.663	0.233	0.514	0.373

The comparison table shows that the recommended methodology is the same in all cases. For clarity, this data is presented in Fig. 7–9 by means of diagrams.

It is also worth comparing the order of ranked methodologies (Table 3).

Table.3 Comparison of the order of ranked methodologies

No.	From the work [9]	A developed software tool		
		Weighted Sum	TOPSIS	Average
1	Scrum	Scrum	Scrum	Scrum
2	XP	Kanban	Kanban	Kanban
3	Kanban	SWEBOK	XP	XP
4	SWEBOK	XP	SWEBOK	SWEBOK
5	PMBOK	PMBOK	PMBOK	PMBOK
6	ISO21500	ISO21500	ISO21500	ISO21500
7	PRINCE2	PRINCE2	PRINCE2	PRINCE2

DOI 10.15588/1607-3274-2022-2-17

It can be seen from the comparison that the methodology recommended by both approaches is the same, but the following two positions differ: in the work [8], the second position is occupied by XP, and the third – by Kanban; in the result of the selection made by our system, on the contrary: Kanban – ranks second and XP ranks third. We can conclude from this research that the system works correctly regarding the results of the work [9].

Also, to verify the operation of the system, its operation was tested using the data of real projects, three anonymized commercial projects from LinkUp company (https://linkupst.com/).

Project No. 1. Web platform for planning meals for groups of people. Main characteristics of the project:

– no experience of work with the customer;
– domain knowledge;
– the team has already worked, having the same composition, with the same tools;
– clear and almost completely known requirements;
– project is not very urgent and does not require strict adherence to deadlines;
– existing risks associated with third-party service;
– communication in the form of correspondence;
– reporting after the implementation of individual components of the product.

Criteria weights: By default. Expected result: Scrum. Results – (Table 4)

Table 4 – The result of selection of methodology for the project No. 1

№	Weighted Sum		TOPSIS		Average	
1.	SWEBOK	0.2913	Scrum	0.5526	Scrum	0.4106
2.	ISO21500	0.2696	SWEBOK	0.5204	SWEBOK	0.4059
3.	PMBOK	0.2696	PMBOK	0.4894	PMBOK	0.3795
4.	Scrum	0.2685	ISO21500	0.4894	ISO21500	0.3795
5.	PRINCE2	0.2619	Kanban	0.4885	PRINCE2	0.3723
6.	Kanban	0.2141	PRINCE2	0.4826	Kanban	0.3513
7.	XP	0.2059	XP	0.4633	XP	0.3346

Project No. 2. Web-based rental platform. Main characteristics of the project:

– no experience of work with the customer;
– minimum domain knowledge;
– a large team;
– requirements are known in large part;
– urgent;
– expensive;
– no significant risks;
– weekly calls;
– reporting every two weeks.

Criteria weights: By default. Expected result: SWEBOK. Results – (Table 5)

Table 5 – The result of selection of methodology for the project No. 2

№	Weighted Sum		TOPSIS	
1.	SWEBOK	0.3565	SWEBOK	0.6810
2.	PRINCE2	0.3424	PRINCE2	0.6612
3.	ISO21500	0.3304	ISO21500	0.6251
4.	PMBOK	0.3304	PMBOK	0.4347
5.	Scrum	0.2087	Scrum	0.4346
6.	Kanban	0.1467	Kanban	0.2962
7.	XP	0.1441	XP	0.2900

Project No. 3. Mobile game. Main characteristics of the project:

– customer's experience of work with the team;
– good domain knowledge;
– a small team consisting of the developers who have already worked together on games;
– most requirements are known;
– not very urgent, but adherence to deadlines is required;
– no significant risks;
– communication in the form of correspondence and weekly calls;
– reporting every week.

Criteria weights: By default. Expected result: Kanban. Results – (Table 6).

Table 6 – The result of selection of methodology for the project No. 3

№	Weighted Sum		TOPSIS	
1.	Scrum	0.3185	Scrum	0.6837
2.	Kanban	0.2543	Kanban	0.5067
3.	SWEBOK	0.2489	SWEBOK	0.3974
4.	PRINCE2	0.2250	XP	0.3935
5.	XP	0.2250	PRINCE2	0.3697
6.	PMBOK	0.2228	PMBOK	0.3639
7.	ISO21500	0.2228	ISO21500	0.3639

7 DISCUSSION

Thus, for the first project, the expected result showed only the selection by means of the TOPSIS method, whereas the weighted sum method produced fundamentally different results. This can be explained by the fact that some criteria compensate for the others in the weighted sum method.

