

Tetiana HOVORUSHCHENKO, Yurii VOICHUR, Dmytro MEDZATYI

Khmelnytskyi National University, Khmelnytskyi, Ukraine

INFORMATION TECHNOLOGY FOR PREDICTION OF SOFTWARE QUALITY LEVEL

Today, there is a contradiction between the rapid increase in the complexity and size of modern software while increasing responsibility for the performance of their functions, the increasing requirements of customers and users to the quality and efficiency of software use and the imperfection of models, methods, tools of predicting software quality at the early stages of the life cycle. Therefore, the task of predicting the software quality level based on requirements is relevant. The aim of this study is to solve this task by developing information technology for prediction software quality levels based on requirements. The proposed information technology for prediction software quality level based on requirements provides analysis of quality attributes in requirements, reflects the dependence (equations) of quality characteristics on attributes, forms a quantitative assessment of quality characteristics, reflects the dependence (equation) of quality on its characteristics, forms a quantitative assessment of quality, performs quality level prediction, provides all the listed services simultaneously and the model, methods, and tools underlying the IT belong to common methodological approaches and are integrated. The developed system of predicting the software quality level based on requirements provides the user with predicted estimates of eight software quality characteristics, geometric interpretation of the software quality characteristics' values, a comprehensive indicator of the predicted software quality, and a conclusion about the future software quality level. On the basis of this, it is possible to compare sets of requirements for software and make a reasonable choice of a set of requirements for further implementation. The information technology and the system of predicting the software quality level based on requirements, developed in this paper, provide the possibility of comparing sets of requirements for software, justified selection of requirements for further implementation of quality software (as experiments have shown, this is only one of the four proposed sets), and rejection or revision of unsuccessful sets of requirements that cannot be used to develop quality software.

Keywords: software quality; software quality characteristic; software quality attribute; set of software requirements; software quality level.

1. Introduction

1.1. Motivation

The process of growing interest in the IT industry leads to the development of a huge amount of software for various functional purposes, the size and complexity of which are constantly growing [1].

The effective functioning of information technologies is determined by the quality of their components: technical, mathematical, software, information, legal and other. One of the most significant IT components is software, the quality management of which largely determines the quality of information technology [2].

Software quality is the degree of satisfaction the user needs by the software when used in certain conditions [3]. The person most interested in software quality is the end user who uses the software product.

At present, software development has become one of the most expensive industries, and any weaknesses in its production process can lead to unwanted consequences – increased software development time, increased software cost, reduced software performance, inability of

the developed software to perform the required functional tasks, reduced competitiveness and reputation of the developer, inefficient decision-making, etc. [4, 5].

Today, software engineering has a high probability of failing the project [6]. In the general practice of software development, about 10-20% of all software projects do not reach completion, 40-60% of projects are not completed on time (such projects are completed 150-200% late on average), 25-40% of projects do not fully implement their tasks, 40-55% of projects required additional funding, and in 20% of projects not all proposals made by the customer were considered [7]. The insufficient quality of the software created requires up to 70% of the budget on software project to be reserved for the maintenance phase. At the same time, up to 60% of all software modifications are performed to eliminate errors, and only the remaining 40% are used to correct the software within the business process, improve certain indicators or prevent potential problems. For every invested \$1 billion, organizations are losing approximately \$97 million due to the lack of software projects' quality and efficiency [8]. The cost of poor software quality (CPSQ) was \$260 billion in 2020 [9]. The cost

of software quality can be differentiated into the cost of preventing defects, assessing performance, and internal and external failures during software operation [10].

The project triangle includes three main elements: time, budget, and functionality (Fig. 1). When changes are made to at least one of these elements, all the others also changing [11, 12]. The nature of the impact of changes in a element on the others depends mainly on the specific circumstances and project specifics. For example, a reduction in the project duration may, in one case, lead to an increase in its cost because of the need to involve more developers or, in another case, to a decrease in cost because of the refusal to implement certain functionalities. Quality is the fourth element of the project triangle, in its center. Therefore, even a slight change in any of the other elements will necessarily affect quality.

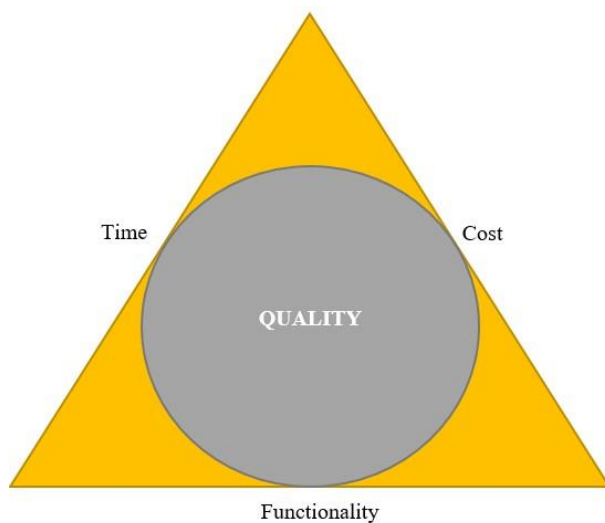


Fig. 1. The project triangle

The complexity and quality of software requirements have changed dramatically in recent years, and consumers have become much more demanding in terms of the cost, time, and quality of software solutions [13, 14]. The rapid growth in the complexity and size of modern software, while increasing the responsibility of the functions it performs, has dramatically increased the requirements of customers and users for the quality and efficiency of software use. Any project's success is its ability to meet customer needs. In this regard, ensuring the required quality of software has become a strategic task in the life cycle of modern software. Therefore, the IT industry trend recently has been to pay increased attention to the topic of software development quality control and management. The basis of quality management is software quality assessment, especially in the early phases of software development (in fact, predicting the quality of future software), since the occurrence frequency and influence percentage of factors related to the early stages of the life cycle is the highest (about one third) [15, 16]. It is generally recognized that high-quality requirements

engineering leads to an increase in software quality and significantly reduces budget overruns or failure risk in software projects [17].

Therefore, the modern software industry requires dynamic development of software quality predicting tools at the life cycle early stages – in order to take preventive measures for reducing the software errors and failures number; in order to select a set of requirements from a set of different alternatives for the same task or problem (when customer orders a software product from various IT companies or its order will be handled by several alternative teams of developers in one IT firm); to understand whether it is possible to develop high-quality software on the basis of the available requirements' set (missing data in requirements for any type of software cause to critical losses of software functionality [18]; software written by different developers using the same requirements usually contain the common errors related to the requirements' inaccuracies or errors [19, 20]; in the same time software versions written by the same developers' team according to different requirements differed significantly in their quality and success [21, 22]).

The software requirements describe the behavior of future software, the functionality and limitations of this software, and its properties and attributes. The values of quality attributes consider the software project type and purpose. If the attribute(s) and its value(s) are included in the software requirements, developers must ensure implementation and the corresponding value of attribute(s) in the software; otherwise, the software will not be verified and validated. Thus, it is possible to determine quality subcharacteristics and characteristics (defined in ISO 25010:2011 [3]) on the basis of the quality attribute values (defined in ISO 25023:2016 [23]), which are specified in the requirements [24, 25], as well as to predict the software quality level for any type of software project [26, 27].

1.2. State-of-the-Art

Let's consider the known models, methods, and tools for analyzing software requirements and assessing software quality. As defined in [28], in terms of the ability to predict the software quality level based on requirements, it is important to determine whether the known models, methods, and tools provide the following criteria:

- C1. Analysis of quality attributes in requirements;
- C2. Dependence (equations) of quality characteristics on attributes;
- C3. Quantification of quality characteristics;
- C4. Dependence (equations) of quality on its characteristics;
- C5. Quantification of quality;
- C6. Prediction of the quality level (as a quantitative and qualitative characterization of the degree of suitability of software to meet a specific demand for it in com-

parison with the corresponding baseline indicators under fixed consumption conditions, i.e. quantitative characterization of the degree of satisfaction the user needs by the software when used in certain conditions);

C7. Whether the known models, methods, and tools provide all of the above services in a complex;

C8. Whether these models, methods, tools refer to the same methodological approaches, as well as whether

they are well integrated with one another (i.e. whether there is presently the information technology for predicting the software quality level based on requirements that provides the above services).

Let's summarize the results of the study of known models, methods, tools for analyzing software requirements and assessing software quality from the point of view of the above criteria satisfaction – Table 1.