As for the second project, users obtained the expected result. However, on the page with the results, users can see that SWEBOK may not meet some of their requirements (Fig. 10). It can be seen that SWEBOK is a bad option for urgent projects, and it does not require the team to be able to quickly learn new things. The users should consider these details when making the final decision on the selection of a software development methodology.

In the third case, a person who was making decisions expected that the recommended methodology would be Kanban, but in the selection with TOPSIS and weighted sum methods, the scrum methodology ranked first. In this case, a user can check why this happened, what values of criteria and to what extent satisfy the methodology by Kanban (Fig. 11).

Thus, a user sees from the results that the following criteria were unsatisfactory for Kanban:

– ability of the team to work without control – Kanban requires the team to work independently and be self-organized without the need of being monitored;
– reporting frequency – Kanban provides for the reporting to be carried out at the end of the project or a large part of the project, but not after every operation;
– understanding of the scope of work – it makes sense to use Kanban if there is a lot of uncertainty about how to implement the product;

DOI 10.15588/1607-3274-2022-2-17

– ability to learn – Kanban is usually used in cases when all team members are able to quickly learn new things;

– adherence to deadlines – Kanban is used in cases when it is not required to strictly adhere to the deadlines, including the intermediate ones;

– frequency of requirements changes – Kanban is an effective solution in cases when frequent changes in requirements are expected. If during the project the requirements remain mostly unchanged, then one of the main advantages of Kanban will not be demonstrated.

The system was tested for the same project but with different criteria weights (Table 7).

In this case, most of the criteria were suitable for the Scrum methodology, therefore, irrespective of the way the criteria weights were arranged, in all cases the Scrum methodology ranked first. The XP methodology was the least suitable in all cases. The weights of criteria influenced all other positions in the ranked list of methodologies.

The critical characteristics for a respective methodology were also determined for each of the projects (Table 8).

Thus, the results of the verification allow us to ensure that in more than 50% of cases the expectations matched the results, namely: for the first project the results met the expectations, for the second – the expected methodology took the second place, for the third one – the expected methodology of the project was recommended by the selection using the TOPSIS method, but not the Weighted Sum – this is justified by the fact that the Weighted Sum method is characterized by compensation between the criteria, therefore we can draw a conclusion, that the results calculated by means of TOPSIS method provide more adequate recommendations. Besides that, the experiment was held, which identified the same values of the criteria, but different weights, and which revealed that the system responds to such changes, but if a certain methodology has a very large advantage over others, the weights do not have much effect on the "victory" of this methodology.

An experiment was also conducted with the change of weights of the criteria while their values remained unchanged, the result of which suggests that the weights of the criteria significantly affect the selection result, especially when the values of the criteria satisfy and do not satisfy each of the methodologies almost equally.

CONCLUSIONS

This paper solves the problem of developing an effective and flexible tool for selecting of the most appropriate methodology for software development considering the characteristics of the project. To solve this problem the analysis of the existing approaches to the selection of software development methodology was carried out, as a reslt of which it was determined that most of these approaches are focused on the selection of a certain methodology out of the fixed set, and they consider a limited range of criteria. We have also developed the algorithm and software system for the selection of the best methodology of software development depending on the characteristics of the project, where the criteria weights provided by the experts were considered and the AHP method was applied to determine user priorities for methodology comparison criteria. TOPSIS and weighted sum methods were chosen to calculate the scores of methodology choice. The software tool provides for the output of useful details about the selection results, namely, an expert evaluation of the set parameter values regarding all methodologies. The verification of the developed software system was performed based on the test data of the paper [8], which showed almost an exact match of recommendations of the best methodologies for this project and on the real projects by the comparison of expected results of the user with the results the user received with the help of the developed software tool. The results of the verification were the following: more than in 50% of cases, the expectation matched the results.