Table 1

Findings from the study of known models, methods and tools for analyzing software requirements and assessing software quality

Models, methods, tools for analyzing software requirements and assessment software quality	Author(s)	C1	C2	C3	C4	C5	C6	C7	C8
Methodology for assessing the quality information's sufficiency in software requirements specifications [29, 30]	Hovorushchenko (2018)	+	+		+				
Information technology for quality information veracity assurance [31]	Hovorushchenko (2018)	+	+		+				
Software quality evaluation and prediction methods and tools [32, 33]	Pomorova et al. (2011-2012)			+	+	+	+		
Machine learning-based model for predicting software quality [34]	Goyal et al. (2020)			+		+			
Fuzzy inference system for prediction software quality on the basis of requirements parameters [35]	Masood et al. (2018)	+		+		+			
Adaptive artificial neural network-based hybrid defect prediction model [36]	Arora et al. (2018)			+		+			
Artificial neural network-based model for the optimizing the software reliability and quality [37]	Tomar et al. (2018)	+				+			
Methodology based on extreme learning machine supported multiobjective grey wolf optimization for software quality prediction [38]	Tripathi et al. (2022)					+	+		+
Expert system for prediction of the quality of use in the development's early stages based on artificial neural networks [39]	Alshareet et al. (2018)					+	+		
Fuzzy mathematics-based methods for assessing the software testing quality [40]	Sun et al. (2021)					+	+		
Methods for analyzing the different software quality aspects [41]	Lakra et al. (2021)			+		+			
Models for predicting the software reuse (incl. artificial neural network) [42]	Padhy et al. (2019)						+		
Applying the artificial neural networks for software defects prediction [43]	Khan et al. (2022)	+		+		+			
ANN-based model for software defects prediction [44]	Kaur et al. (2019)	+		+		+			
Three-parent child and genetic evolution-based approach (3PcGE) for software defects prediction [45]	Goyal (2023)			+		+			
Model of software reliability and quality with more neural networks [46]	Kumaresan et al. (2019)					+	+		
Profile-oriented approach to assessing software requirements profile quality, profile-oriented software quality assessment [47, 48]	Gordieiev et al. (2020, 2022)	+				+			

Therefore, methodology for assessing the quality information's sufficiency in software requirements specifications [29, 30], as well as information technology for quality information veracity assurance [31], provide analysis of quality attributes in requirements, dependence of quality characteristics on attributes, and dependence of quality on its characteristics. Software quality evaluation and prediction methods and tools [32, 33] provide quantification of quality characteristics and quality, set dependence (equations) of quality on its characteristics, and predict the quality level. Machine learning-based model for prediction software quality [34], an adaptive artificial neural network-based hybrid defect prediction model [36], methods for analyzing different software quality aspects [41], and a three-parent child and genetic evolution-based approach (3PcGE) for software defect prediction [45] provide quantification of quality characteristics and quality. A fuzzy inference system for prediction software quality on the basis of requirements parameters [35], applying artificial neural networks for software defects prediction [43], and ANN-based model for software defects prediction [44] provide analysis of quality attributes in requirements, quantification of quality characteristics, and quantification of quality. An artificial neural network-based model for optimization software reliability and quality [37] and profile-oriented software quality assessment [47, 48] provide analysis of quality attributes in requirements and quality quantification. Methodology based on extreme learning machine supported multi-objective grey wolf optimization for software quality prediction [38] provides quantification of quality, prediction of the quality level and using common methodological approaches with the possibility of integrating all methods and tools. Expert system for prediction the quality of use in the development's early stages based on artificial neural networks [39], fuzzy mathematics-based methods for assessing the software testing quality [40], and models of software reliability and quality with more neural networks [46] provide quantification of quality and prediction of the quality level. Models for predicting software reuse (incl. artificial neural network) [42] provide only prediction of the quality level.

So, there are currently solutions aimed at analyzing quality attributes in requirements, aimed at establishing the dependence (equations) of quality characteristics on attributes, and quality on quality characteristics, aimed at quantifying quality characteristics and quantifying quality, aimed at predicting the level of quality. Finally, there are even solutions that represent even methodological approaches. But there are no solutions that provide all the above 6 services in the complex. Criterion 7 is not satisfied by any analyzed method/tools. The information technology for predicting the software quality level based on requirements that provides the above services in the complex is nowadays absent. Therefore, a study of a signifi-

cant number of known models, methods, tools in terms of meeting the above criteria has shown that none of the analyzed decisions satisfies all 8 criteria in the complex.

Thus, there is a contradiction between the rapid increase in the complexity and size of modern software while increasing responsibility for the performance of their functions, the increasing requirements of customers and users to the quality and efficiency of software use, on one side, and the imperfection of models, methods, tools of predicting software quality at the life cycle early stages, on other side. Therefore, predicting the level of software quality based on requirements is a *relevant problem* today.

1.3. Objective and approach

The purpose of this study is to solve this task by developing an information technology for predicting the software quality level (as quantitative and qualitative characterization of the degree of satisfaction the user needs by the software when used in certain conditions) based on requirements, which will satisfy all above 8 criteria simultaneously.

The following tasks should be solved for achieving such a purpose: 1) to describe the concept and method of software quality level prediction based on quality attributes; 2) to design and develop the information technology for predicting the software quality level based on requirements; 3) to design and develop the system for predicting the software quality level based on requirements.

Therefore, *the rest of the paper* is structured as follows. Section 2 describes the proposed concept and method of software quality level prediction based on quality attributes. Section 3 is devoted to the development of information technology for prediction software quality levels based on requirements. Section 4 presents the designed and developed system for predicting the software quality level based on requirements. Section 5 provides the results of the implementation of the proposed information technology and system and a discussion of the proposed information technology Section 6 provides the conclusions.

2. Concept and method of software quality level prediction based on quality attributes

To solve the task of predicting the level of software quality based on requirements, it is necessary to first determine 8 software quality characteristics (reliability, usability, performance efficiency, functional suitability, compatibility, portability, security, maintainability), which are determined by ISO 25010:2011 [3], based on 138 software quality attributes, contained in the requirements for software (according to ISO 25023:2016 [23], quality subcharacteristics depend on 203 attributes, but

only on 138 different attributes, the rest of the attributes are repeated, i.e., some attributes affect more than one software quality subcharacteristic and characteristic).

Now ISO standards [3, 23] allow establishing only the fact that some characteristics depend on certain attributes (for example, Functional Suitability depends on Number of functions, Functional implementation completeness, Functional adequacy, Functional implementation coverage, etc.), but do not show the nature of the dependence, the dependence formula (equation), do not reflect the degree of influence of this or that attribute (weight factor) on the characteristic, etc.

To solve such a difficult-to-formalize task, it is necessary to determine the weighting coefficients of quality attributes (which are influenced by the number of subcharacteristics and characteristics that depend on the attribute) and the relationships (equations) of quality attributes within each subcharacteristic/characteristic. Artificial neural networks (ANNs) are the main tool for representing a function of several variables and determining the weighting coefficients and relationships of the initial and result data (Hecht-Nielsen theorem, etc.).

The concept of predicting software quality characteristics based on quality attributes, available in requirements, using artificial neural network (ANN) was developed by the authors in [28] and was as follows: the values of 138 quality attributes defined in ISO 25023:2016 [23] are fed to the ANN inputs; the ANN processes this set of attributes' values, generalizes information, approximates it, establishes dependencies (equations, which provide to determine quantitative characteristics' values based on attribute values) between attributes and subcharacteristics/characteristics of software quality, and, as a result, produces predicted values of 8 software quality characteristics in accordance with ISO 25010:2011 [3] (reliability, usability, performance efficiency, functional suitability, compatibility, portability, security, maintainability) in the range [0;1], where the value "0" indicates the worst (lowest) level of the corresponding characteristic, and the value "1" indicates the best (highest) level of the corresponding characteristic. For these tasks realization, the ANN with multilayer perceptron architecture was used, for training (the training sample from 15180 vectors was used), and for testing (the testing sample from 1518 vectors was used). The development of the ANN architecture, training, and testing of ANN are the results of the authors' previous research and are presented in [28].

In [28], the authors also developed a method for predicting the level of software quality based on quality attributes, which uses ANN to form the predicted estimates of eight software quality characteristics. Further processing of the predicted values of these characteristics makes it possible to predict the quality level of future software. The method for predicting the level of software quality based on quality attributes, developed

by the authors in [28], consists of the following stages:

- processing the values of 138 quality attributes by an artificial neural network, which provides the predicted values of 8 quality characteristics in the range [0; 1];
- geometric interpretation of the obtained values of 8 quality characteristics (octagon);
- if the constructed octagon is not convex (one or more internal angles exceed 180°), then the software quality level predicting is not performed due to the inadmissibility of compensation of the quality characteristics values;
- if the constructed octagon is convex, the comprehensive indicator of the predicted software quality is calculated (the area of the constructed octagon);
- predicting the level of software quality: if the value of the comprehensive indicator belongs to the range [0; 0.2545), then a low level of quality of the software under development is predicted; if the value of the comprehensive indicator belongs to the range [0.2545; 2.0435), then a medium level of quality of the software under development is predicted; if the value of the comprehensive indicator belongs to the range [2.0435; 2.8284], then a high level of quality of the software under development is predicted (the scale (range of values) for the comprehensive indicator of predicted software quality is [0; 2.8284], because 2.8284 is area of the octagon with the maximum possible values of all characteristics).

If there is a set of prognoses, it is possible to evaluate the decisions (prognosis) made using the developed method. To evaluate the made prognoses, we calculate the correctness of the prognoses (the proportion of correctly made prognoses about the level of software quality among all made prognoses) using the following formula:

$$\text{Precision} = \frac{\text{TPL} + \text{TPM} + \text{TPH}}{\text{TPL} + \text{FPL} + \text{TPM} + \text{FPM} + \text{TPH} + \text{FPH}}, \quad (1)$$

where TPL is the correct prognosis of low software quality, TPM is the correct prognosis of medium software quality, TPH is the correct prognosis of high software quality, FPL is the incorrect prognosis of low software quality, FPM is the incorrect prognosis of medium software quality, and FPH is the incorrect prognosis of high software quality.

3. Information technology for prediction of software quality level

The concept underlying the information technology for predicting the software quality level based on requirements (as a set of processes that uses model(s), method(s) and tool(s) for accumulating, processing and

transmitting primary information to obtain new quality information (information product) about the state of an object, process or phenomenon) is predicting the software quality level on the basis of requirements' set.

The principles of designing and operating such an information technology are as follows:

- the principle of automation of information flow processing – the use of automated tools at all stages of information processing;
- the principle of development – modification of functions and composition of information technology without disrupting its operation;
- the principle of efficiency – maximum effect at minimum cost;
- the principle of compatibility – interaction (reliable and uninterrupted operation) of information technology with other technologies that users of the proposed information technology use in their computer;
- the principle of systematicity – basing information technology on a single methodological approach;
- the principle of adaptability to new tasks – modification of information technology in accordance with new tasks;

– the principle of legality – compliance with the requirements of actual standards in the software quality assessment area (in particular, proposed information technology processes the values of 138 quality attributes from ISO 25023:2016 [23] and forms the values of 8 quality characteristics from ISO 25010:2011 [3]);

– the principle of openness of information – ensuring promptness, regularity and re-liability of the processed information;

– the phasing principle – consistent information technology's development.

The functioning of information technology for predicting the software quality level based on requirements is based on two methods: the method of predicting the software quality level based on quality attributes, developed in [28] and briefly described above, as well as the method of semantic analysis of software requirements to find the values of quality attributes, the development of which will be the purpose of the authors' future research.

Therefore, the structure of the formation of information technology for predicting the software quality level based on requirements is represented in Fig. 2.

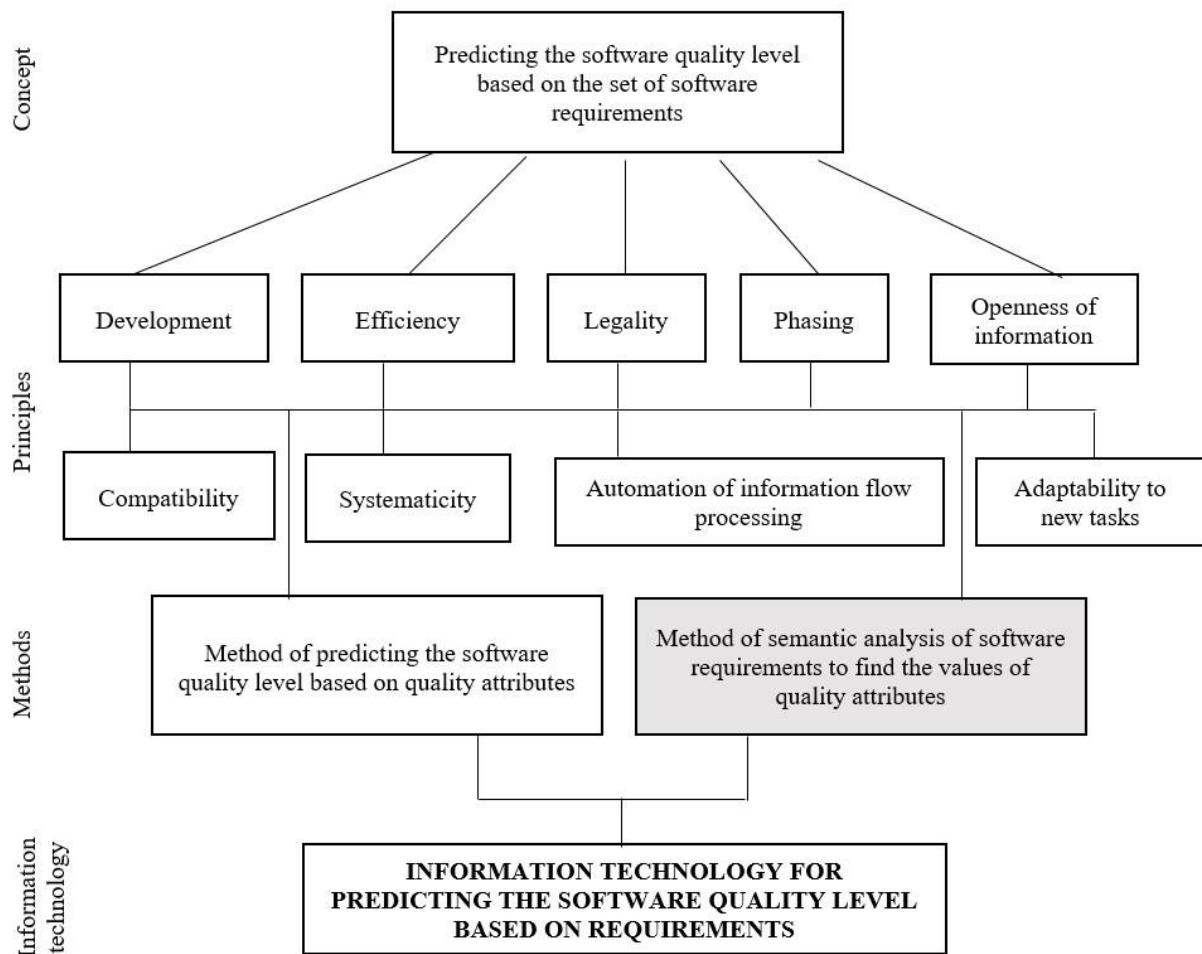


Fig. 2. The structure of formation of information technology for prediction of software quality level

The generalized structure of the information technology for predicting the software quality level based on requirements is shown in Fig. 3.

The detailed structure of the information technology for predicting the software quality level based on requirements is shown in Fig. 4.

Information technology for predicting the software quality level based on requirements				
Object – the process of assessing and predicting software quality based on quality attributes contained in requirements	Goal – predicting the software quality level based on requirements	Model – artificial neural network for predicting the software quality characteristics based on quality attributes	Methods – method for predicting the level of software quality based on quality attributes; method of semantic analysis of software requirements to find the values of quality attributes (will be the purpose of the authors' future research)	Tool – system of predicting the software quality level based on requirements

Fig. 3. The generalized structure of information technology for prediction of software quality level