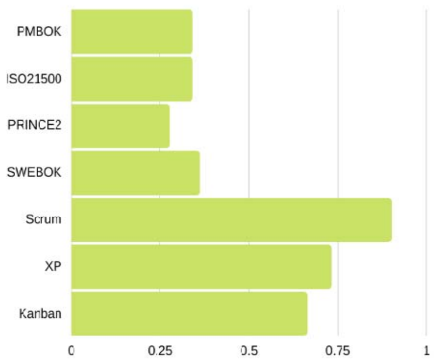

Figure 7 – Scores of methodologies in the work [8]

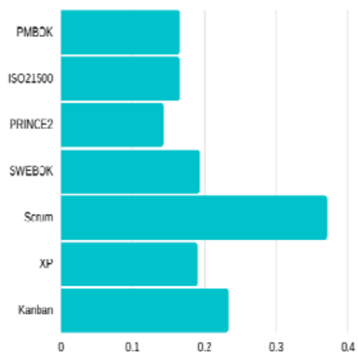

Figure 8 – Scores of methodologies using the weighted sum method

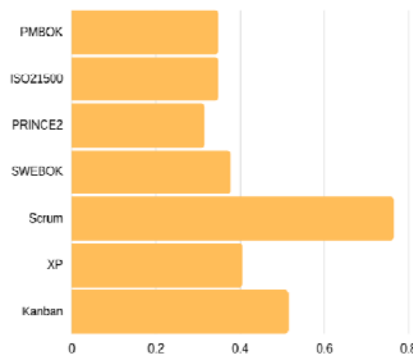

Figure 9 – Scores of methodologies using the TOPSIS method

DOI 10.15588/1607-3274-2022-2-17

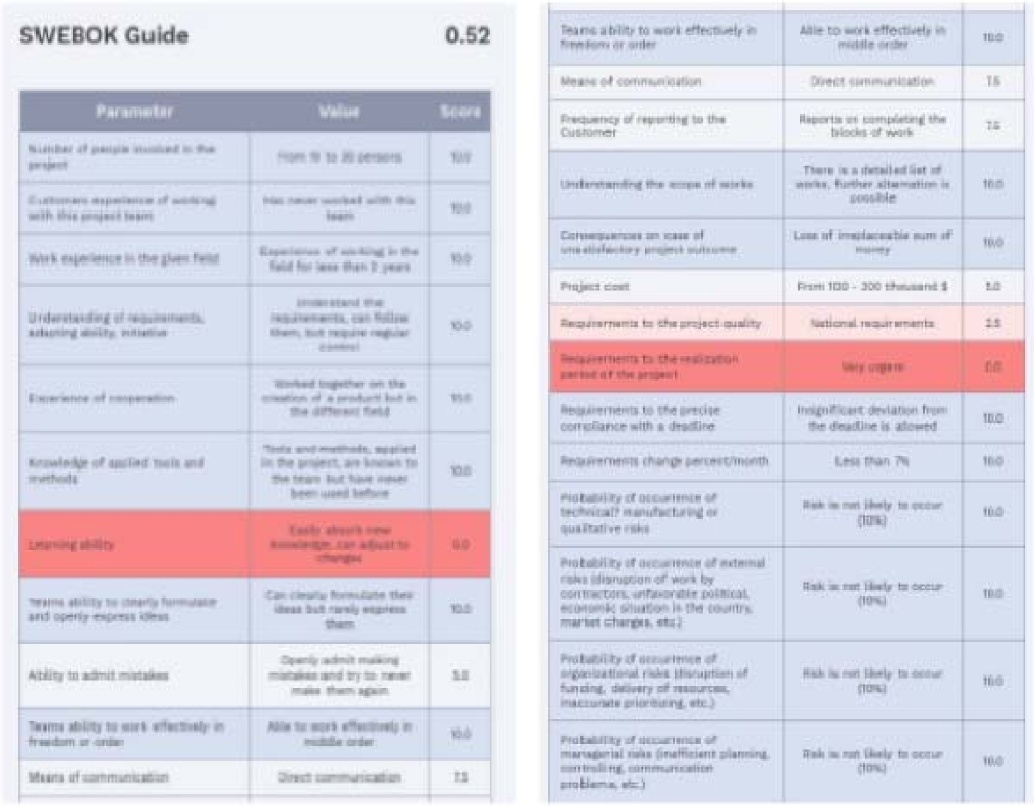

Figure 10 – Details of the scores of SWEBOK methodology for the project No. 2

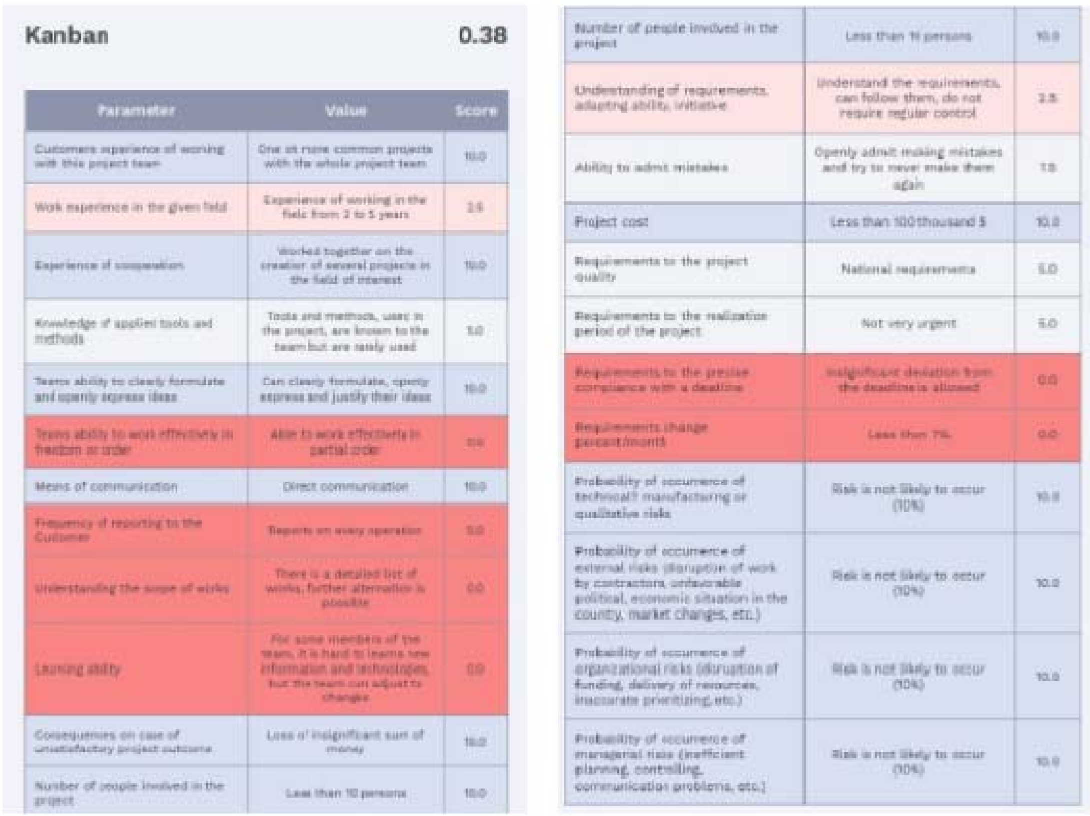

Figure 11 – Details of the scores of Kanban methodology for the project No. 3

DOI 10.15588/1607-3274-2022-2-17

Table 7 – Comparison of results with different versions of criterion weights

No.	Weights by default		Same weight (= 1)		User weights	
1	Scrum	0.5782	Scrum	0.6018	Scrum	0.5562
2	Kanban	0.5193	SWEBOK	0.4838	PMBOK	0.5167
3	SWEBOK	0.4796	Kanban	0.4824	ISO21500	0.5167
4	PRINCE2	0.4562	PMBOK	0.4771	SWEBOK	0.5154
5	PMBOK	0.4458	ISO21500	0.4771	PRINCE2	0.4874
6	ISO21500	0.44588	PRINCE2	0.4536	Kanban	0.4302
7	XP	0.3843	XP	0.4126	XP	0.3880

Table 8 – The critical characteristics for a respective methodology for each of the projects

Project No. 1 (Scrum)	**Project No. 2 (SWEBOK)**	**Project No. 3 (Kanban)**
Requirements change percent/month	Requirements change percent/month	Teams ability to work effectively in freedom or order
Work experience in the given field	Teams ability to work effectively in freedom or order	Frequency of reporting to the Customer
Understanding of the scope of works	Understanding of requirements, adapting ability, initiative	Understanding of the scope of works
Experience of cooperation	Frequency of reporting to the Customer	Learning ability
Customers experience of working with this project team	Customers experience of working with this project team	Requirements to the precise compliance with a deadline
		Requirements change percent/month

The **scientific novelty** of the received results lies in the fact that it is for the first time when the algorithm for the selecting a methodology of software development was designed, and unlike other existing algorithms this one provides for collecting of expert evaluation, yet considering the values of criteria, specified by the user independently, and implements the process of selecting methodologies using the methods of multi-criteria analysis AHP, TOPSIS and Weighted Sum.