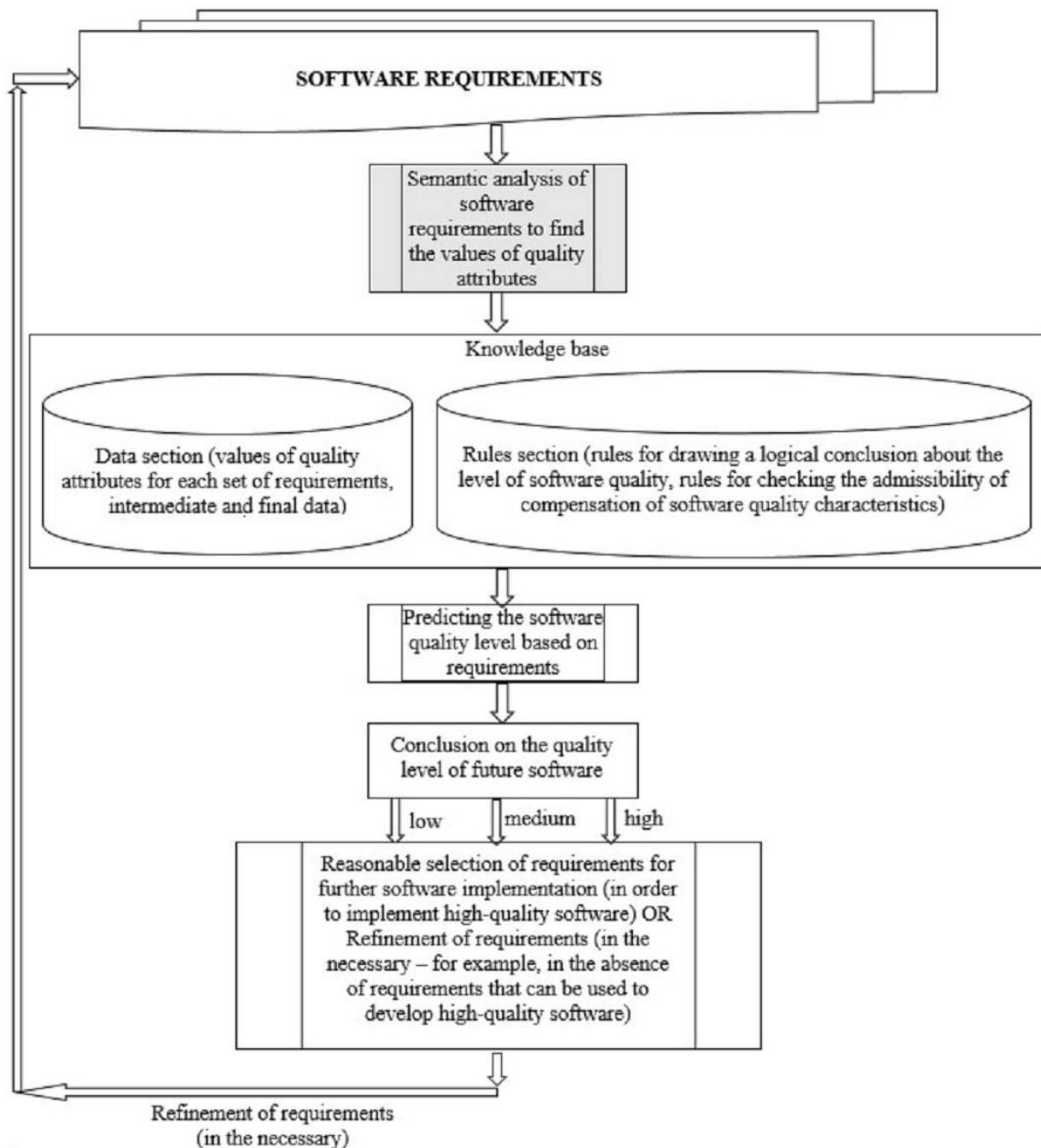


Fig. 4. The detailed structure of the information technology for predicting the software quality level

Thus, the proposed information technology for predicting the software quality level based on requirements functions as follows:

- the information technology is fed with software requirements (for example, some different versions of requirements, for each of which it is necessary to predict the future software quality level);
- the semantic analysis of software requirements is carried out to find the quality attributes' values (this method will be developed by the authors in the future to fully automate the processing of requirements and completely eliminate the subjective influence and human participation in the information handling and knowledge gaining);
- the data section of the knowledge base stores the values of quality attributes for each set of requirements, as well as all intermediate and final data of information technology;
- according to the method of predicting the level of software quality based on quality attributes, which was developed in [28] the artificial neural network generates predicted estimates of eight software quality characteristics. Further processing of these values provides geometric interpretation of these values, verification of the admissibility of software quality characteristics' compensation (according to the rules for checking the admissibility of compensation of software quality characteristics contained in the rules section of the knowledge base), calculation of the predicted software quality comprehensive indicator and formulation of a conclusion about the level of software quality (according to the rules for forming the logical conclusion about the level of software quality contained in the rules section of the knowledge base);
- based on the conclusion about the level of quality of the future software, which will be developed according to a certain set of requirements, a reasonable selection of requirements for further software implementation is made (in order to implement high-quality software), or sets of requirements are sent for revision if there is no set of requirements among the analyzed sets that can provide the high-quality software development;
- after the information technology has developed a set of prognoses, the decisions (prognoses) are evaluated according to formula (1).

The proposed information technology for predicting the software quality level based on requirements satisfies all eight defined criteria simultaneously – provides analysis of quality attributes in requirements, reflects the dependence (equations) of quality characteristics on attributes, forms a quantitative assessment of quality characteristics, reflects the dependence (equation) of quality on its characteristics, forms a quantitative assessment of quality, performs quality level prediction, provides all the listed services simultaneously

and the model, methods, and tools underlying the IT belong to common methodological approaches and are integrated.

Scientific novelty is the first developed information technology for predicting the software quality level based on requirements, which, unlike the known ones, satisfies eight criteria in a complex and provides analysis of quality attributes in requirements, formation of a predicted quantitative assessment of quality, prediction of the quality level of software under development, the ability to compare sets of software requirements, the ability to select requirements for further implementation of high quality software, minimization or complete elimination of subjective influence and human participation in information handling and knowledge gaining.

4. System of prediction of software quality level

According to the generalized structure of information technology for predicting the software quality level based on requirements, presented in Fig. 3, to complete the formation of IT, it is necessary to develop a system of predicting the software quality level based on requirements.

Let's describe the main requirements for a system of predicting the software quality level based on requirements. Such a system should:

- analyze the values of quality attributes in the requirements;
- set of dependence (equations) of quality characteristics on attributes;
- calculate the quantitative values of 8 quality characteristics;
- set of dependence (equation) of quality on its 8 characteristics;
- calculate the quantitative value of the quality;
- predict the quality level of future software;
- provide a prognosis of the quality level of future software in an accessible and understandable form for the user, in natural language;
- have a usable user interface;
- provide the ability to manually enter attribute values from a set of requirements;
- provide the ability to automatically process a set of requirements to find attribute values;
- output to the user predicted estimates of 8 quality characteristics, geometric interpretation of quality characteristics, comprehensive indicator of software quality, and conclusion about the quality level of future software;
- be ground on the method for predicting the level of software quality based on quality attributes developed by the authors as a common methodological approach with the aim of providing the possibility of integration

into the proposed information technology for predicting the software quality level based on requirements.

Fig. 5 shows the architecture of such a system.

The user interface of the system for prediction software quality level based on requirements consists of the following: the module for entering quality attribute values from software requirements, the module for automatic analysis of software requirements to find the values of quality attributes, the user support module, and the results output module. The module for entering quality attribute values from software requirements fo-

cused on manual parsing of software requirements by the user and manual entry of software quality attribute values contained in the requirements set. Module for automatic analysis of software requirements to find quality attributes values will be implemented and operated on the basis of the method of semantic analysis of software requirements to find the values of quality attributes in the authors' further research. The results output module consists of a block for outputting predicted estimates of eight software quality characteristics, and a block for outputting geometric