Practical value of the results of this paper lies in the fact, that we suggested an approach, which helps software engineers to choose a methodology of software development, which meets their requirements and expectations. The approach is based on the developed algorithm, which uses 7 methodologies and 23 criteria of the projects and provides for collecting of expert evaluation as well as implements the process of selecting methodologies my means of methods of multi-criteria analysis AHP and Weighted Sum and allows us to determine "unsatisfactory" criteria for a particular methodology. The practical application of the suggested approach allows us to reduce the time spent on the process of selecting the methodology for a particular project by 2–3% of the total project cost, as well as increase the adequacy of the methodology selection, especially for the teams with little experience, which generally confirms the feasibility of its use when choosing the best software development methodology.

The developed software tool for multi-criteria decision-making regarding the selection of development methodology, depending on the characteristics of the project and organization of business processes in an IT company, allows for systematic investigating the problem of the methodology selection, prioritization and evaluation of the goals and alternatives of the selection.

REFERENCES

1. Sommerville, Ian. Software engineering. Boston, Pearson, 2011, 773 p.
2. V. B. S. Silva, F. Schramm and A. C. Damasceno, A multicriteria approach for selection of agile methodologies in software development projects, *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC).* Budapest, Hungary, 2016, pp. 2056–2060. DOI: 10.1109/SMC.2016.7844542.
3. Preeti Rai Impact of Different Methodologies in Software Development Process, (IJCSIT), *International Journal of Computer Science and Information Technologies*, 2014, Vol. 5 (2), pp. 1112–1116.
4. Harb Y. A., Noteboom C. Evaluating Project Characteristics for Selecting the Best-fit Agile Software Development Methodology: A Teaching Case Evaluating Project Characteristics for Selecting the Bestfit, *Journal of the Midwest Association for Information Systems*, 2015, Vol. 1 (1), pp. 33–52.
5. Kumar K., Kumar S. A Rule-based Recommendation System for Selection of Software Development Life Cycle Models, *ACM SIGSOFT Software Engineering Notes*, 2013, Vol. 38, Issue 4, pp. 1–6. DOI: 10.1145 / 2492248.2492269
6. Gil Hernández-Ledesma, Erik G. Ramos, Carlos A. Fernández-y-Fernández et al. Selection of Best Software Engineering Practices: A Multi-Criteria Decision Making Approach, *Research in Computing Science*, 2017, No. 136, pp. 47–60. DOI: 10.13053/rcs-136-1-4
7. Victor M., Upadhyay N. Selection of Software Testing Technique: A Multi Criteria Decision Making Approach, *Trends in Computer Science, Engineering and Information Technology. CCSEIT 2011. Communications in Computer and Information Science, Goa-403726, India 2011, proceedings.* Springer, Berlin, Heidelberg, Vol. 204, pp. 453–462 https://doi.org/10.1007/978-3-642-24043-0_46
8. Kononenko I. V., Lutsenko S. Yu. Method for selection of project management approach based on fuzzy concepts, *Bulletin of NTU "KhPI". Series: Strategic management, portfolio, program and project management*, 2017, Vol. 2 (1224). pp. 8–17. DOI: https://doi.org/10.20998/2413-3000.2017.1224.2
9. Despa M. L. Comparative study on software development methodologies, *Database Systems Journal*, 2014, Vol. V, № 3/2014, pp. 37–56.