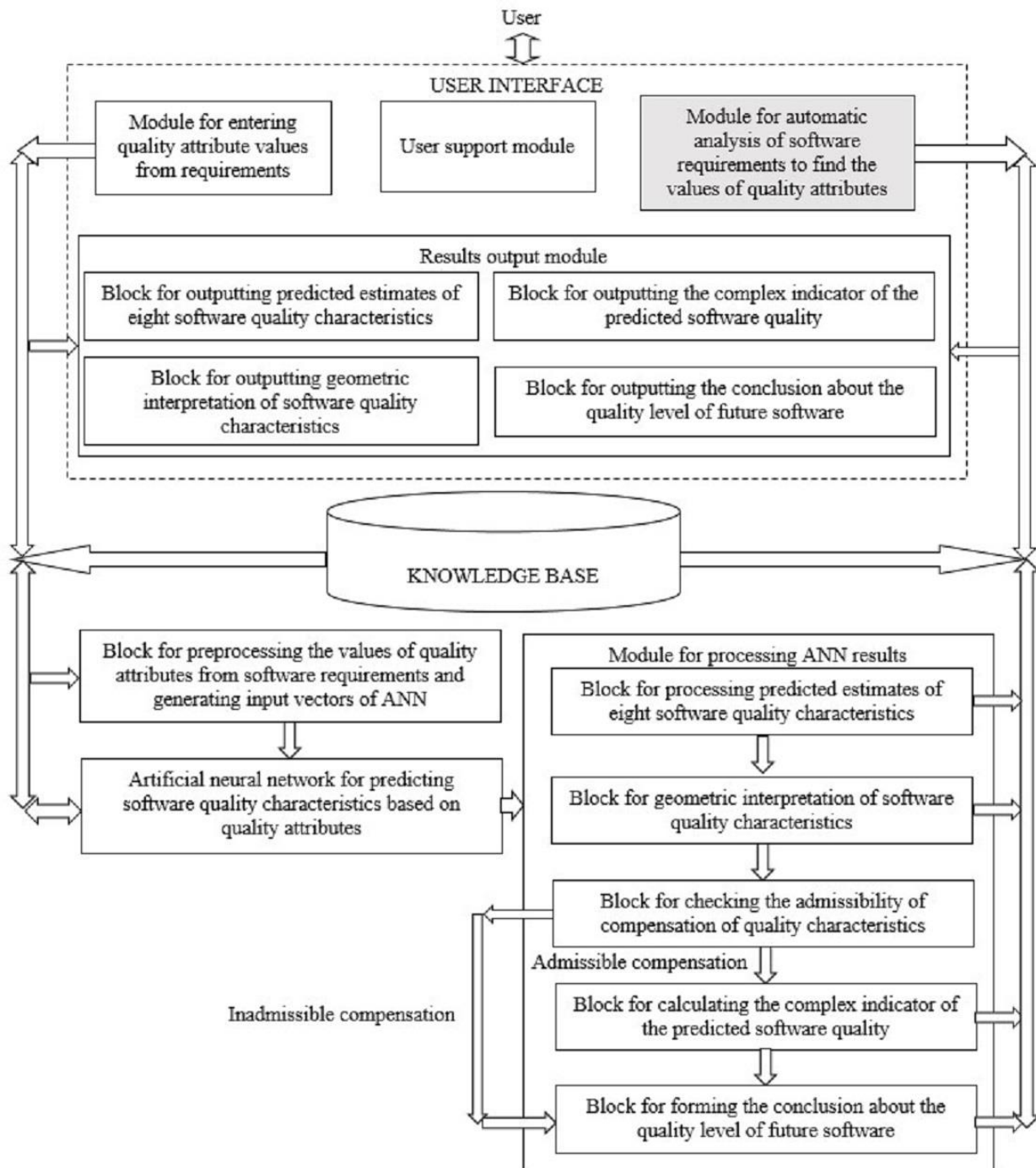


Fig. 5. The architecture of system of predicting the software quality level

interpretation of software quality characteristics, a block for outputting the comprehensive indicator of the predicted software quality, and a block for outputting the conclusion about the quality level of future software. The block for outputting predicted estimates of eight software quality characteristics gives to the user the values of quality characteristics, which are provided by an artificial neural network (its input values are formed by a block for preprocessing the values of quality attributes from software requirements and forming the input vectors of ANN) and the block for processing predicted estimates of eight software quality characteristics according to the developed method's first stage. The block for outputting geometric interpretation of software quality characteristics gives the user the geometric interpretation, which is formed by the block for geometric interpretation of software quality characteristics in accordance with the second stage of the developed method. The block for outputting the comprehensive indicator of the predicted software quality gives the value of the comprehensive indicator to the user, which is formed by the block for calculating the comprehensive indicator of the predicted software quality in accordance with the developed method's fourth stage. However, this value is formed after passing the check for the admissibility of compensation of software quality characteristics, which is carried out by the block for checking the admissibility of compensation of software quality characteristics in accordance with the developed method's third stage. The block for outputting the conclusion about the quality level of future software gives the user the conclusion, which is formed by the block for forming the conclusion about the quality level of future software in accordance with the developed method's fifth stage. The knowledge base of the system consists of a data section and a rule section. The data section stores the values of quality attributes for each set of requirements, as well as all data of the system processing (predicted estimates of quality characteristics, geometric interpretation of these values, a comprehensive indicator, conclusion about the quality level, etc.). The rule section stores rules for checking the admissibility of compensation of quality characteristics and for forming logical conclusions about the software quality level.

The system of predicting the software quality level based on requirements is implemented as a web-based application available 24/7 without registration. Currently, work is ongoing on hosting the developed application.

The proposed system of predicting the software quality level based on requirements is the tool of the developed information technology for predicting the software quality level based on requirements, which provides analysis of requirements, on the basis of which provides to the user predicted estimates of eight soft-

ware quality characteristics, geometric interpretation of the software quality characteristics' values, the comprehensive indicator of the predicted software quality, and the conclusion about the future software quality level, on the basis of which it is possible to compare sets of requirements for software and make a reasonable choice of a set of requirements for further implementation.

5. Case study, results and discussion

Using the system of predicting the software quality level based on requirements of proposed information technology, 6 sets of requirements that were developed by various IT companies in Khmelnytskyi (Ukraine) were processed to solve the same task (customer ordered the execution of the formulation of requirements for a software product from various IT companies with the purpose the selection of the best set of requirements).

Before processing these requirements' sets, we check the sufficiency of such sets using information technology to assess the quality information's sufficiency in software requirements specifications [30]. This checking provided availability of all 138 quality attributes with their values (requirements' developers iterative revised the requirements' sets with involving the customer).

Since the module for automatic analysis of software requirements to find the quality attributes' values is not yet implemented, the user used the module for entering quality attribute values from requirements and, based on manual analysis of 6 analyzed sets of requirements, manually entered the values of 138 quality attributes contained in each set of requirements.

The values of quality attributes from each set of requirements were saved in the data section of the knowledge base and transferred to the block for preprocessing the values of quality attributes from software requirements and generating input vectors for the ANN, which formed 6 sets of input vectors for the ANN. Such sets of input vectors for the ANN were sequentially fed to the input layer's neurons of the artificial neural network to predict software quality characteristics based on quality attributes. The results of the ANN functioning were processed by the block for processing predicted estimates of eight quality characteristics and displayed to the user using the block for outputting predicted estimates of eight software quality characteristics in Table 2.

The predicted estimates of the eight quality characteristics were transferred to the block for geometric interpretation of software quality characteristics, the results of which were displayed to the user using the block for outputting geometric interpretation of software quality characteristics in the following form – Fig. 6.

Table 2

Results of ANN functioning (the predicted values of 8 quality characteristics)

Requirements' set	Usability	Functional suitability	Reliability	Performance efficiency	Security	Compatibility	Portability	Maintainability
no. 1	0.52	0.48	0.53	0.49	0.45	0.44	0.55	0.60
no. 2	0.12	0.40	0.33	0.38	0.36	0.43	0.45	0.51
no. 3	0.93	0.97	0.89	0.87	0.96	0.91	0.90	0.85
no. 4	0.80	0.77	0.74	0.70	0.65	0.10	0.82	0.84
no. 5	0.13	0.16	0.14	0.20	0.15	0.13	0.12	0.15
no. 6	0.92	0.84	0.80	0.79	0.95	0.96	0.99	0.90

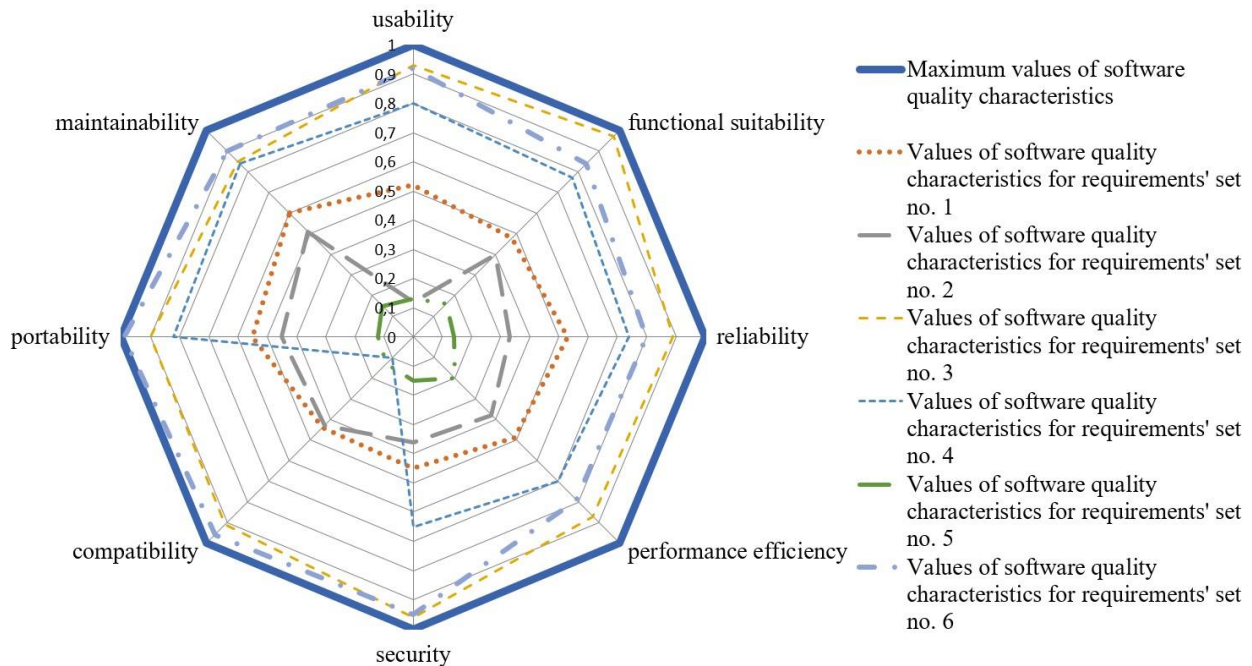


Fig. 6. Geometric interpretation of the software quality characteristics for 6 analyzed sets of software requirements

The predicted estimates of the eight characteristics and the geometric interpretation of the characteristics' values were also transferred to the block for checking the admissibility of compensation of quality characteristics. In this block all 6 octagons for 6 different requirements' sets were checked for convexity by the system. Thus, the octagons formed for sets of requirements no. 1, no. 3, no. 5, and no. 6 are convex, so compensation of software quality characteristics for these sets of requirements is admissible. The octagons formed for requirements sets no. 2, no. 4, are not convex; therefore, compensation of software quality characteristics for such sets of requirements is not admissible. For requirement sets no. 2, no. 4, the system does not form a comprehensive indicator of the predicted software quality and a conclusion about the future software quality level because such sets of requirements should not be considered and implemented (without revision and refinement) because of the very low value of one charac-

teristic compared to other characteristics, although all of them are similarly important for the developed software.

Predicted estimates of eight software quality characteristics and geometric interpretation of the characteristics' values for requirements sets no. 1, no. 3, no. 5, no. 6 were transferred to the block for calculating the comprehensive indicator of the predicted software quality, the results of which were displayed to the user using the block for outputting the comprehensive indicator of the predicted software quality in Table 3.

Values of comprehensive indicators of the predicted software quality for requirements' sets no. 1, no. 3, no. 5, no. 6 and information about inadmissibility of compensation of software quality characteristics values for sets of requirements no. 2 and no. 4 were transferred to the block for forming the conclusion about the quality level of future software, the results of which were displayed to the user using the block for outputting the conclusion about the quality level of future software in Table 3.