DOI 10.15588/1607-3274-2022-2-17

10. Matharu G. S., Mishra A., Singh H. et al. Empirical Study of Agile Software Development Methodologies: A Comparative Analysis, *ACM SIGSOFT Software Engineering Notes,* 2015, Vol. 40, No. 1, pp. 1–6. https://doi.org/10.1145/2693208.2693233
11. Vijayasarathy L. R., Butler C. W. Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter? *IEEE Software*, 2016, Vol. 33, No. 5, pp. 86–94. DOI: 10.1109/MS.2015.26.
12. Gonzalez-Perez C., Henderson-Sellers B. Modelling software development methodologies: A conceptual foundation, *Journal of Systems and Software*, 2007, Vol. 80, No. 11, pp. 1778–1796. https://doi.org/10.1016/j.jss.2007.02.048
13. Saeed S., Jhanjhi NZ, Naqvi M. et al. Analysis of Software Development Methodologies, *International Journal of Computing and Digital Systems*, 2019, Vol. 8, No. 5, pp. 445–460. http://dx.doi.org/10.12785/ijcds/080502
14. Saeedi K., Visvizi A. Software Development Methodologies, HEIs, and the Digital Economy, *Education Sciences,* 2021, Vol. 11, No. 2, P. 73. https://doi.org/10.3390/educsci11020073
15. Symets I., Yakovyna V., Seniv M. The Relation between Software Development Methodologies and Factors Affecting Software Reliability, *Computer Sciences and Information Technologies (CSIT), 15th IEEE International Conference*. Lviv-Zbarazh, 23–26 September 2020, pp. 377–381. doi: 10.1109/CSIT49958.2020.9321937
16. Saaty T., Greco S., Ehrgott M., Figueira J. (eds). The analytic hierarchy and analytic network processes for the measurement of intangible criteria and for decision-making, *Multiple Criteria Decision Analysis. International Series in Operations Research & Management Science*. Springer, NY 2016, Vol. 233, pp. 363–419.

Received 30.01.2022.
Accepted 16.02.2022.

УДК 004.05; 004.4

ЗАСОБИ ПІДБОРУ МЕТОДОЛОГІЇ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З УРАХУВАННЯМ ХАРАКТЕРИСТИК ПРОЕКТУ

Сенів М. М. – канд. техн. наук, доцент, доцент кафедри Програмного забезпечення, Національний університет «Львівська політехніка», Львів, Україна.

Ковтонюк А. М. – магістр інженерії програмного забезпечення, Національний університет «Львівська політехніка», Львів, Україна.

Яковина В. С. – д-р техн. наук, професор, професор кафедри Систем штучного інтелекту, Національний університет «Львівська політехніка», Львів, Україна; Факультет математики та комп'ютерних наук, Вармінсько-Мазурський Університет в Ольштині, Польща.

АНОТАЦІЯ

Актуальність. В процесі розробки програмного забезпечення вибір методології його розроблення є одним з важливих етапів, який суттєво впливає на успіх/провал проекту. Вибір оптимальної методології розробки залежить від багатьох факторів та є трудомісткою і нетривіальною задачею.

Мета. Відповідно, існує потреба у розробленні ефективного та гнучкого програмного засобу для підбору оптимальної методології розроблення програмного забезпечення, який би автоматизував даний процес а також враховував ключові характеристики проєкту.

Метод. В даній роботі представлено алгоритм підбору методології розроблення програмного забезпечення з використанням методів багатокритеріального аналізу та експертних оцінок, який передбачає збір оцінок експертів та реалізує процес підбору методології за допомогою методів AHP, TOPSIS та Weighted Sum.

Результати. З використанням вищезазначеного алгоритму було розроблено програмну систему для підбору оптимальної методології розроблення програмного забезпечення в залежності від характеристик проєкту, де враховано ваги критеріїв, надані експертами, а також застосовано метод AHP для визначення користувацьких пріоритетів критеріїв порівняння методологій. Для обчислення оцінок вибору методологій було обрано метод зваженої суми та TOPSIS. Програмний засіб передбачає виведення корисних деталей про результати підбору, а саме експертну оцінку заданих значень параметрів відносно всіх методологій, та може бути використаний для підвищення ефективності процесу розроблення програмного забезпечення в частині автоматизації надання рекомендацій керівникам IT-проектів.