Table 3

Results of the system functioning (the values of comprehensive indicators and the conclusion about the quality level of future software)

Requirements' set	Comprehensive indicator	Conclusion regarding the software quality level
no. 1	0.7305	The medium quality level. If it is necessary to implement this particular set of requirements, the requirements should be further revised and refined, and additionally analyzed to predict the software quality level;
no. 2	Not determined	Compensation for software quality characteristics is not admissible. If it is necessary to implement this particular set of requirements, the requirements should be further revised, refined, and analyzed to predict the level of software quality
no. 3	2.3416	High quality level
no. 4	Not determined	Compensation for software quality characteristics is not admissible. If it is necessary to implement this particular set of requirements, the requirements should be further revised, refined, and analyzed to predict the level of software quality
no. 5	0.0615	Low quality level. If it is necessary to implement this particular set of requirements, the requirements should be further revised and refined and analyzed to predict the software quality level.
no. 6	2.2658	High quality level

Therefore, future software implemented according to requirements' sets no. 3 and no. 6 will predictably have a high quality level, so the system of predicting the software quality level based on requirements recommends choosing these sets of requirements for further work as those for which high quality software can be implemented. The other 4 analyzed sets of requirements (if there is a need to use them) need to be revised, refined, and re-analyzed using the developed system of predicting the software quality level based on requirements to establish the quality level of software that will be implemented according to the finalized set of requirements.

Thus, as shown above and experimentally proven, the proposed information technology for predicting the software quality level based on requirements (and the system of predicting the software quality level based on requirements as its component) provides analysis of quality attributes in requirements, determines the dependence (equations) of quality characteristics on attributes, forms a quantitative assessment of quality characteristics, determines the dependence (equation) of

quality on its characteristics, forms a quantitative assessment of quality, performs quality level prediction, provides all of the above services simultaneously, and demonstrates a single methodological approach.

In further use of the developed system of predicting the software quality level based on the requirements of the proposed information technology, another 284 different sets of requirements, developed by various IT companies in Khmelnytskyi (Ukraine), were considered.

For 82 sets from the available set of 290 prognoses, the system reached a conclusion about the high quality level of future software; for 98 sets, the system reached a conclusion about the medium quality level of future software; for 75 sets, the system reached a conclusion about the low quality level of future software; for 35 sets, it was concluded that compensation for software quality characteristics is not admissible (i.e., no quality level prediction was provided for those sets of requirements); therefore, only 28% of the sets of analyzed requirements can potentially be implemented with high quality software (Fig.7).

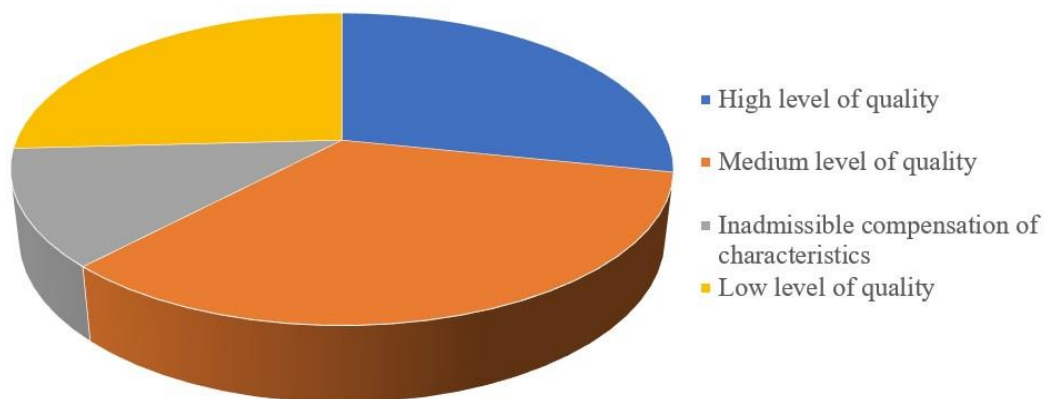


Fig. 7. Distribution of the set of prognoses for the analyzed 290 requirements' sets

Let's calculate the correctness of the 255 prognoses of software quality level, provided by the system, using formula (1):

$$\text{Precision} = \frac{75+97+80}{75+0+97+1+80+2} = \frac{252}{255} = 0.988,$$

i.e. on the 255 analyzed (with the prognoses of software quality level) sets of requirements, the system worked with almost 99% correctness, incorrectly generating one conclusion about medium quality level and two conclusions about high quality level of future software (for this 255 requirements' sets developed by several IT companies in Khmelnytskyi (Ukraine) were analyzed, for which the developed system predicted software quality level, and 255 corresponding finished programs implemented according to the corresponding requirements' sets, for which the quality level is already available).

The analysis of the available set of prognoses made by the developed system showed that only for 28% of the analyzed sets of requirements high quality software can potentially be implemented, while for 34% of the analyzed sets of requirements only medium quality software can potentially be implemented, and for 26% of the analyzed sets of requirements low quality software can potentially be implemented, and 12% of the analyzed requirements' sets should not be used for software development at all without their thorough revision because of too low value of one characteristic compared to other characteristics, i.e., in fact, only a quarter of the requirements' sets can potentially lead to the implementation of high-quality software. Therefore, as the experiments have shown, the proposed information technology and system for predicting the software quality level based on requirements provide the ability to compare sets of software requirements, created, for example, by different developers to solve the same task, to make a justified choice of requirements for later implementation of high-quality software, as well as to reject or revise unsuccessful sets of requirements that cannot be used to develop high quality software.

In addition, unlike the known tools and IT for software quality assessment, which involve human participation and human interpretation of information at almost all stages of information processing, which often leads to the significant information's loss, the proposed information technology for predicting the software quality level based on requirements involves the automation of information processing and minimization or complete elimination of subjective influence and human participation in the information handling and knowledge gaining, which is a significant advantage of the proposed information technology.

The disadvantages and limitations of the proposed information technology and system of predicting the software quality levels based on requirements are as

follows: inability to automatically analyze software requirements to find the quality attributes' values and lack of full automation of requirements processing.

In this regard, the following discussion questions remain open at present:

- is it possible to use another component of artificial intelligence, another modern popular algorithm (e.g., machine learning, ontologies, fuzzy logic, etc.) instead of ANNs to ensure the same (or better) results and the same (or higher) level of correctness of the proposed information technology and system work?
- what algorithm or method (one of the known ones or completely original) should be used to ensure automatic analysis (parsing) of natural language software requirements to find the values of quality attributes?

Conclusions

There is a contradiction between the rapid increase in the complexity and size of modern software while increasing responsibility for the performance of their functions, the increasing requirements of customers and users to the quality and efficiency of software use and the imperfection of models, methods, tools of predicting software quality at the early stages of the life cycle. Therefore, the task of predicting the level of software quality based on requirements is relevant today. The purpose of this study is to solve this task by developing an information technology for predicting the software quality level (as quantitative and qualitative characterization of the degree of satisfaction the user needs by the software when used in certain conditions) based on requirements.

The proposed information technology for predicting the software quality level based on requirements provides analysis of quality attributes in requirements, reflects the dependence (equations) of quality characteristics on attributes, forms a quantitative assessment of quality characteristics, reflects the dependence (equation) of quality on its characteristics, forms a quantitative assessment of quality, performs quality level prediction, provides all the listed services simultaneously and the model, methods, and tools underlying the IT belong to common methodological approaches and are integrated.

The scientific novelty of this article is the development of information technology for predicting the software quality level based on requirements, which, unlike the known ones, satisfies eight defined criteria in a complex and provides analysis of quality attributes in requirements, prediction of the quality level of software under development, the possibility of comparing and reasonable selection of sets of software requirements, and automation of information processing.

The proposed system of predicting the software quality level based on requirements is the tool of the developed information technology for predicting the software quality level based on requirements, which provides analysis of requirements, on the basis of which provides to the user predicted estimates of eight software quality characteristics, geometric interpretation of the software quality characteristics' values, the comprehensive indicator of the predicted software quality, and the conclusion about the future software quality level.

The developed system has significant practical value due to its ability to obtain predicted quantitative values of quality characteristics and a conclusion about the predicted level of future software quality only on the basis of the existing set of requirements, i.e., at the early stages of the software project life cycle.

The presented information technology and system for predicting the software quality level based on requirements provides the ability to compare sets of requirements for software, to reasonably select requirements for further implementation of high-quality software (as experiments have shown, this is only one of the four proposed sets), and to reject or revise unsuccessful sets of requirements that cannot be used to develop high-quality software. As experiments have shown, on the 255 given prognoses, the proposed system worked with almost 99% accuracy.

Future research. Areas for future research of the authors are:

- development of a method of semantic analysis of software requirements to find the values of quality attributes, which will ensure the complete elimination of subjective influence and human participation in the information handling and knowledge gaining regarding the future quality of future software;
- implementation of the module for automatic analysis of software requirements to find the quality attributes' values, which will ensure full automation of requirements processing;
- designing a methodology for the development and application of information technologies for assessing and predicting the level of software quality, which will provide theoretical foundations for the development and application of information technologies for assessing and predicting the level of software quality at different stages of the life cycle.