Висновки. Розроблено алгоритм для вибору методології розроблення програмного забезпечення, який, на відміну від існуючих, передбачає збір оцінок експертів, враховуючи при цьому значення критеріїв, заданих користувачем самостійно, і реалізує процес підбору методологій використовуючи методи багатокритеріального аналізу AHP, TOPSIS та Weighted Sum. З використанням вищеописаного алгоритму було розроблено програмну систему для підбору оптимальної методології розроблення програмного забезпечення в залежності від характеристик проєкту, де враховано ваги критеріїв, надані експертами, а також застосовано метод AHP для визначення користувацьких пріоритетів критеріїв порівняння методологій. Для обчислення оцінок вибору методологій було обрано метод зваженої суми та TOPSIS. Програмний засіб передбачає виведення корисних деталей про результати підбору, а саме експертну оцінку заданих значень параметрів відносно всіх методологій.

КЛЮЧОВІ СЛОВА: програмне забезпечення; методології розробки програмного забезпечення; інженерія програмного забезпечення.

УДК 004.05; 004.4

СРЕДСТВА ПОДБОРА МЕТОДОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С УЧЕТОМ ХАРАКТЕРИСТИК ПРОЕКТА

Сенив М. М. – канд. техн. наук, доцент, доцент кафедры Программного обеспечения, Национальный университет «Львовская политехника», Львов, Украина.

Ковтонюк А. М. – магистр инженерии программного обеспечения, Национальный университет «Львовская политехника», Львов, Украина.

Яковина В. С. – д-р техн. наук, профессор, профессор кафедры Систем искусственного интеллекта, Национальный университет «Львовская политехника», Львов, Украина; Факультет математики и компьютерных наук, Варминско-Мазурский Университет в Ольштыне, Польша.

DOI 10.15588/1607-3274-2022-2-17

АННОТАЦИЯ

Актуальность. В процессе разработки программного обеспечения выбор методологии его разработки является одним из важных этапов, существенно влияющих на успех/провал проекта. Выбор оптимальной методологии разработки зависит от многих факторов и является трудоемкой и нетривиальной задачей.

Цель. Соответственно, существует потребность в разработке эффективного и гибкого программного средства для подбора оптимальной методологии разработки программного обеспечения, которое бы автоматизировало данный процесс а также учитывало ключевые характеристики проекта.

Метод. В данной работе представлен алгоритм подбора методологии разработки программного обеспечения с использованием методов многокри-териального анализа и экспертных оценок, предусматривающий сбор оценок экспертов и реализующий процесс подбора методологии с помощью методов AHP, TOPSIS и Weighted Sum.

Результаты. С использованием вышеупомянутого алгоритма была разработана программная система для подбора оптимальной методологии разработки программного обеспечения в зависимости от характеристик проекта, где учтены весы критериев, предоставленные экспертами, а также применен метод AHP для определения пользовательских приоритетов критериев сравнения методологий. Для вычисления оценок выбора методологии был выбран метод взвешенной суммы и TOPSIS. Программное средство предполагает вывод полезных деталей о результатах подбора, а именно экспертную оценку заданных значений параметров относительно всех методологий, и может быть использован для повышения эффективности процесса разработки программного обеспечения в части автоматизации предоставления рекомендаций руководителям ИТ-проектов.

Выводы Разработан алгоритм выбора методологии разработки проограммного обеспечения, который, в отличие от существующих, предусматривает сбор оценок экспертов, учитывая при этом значение критериев, заданных пользователем самостоятельно, и реализует процесс подбора методологий используя методы многокритериального анализа AHP, TOPSIS и Weighted Sum. С использованием вышеописанного алгоритма была разработана программная система для подбора оптимальной методологии разработки программного обеспечения в зависимости от характеристик проекта, где учтены весы критериев, предоставленные экспертами, а также применен метод AHP для определения пользовательских приоритетов критериев сравнения методологий. Для вычисления оценок выбора методологии был выбран метод взвешенной суммы и TOPSIS. Программное средство предусматривает вывод полезных деталей о результатах подбора, а именно экспертную оценку заданных значений параметров в отношении всех методологий.

КЛЮЧЕВЫЕ СЛОВА: программное обеспечение; методологии разработки программного обеспечения; инженерия программного обеспечения.