Contributions of authors: conceptualization – **Tetiana Hovorushchenko, Yurii Voichur**; methodology – **Yurii Voichur, Tetiana Hovorushchenko**; formulation of tasks – **Tetiana Hovorushchenko, Dmytro Medzaty**; analysis – **Yurii Voichur**; development of concept and methods – **Yurii Voichur, Tetiana Hovorushchenko**; development of information technology – **Yurii Voichur, Dmytro Medzaty**; develop-

ment of systems – **Yurii Voichur**; verification – **Tetiana Hovorushchenko**; analysis of results – **Tetiana Hovorushchenko, Yurii Voichur, Dmytro Medzaty**; writing-original draft preparation – **Tetiana Hovorushchenko, Yurii Voichur**; writing-review and editing – **Tetiana Hovorushchenko, Yurii Voichur**.

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

References

1. Mahaju, S., Carver, J. C., & Bradshaw, G. L. Human error management in requirements engineering: Should we fix the people, the processes, or the environment? *Information and Software Technology*, 2023, vol. 160, article no. 107223. DOI: 10.1016/j.infsof.2023.107223.
2. Guerra-García, C., Nikiforova, A., Jiménez, S., Perez-Gonzalez, H. G., Ramirez-Torres, M., & Ontañón-García, L. ISO/IEC 25012-based methodology for managing data quality requirements in the development of information systems: Data quality by design. *Data & Knowledge Engineering*, 2023, vol. 145, article no. 102152. DOI: 10.1016/j.datak.2023.102152.
3. *ISO/IEC 25010:2011. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models*. Geneva, 2011. 34 p.
4. Tamura, Y., & Yamada, S. Deep Learning Based on Fine Tuning with Application to the Reliability Assessment of Similar Open Source Software. *International Journal of Mathematical, Engineering and Management Sciences*, 2023, vol. 8, no. 4, pp. 632-639. DOI: 10.33889/ijmems.2023.8.4.036.
5. Bhandari, K., Kumar, K., & Sangal, A. L. Data quality issues in software fault prediction: a systematic literature review. *Artificial Intelligence Review*, 2023, vol. 56, pp. 7839-7908. DOI: 10.1007/s10462-022-10371-6.
6. Sadia, H., Abbas, S. Q., & Faisal, M. A Bayesian Network-Based Software Requirement Complexity Prediction Model. *Lecture Notes on Data Engineering and Communications Technologies*, 2022, vol. 139, pp. 197-213. DOI: 10.1007/978-981-19-3015-7_15.
7. *Pulse of the Profession 2023: Power Skills, Redefining Project Success. 14th edition*. Available at: <https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pmi-pulse-of-the-profession-2023-report.pdf?v=7933da8f-304b-4fe3-a655-78dace54174a&rev=427949fcdb684485a020cc72ea219f32> (accessed 21.08.2023).
8. *PMI's Pulse of the Profession 9-th Global Project Management Survey*. Available at: <https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf> (accessed 21.08.2023).
9. *The Cost of Poor Software Quality in the US: A 2020 Report*. Available at: <https://www.it-cisq.org/pdf/CPSQ-2020-report.pdf> (accessed 21.08.2023).

10. Babatunde, A. N., Ogundokum, R. O., Adeoye, L. B., & Misra, S. Software Defect Prediction Using Dagging Meta-Learner-Based Classifiers. *Mathematics*, 2023, vol. 11, no. 12, article no. 2714. DOI: 10.3390/math11122714.
11. Hai, T., Zhou, J., Li, N., Jain, S. K., Agrawal, S., & Dhaou, I. B. Cloud-based bug tracking software defects analysis using deep learning. *Journal of Cloud Computing*, 2022, vol. 11, no. 1, article no. 32. DOI: 10.1186/s13677-022-00311-8.
12. Aversano, L., Bernardi, M. L., Cimitile, M., Iammarino, M., & Montano, D. Forecasting technical debt evolution in software systems: an empirical study. *Frontiers of Computer Science*, 2023, vol. 17, no. 3, article no. 173210. DOI: 10.1007/s11704-022-1541-7.
13. Alweshah, M., Kassaymeh, S., Alkhalaleh, S., Almseidin, M., & Altarawni, I. An Efficient Hybrid Mine Blast Algorithm for Tackling Software Fault Prediction Problem. *Neural Processing Letters*, 2023. DOI: 10.1007/s11063-023-11357-3.
14. Alghamidi, A., & Niazi, M. Toward Successful Secure Software Deployment: An Empirical Study. *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*, Oulu, Finland, 2023, pp. 487-492. DOI: 10.1145/3593434.3593966.
15. Standish Group 2015 Chaos Report – Q&A with Jennifer Lynch. Available at: <http://www.infoq.com/articles/standish-chaos-2015> (accessed 21.08.2023).
16. Li, J., & Liu, S. Requirements- related fault prevention during the transformation from formal specifications to programs. *IET Software*, 2023, vol. 17, iss. 3, pp. 316-332. DOI: 10.1049/sfw2.12126.
17. Daun, M., Grubb, A. M., Stenkova, V., & Tenbergen, B. A systematic literature review of requirements engineering education. *Requirements Engineering*, 2023, vol. 28, pp. 145-175. DOI: 10.1007/s00766-022-00381-9.
18. Izonin, I., Kryvinska, N., Tkachenko, R., & Zub, K. An approach towards missing data recovery within IoT smart system. *Procedia Computer Science*, 2019, vol. 155, pp. 11-18. DOI: 10.1016/j.procs.2019.08.006.
19. Jadhav, A., Shandilya, S. K., Izonin, I., & Gregus, M. Effective Software Effort Estimation enabling Digital Transformation. *IEEE Access*, 2023, pp. 83523-83536. DOI: 10.1109/access.2023.3293432.
20. Huang, F., & Strigini, L. HEDF: A Method for Early Forecasting Software Defects based on Human Error Mechanisms. *IEEE Access*, 2023, pp. 3626-3652. DOI: 10.1109/access.2023.3234490.
21. Ahmad, K., Abdelrazek, M., Arora, C., Baniya, A. A., & Bano, M.; Grundy, J. Requirements engineering framework for human-centered artificial intelligence software systems. *Applied Soft Computing*, 2023, vol. 143, article no. 110455. DOI: 10.1016/j.asoc.2023.110455.
22. Zhi, Q., Gong, L., Ren, J., Liu, M., Zhou, Z., & Yamamoto, S. Element quality indicator: A quality assessment and defect detection method for software requirement specification. *Heliyon*, 2023, vol. 9, no. 5, article no. e16469. DOI: 10.1016/j.heliyon.2023.e16469.
23. ISO 25023:2016. *Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). Measurement of system and software product quality*. Geneva, 2016. 45 p.
24. Salomon, S., Duque, R., Montana, J. L., & Tenes, L. Towards automatic evaluation of the Quality-in-Use in context-aware software systems. *Journal of Ambient Intelligence and Humanized Computing*, 2023, vol. 14, pp. 10321-10346. DOI: 10.1007/s12652-021-03693-w.
25. Strielkina, A., & Tetskyi, A. Methodology for assessing satisfaction with requirements at the early stages of the software development process. *Radioelectronic and Computer Systems*, 2023, vol. 1, pp. 197–206. DOI: 10.32620/reks.2023.1.16.
26. Kononenko, I., Kpodjedo, M., Morhun, A., & Oliinyk, M. Information technology for choosing the project portfolio management approach and the optimal level of maturity of an organization. *Radioelectronic and Computer Systems*, 2022, vol. 4, pp. 173–190. DOI: 10.32620/reks.2022.4.14.
27. Kononenko, I., & Sushko, H. Mathematical model of software development project team composition optimization with fuzzy initial data. *Radioelectronic and Computer Systems*, 2021, vol. 3, pp. 149–159. DOI: 10.32620/reks.2021.3.12.
28. Hovorushchenko, T., Medzaty, D., Voichur, Yu., & Lebiga, M. Method for forecasting the level of software quality based on quality attributes. *Journal of Intelligent & Fuzzy Systems*, 2023, vol. 44, no. 3, pp. 3891-3905. DOI: 10.3233/jifs-222394.
29. Hovorushchenko, T., & Pomorova, O. Methodology of Evaluating the Sufficiency of Information on Quality in the Software Requirements Specifications. *Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies*, Kyiv, Ukraine, 2018, pp. 385-389. DOI: 10.1109/DESSERT.2018.8409161.
30. Hovorushchenko, T. Methodology of Evaluating the Sufficiency of Information for Software Quality Assessment According to ISO 25010. *Journal of information and organizational sciences*, 2018, vol. 42, no. 1, pp. 63–85. DOI: 10.31341/jios.42.1.4.
31. Hovorushchenko, T. Information Technology for Assurance of Veracity of Quality Information in the Software Requirements Specification. *Advances in Intelligent Systems and Computing*, 2018, vol. 689, pp. 166–185. DOI: 10.1007/978-3-319-70581-1_12.
32. Pomorova, O., & Hovorushchenko, T. Research of artificial neural network's component of software quality evaluation and prediction method. *Proceedings of 2011 IEEE 6th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Prague, Czech Republic, 2011, vol. 2, pp. 959-962. DOI: 10.1109/IDAACS.2011.6072916.

33. Pomorova, O., & Hovorushchenko, T. Artificial neural network for software quality evaluation based on the metric analysis. *Proceedings of 2012 10th East-West Design and Test Symposium*, Kharkiv, Ukraine, 2012, pp. 200-203. DOI: 10.1109/EWDTS.2013.6673193.
34. Goyal, S. Comparison of Machine Learning Techniques for Software Quality Prediction. *International Journal of Knowledge and Systems Science*, 2020, vol. 11, no. 2, pp. 20-40. DOI: 10.4018/IJKSS.2020040102.
35. Masood, M., & Khan, M. Early Software Quality Prediction Based on Software Requirements Specification Using Fuzzy Inference System. *Lecture Notes in Artificial Intelligence*, 2018, vol. 10956, pp. 722-733. DOI: 10.1007/978-3-319-95957-3_75.
36. Arora, I., & Saha, A. Software fault prediction using firefly algorithm. *International Journal of Intelligent Engineering Informatics*, 2018, vol. 6, no. 3-4, pp. 356-377. DOI: 10.1504/IJIEI.2018.10013012.
37. Tomar, P., Mishra, R., & Sheoran, K. Prediction of quality using ANN based on Teaching-Learning Optimization in component-based software systems. *Software-Practice & Experience*, 2018, vol. 48, no. 4, pp. 896-910. DOI: 10.1002/spe.2562.
38. Tripathi, V., & Singh, M. An efficient metrics based self-adaptive design model by multiobjective gray wolf optimization with extreme learning machine for autonomic computing system application. *Concurrency and Computation-Practice & Experience*, 2022, vol. 34, no. 4, article no. e6609. DOI: 10.1002/cpe.6609.
39. Alshareet, O., Itradat, A., Abu Doush, I., & Quttoum, A. Incorporation of ISO 25010 with machine learning to develop a novel quality in use prediction system (QiUPS). *International Journal of System Assurance Engineering and Management*, 2018, vol. 9, no. 2, pp. 344-353. DOI: 10.1007/s13198-017-0649-x.
40. Sun, T., Lv, X., Cai, Y., Pan, Y., & Huang, J. Software test quality evaluation based on fuzzy mathematics. *Journal of Intelligent & Fuzzy Systems*, 2021, vol. 40, no. 4, pp. 6125-6135. DOI: 10.3233/JIFS-189451.
41. Lakra, K., & Chug, A. Application of metaheuristic techniques in software quality prediction: a systematic mapping study. *International Journal of Intelligent Engineering Informatics*, 2021, vol. 9, no. 4, pp. 355-399. DOI: 10.1504/IJIEI.2021.120322.
42. Padhy, N., Singh, R., & Satapathy, S. Cost-effective and fault-resilient reusability prediction model by using adaptive genetic algorithm based neural network for web-of-service applications. *Cluster Computing - The Journal of Networks Software Tools and Applications*, 2019, vol. 22, pp. 14559-14581. DOI: 10.1007/s10586-018-2359-9.
43. Khan, M., Elmitwally, N., Abbas, S., Aftab, S., Ahmad, M., Fayaz, M., & Khan, F. Software Defect Prediction Using Artificial Neural Networks: A Systematic Literature Review. *Scientific Programming*, 2022, vol. 2022, article no. 2117339. DOI: 10.1155/2022/2117339.
44. Kaur, R., & Sharma, S. An ANN Based Approach for Software Fault Prediction Using Object Oriented Metrics. *Communications in Computer and Information Science*, 2019, vol. 955, pp. 341-354. DOI: 10.1007/978-981-13-3140-4_31.
45. Goyal, S. 3PcGE: 3-parent child-based genetic evolution for software defect prediction. *Innovations in Systems and Software Engineering*, 2023, vol. 19, pp. 197-216. DOI: 10.1007/s11334-021-00427-1.
46. Kumaresan, K., & Ganeshkumar, P. Software reliability modeling using increased failure interval with ANN. *Cluster Computing - The Journal of Networks Software Tools and Applications*, 2019, vol. 22, no. 2, pp. 3095-3102. DOI: 10.1007/s10586-018-1942-4.
47. Gordieiev, O., & Kharchenko, V. Profile-Oriented Assessment of Software Requirements Quality: Models, Metrics, Case Study. *International Journal of Computing*, 2020, vol. 19, no. 4, pp. 656-665. DOI: 10.47839/ijc.19.4.2001.
48. Gordieiev, O., Kharchenko, V., & Gordieieva, D. Software Requirements Profile Quality Model. *International Journal of Computing*, 2022, vol. 21, no. 1, pp. 111-119. DOI: 10.47839/ijc.21.1.2524.

Received 22.08.2023, Accepted 20.09.2023

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОГНОЗУВАННЯ РІВНЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тетяна Говорущенко, Юрій Войчур,
Дмитро Медзатий

Наразі наявна суперечність між швидким зростанням складності і розмірів сучасного програмного забезпечення (ПЗ) при одночасному зростанні відповідальності його функцій, підвищенням вимог замовників і користувачів до якості та ефективності використання ПЗ, з одного боку, та недосконалістю методів та засобів прогнозування якості ПЗ на ранніх етапах життєвого циклу, з іншого боку. Отже, на сьогодні актуальною є задача прогнозування рівня якості ПЗ на основі вимог. Метою даного дослідження є розв'язання цієї задачі шляхом розроблення інформаційної технології прогнозування рівня якості програмного забезпечення на основі вимог. Запропонована інформаційна технологія прогнозування рівня якості програмного забезпечення на основі вимог забезпечує аналіз атрибутів якості у вимогах, відображає залежність характеристик якості від атрибутів, формує кількісну оцінку характеристик якості, відображає залежність якості від її характеристик, формує кількісну оцінку якості, виконує прогнозування рівня якості, надає всі перераховані сервіси одночасно, в комплексі, а модель, методи та засіб, які лежать в основі ІТ, належать до спільних методо-

логічних підходів та інтегруються між собою, тобто задовольняє всі всім визначених критеріїв одночасно. Запропонована система прогнозування рівня якості програмного забезпечення на основі вимог є засобом розробленої інформаційної технології прогнозування рівня якості програмного забезпечення на основі вимог, який забезпечує аналіз вимог, на основі якого надає користувачу прогнозовані оцінки восьми характеристик якості ПЗ, геометричну інтерпретацію значень характеристик якості ПЗ, комплексний показник прогнозованої якості ПЗ та висновок про рівень якості майбутнього ПЗ, на основі якого можна виконати порівняння наборів вимог до ПЗ та обґрунтований вибір набору вимог для подальшої реалізації. Розроблені у статті інформаційна технологія прогнозування рівня якості програмного забезпечення на основі вимог та система прогнозування рівня якості програмного забезпечення на основі вимог забезпечують можливість порівняння наборів вимог до ПЗ, обґрунтованого вибору вимог для подальшої реалізації якісного ПЗ (як показали експерименти, це лише один з чотирьох запропонованих наборів), а також відмови або доопрацювання невдалих наборів вимог, за якими неможливо розробити якісне ПЗ.

Ключові слова: якість програмного забезпечення; характеристика якості програмного забезпечення; атрибут якості програмного забезпечення; набір вимог до програмного забезпечення; рівень якості програмного забезпечення.

Говорушенко Тетяна Олександрівна – д-р техн. наук, проф., зав. каф. комп’ютерної інженерії та інформаційних систем, Хмельницький національний університет, Хмельницький, Україна.

Войчур Юрій Олексійович – асп. каф. комп’ютерної інженерії та інформаційних систем, Хмельницький національний університет, Хмельницький, Україна.

Медзатий Дмитро Миколайович – канд. техн. наук, доц., доц. каф. комп’ютерної інженерії та інформаційних систем, Хмельницький національний університет, Хмельницький, Україна.

Tetiana Hovorushchenko – Doctor of Engineering Sciences, Professor, Head of the Computer Engineering and Information Systems Department, Khmelnytskyi National University, Khmelnytskyi, Ukraine, e-mail: hovorushchenko@khnmu.edu.ua, ORCID: 0000-0002-7942-1857, Scopus Author ID: 54420153900.

Yurii Voichur – PhD student of the Computer Engineering and Information Systems Department, Khmelnytskyi National University, Khmelnytskyi, Ukraine, e-mail: voichury@khnmu.edu.ua, ORCID: 0000-0003-3085-7315, Scopus Author ID: 58041891300.

Dmytro Medzaty – Candidate of Engineering Sciences, Associate Professor of the Computer Engineering and Information Systems Department, Khmelnytskyi National University, Khmelnytskyi, Ukraine, e-mail: medzatyid@khnmu.edu.ua, ORCID: 0000-0002-1879-2945, Scopus Author ID: 56008190300.