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Sommerville, Ian. Software engineering. / I. Sommerville. – Boston : Pearson, 2011. – 773 p.
2. Silva V. B. S. A multicriteria approach for selection of agile methodologies in software development projects / V. B. S. Silva, F. Schramm and A. C. Damasceno //2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). – Budapest, Hungary. – 2016. – P. 2056–2060. DOI: 10.1109/SMC.2016.7844542.
3. Preeti Rai Impact of Different Methodologies in Software Development Process, (IJCSIT) / R. Preeti// International Journal of Computer Science and Information Technologies. – 2014. – Vol. 5 (2). – P. 1112–1116.
4. Harb Y.A. Evaluating Project Characteristics for Selecting the Best-fit Agile Software Development Methodology: A Teaching Case Evaluating Project Characteristics for Selecting the Bestfit / Y. A. Harb., C. Noteboom // Journal of the Midwest Association for Information Systems. – 2015. – Vol. 1 (1). – P. 33–52.
5. Kumar K. A Rule-based Recommendation System for Selection of Software Development Life Cycle Models / K. Kumar, S. Kumar // ACM SIGSOFT Software Engineering Notes. – 2013. – Vol. 38, Issue 4. – P. 1–6. DOI: 10.1145 / 2492248.2492269
6. Selection of Best Software Engineering Practices: A Multi-Criteria Decision Making Approach / Gil Hernández-Ledesma, Erik G. Ramos, Carlos A. Fernández-y-Fernández et al. // Research in Computing Science. – 2017. – № 136. – P. 47–60. DOI: 10.13053/rcs-136-1-4
7. Victor M. Selection of Software Testing Technique: A Multi Criteria Decision Making Approach / M. Victor, N. Upadhyay // Trends in Computer Science, Engineering and Information Technology. CCSEIT 2011. Communications in Computer and Information Science, Goa-403726, India 2011: proceedings. – Springer, Berlin, Heidelberg. – Vol. 204. – P. 453–462. https://doi.org/10.1007/978-3-642-24043-0_46
8. Kononenko I. V. Method for selection of project management approach based on fuzzy concepts / I. V. Kononenko S. Yu. Lutsenko // Bulletin of NTU “KhPI”. Series: Strategic management, portfolio, program and project management. – 2017. – Vol. 2 (1224). – P. 8–17. DOI: https://doi.org/10.20998/2413-3000.2017.1224.2
9. Despa M. L. Comparative study on software development methodologies / M. L. Despa // Database Systems Journal. – 2014. – Vol. V, № 3/2014. – P. 37–56.
10. Empirical Study of Agile Software Development Methodologies: A Comparative Analysis / G. S. Matharu. A. Mishra, H. Singh et al. // ACM SIGSOFT Software Engineering Notes. – 2015. – Vol. 40, № 1. – P. 1–6. https://doi.org/10.1145/2693208.2693233
11. Vijayasarathy L.R. Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter? / L. R. Vijayasarathy. C. W. Butler // IEEE Software. – 2016. – Vol. 33, № 5. – P. 86–94. DOI: 10.1109/MS.2015.26.
12. Gonzalez-Perez C. Modelling software development methodologies: A conceptual foundation / C. Gonzalez-Perez, B. Henderson-Sellers // Journal of Systems and Software. – 2007. – Vol. 80, № 11, – P. 1778–1796. https://doi.org/10.1016/j.jss.2007.02.048
13. Analysis of Software Development Methodologies / S. Saeed, NZ Jhanjhi, M. Naqvi et al. // International Journal of Computing and Digital Systems. – 2019. – Vol. 8, № 5. – P. 445–460. http://dx.doi.org/10.12785/ijcds/080502
14. Saeedi K. Software Development Methodologies, HEIs, and the Digital Economy / K. Saeedi, A. Visvizi. // Education Sciences. – 2021. – Vol. 11, № 2. – P. 73. https://doi.org/10.3390/educsci11020073
15. Symets I. The Relation between Software Development Methodologies and Factors Affecting Software Reliability / I. Symets, V. Yakovyna, M. Seniv // Computer Sciences and Information Technologies (CSIT) : 15th IEEE International Conference, Lviv-Zbarazh, 23–26 September. – 2020. – P. 377–381. DOI: 10.1109/CSIT49958.2020.9321937
16. Saaty T. The analytic hierarchy and analytic network processes for the measurement of intangible criteria and for decision-making / T. Saaty // Multiple Criteria Decision Analysis. International Series in Operations Research & Management Science / Greco S., Ehrgott M., Figueira J. (eds). – Springer, NY. – 2016. – Vol. 233. – P. 363–419.

DOI 10.15588/1607-3274-2022-2-17