International Journal for Quality Research 15(2) 619–636
ISSN 1800-6450

Alexey S. Buryi[1]
Mikhail I. Lomakin
Alexander V. Dokukin
Evgeny V. Morin
Anatoly A. Strekha
Mikhail I. Zlydnev

Article info:
Received 10.08.2019
Accepted 20.07.2020

UDC – 004.415.5
DOI – 10.24874/IJQR15.02-16

A STUDY THE TECHNIQUES OF ASSESSING THE QUALITY OF SOFTWARE PRODUCTS

Abstract: *In order to improve the scientific and methodological base of data and measurement systems of testing laboratories within the stage of software products certification, the article suggests an integrated approach to assessing the quality of software. The approach includes structural analysis of information and measurement tasks, implementation of ISO standards when developing information features models and creating databases as well as formalized representation of the vector of* features of *quality in the course of software testing, based on comparison of current measurements with specified requirements of standards (images). A cognitive model of software product quality was obtained on the basis of the analysis of fuzzy cognitive maps and weakly structured scenarios of interaction between external and internal factors affecting the formation of quality properties of software products. The results of modeling are presented and conclusions are formulated.*

Keywords: *Software quality; Standardization; Certification of software tools; Cognitive modeling.*

1. Introduction

The evolutionary process of convergence of knowledge and technology is one of the dominant trends in the development of modern society. This involves interdisciplinary integration in the fields of nanotechnology, biotechnology, information and cognitive sciences (Roco& Bainbridge, 2013). This way, software is the main functional link in the structure of information processes. And software quality determines the effectiveness of information systems in general (Arsovski & Arsovski, 2008) as well as plays an important role factor in a number of particular tasks, such as improving the quality of information systems by providing a fault-tolerant computational process (Buryi, 2016) and improving the quality of business processes by improving data presentation models (Arsovski et al., 2012).

Currently, following the rapid development of information and communication technologies, the issue of improving the quality of software products, in the sense of ISO 9001, becomes quite relevant. This is more important especially given the advantages of the latest edition-ISO 9001: 2015 (see Fonseca & Domingues, 2017), (Kakouris and Sfakianaki, 2019) and ISO/IEC 25000 (Boiral, 2012), and in particular, within the framework of the software quality model, which is a certain set of characteristics and relations between them, for ensuring specific quality requirements and evaluating this quality (Amin and Salih, 2017).

The pace of development, distribution and implementation of computer tools in

[1] Corresponding author: Alexey Buryi
Email: a.s.burij@gostinfo.ru

everyday life has led to an immeasurably powerful flow of software, which determines the modern functionality of the developed computer equipment. Thus, Global IT spending is expected to reach $ 3.7 trillion in 2018, according to the survey (Sakovich, 2018). The task of increasing the role of software moves into a new plane when it should be implemented with the help of smart technologies, big data analysis, cloud services, etc. (Porter & Heppelmann, 2015; Abd-Elrahman et al., 2020).

An important role in maintaining and improving the quality of software products is played by certification of software products (Aleksic et al., 2017). Such role is being carried out by special licensed bodies (test laboratories, centers) within the framework of ISO/IEC 17025:2017, which provide the control of the software products quality, their safety, both at the level of individual programs and functional or target modules: application programming interfaces (see ISO/IEC 24730:2014 standards), application packages and many others.

In accordance with the current Russian legislation, it is mandatory to possess software certification for software and databases of soft- and hardware complexes, ensuring the protection of state information resources (Barabanov et al., 2017) and confidentiality of information in an infrastructure of information systems, both in government and in commercial enterprises. Software certification has an additional user benefit – a guarantee of quality, providing a set level of compatibility (through specifying an operating system, necessary resource constraints, input-output data formats, etc.).

ISO/IEC 17025:2017 establishes general requirements for the competence, impartiality and consistency of operation of laboratories that certify software products in the framework of ISO 9000, ISO 9001, ISO 10012, and several others.

This paper has the following structure: the second section provides a literature review as well as describes the structure of the scientific problem solved in the research; the third section describes a cognitive-map based model; the fourth section describes the results of machine modeling and their discussion. Finally, the fifth section contains the conclusions of the work.

2. Literature review and methodology

Increasing complexity of the functions implemented by programs in the information environment leads to an increase in their volume and complexity of their creation and development. At the same time, there is an increase in the requirements for service capabilities of programs, compatibility with various operating systems and other factors that are not directly related to the intended purpose of a software product.

The noted features lead to a rise of defects and programming errors, most of which are eliminated during the testing of programs by a manufacturer. However, in many cases, the complete identification and elimination of possible software defects can only be carried out during the certification of software products. In this regard, we need a mechanism that is able to detect existing defects in time, to predict possible consequences when they occur and to develop requirements for ensuring the quality of software products.

The task of this research is to offer scientifically based recommendations and develop models of software quality control for improving information systems of software products testing based on the analysis of organizational and technical interaction factors in the processing of information during the certification tests of software tools.

There are a number of works that anticipated this study. Among those are the works on the organizational aspects of the interaction of information systems. For example, see the following papers on a number of selected topics: (Ivanyuk et al., 2014), (Klir, 1985), (Yusupov & Musaev, 2017); improving the model-algorithmic support of information control systems and complexes (Sokolov and Yusupov, 2006), relevant measurement systems and communication circuits (Buryi et al., 1998), methods for evaluating and optimizing data structures (Buryi, 2016), (Lovtsov et al., 2018), (Kul'ba, 2005); quality assessment and standardization of products in general, and software in particular (Boehm, 2006), (Lomakin & Glushakova, 2015).

The matters of improvement of the system of certification and testing of software are studied in sufficient detail in the papers by (Barabanov et al., 2017), (Dugalic & Mishev, 2012), (Eda & Do, 2019). However, those papers stay on quite a conceptual level concerning the management models and the considered requirements. They are, as a rule, poorly suitable for practical use in analyzing software quality indicators since estimates used provide an integration picture, from which it is difficult to identify the role of particular indicators and individual functional subsystems.

Quality indicators of information systems should reflect both the target factors of the system and individual functional quality indicators in the decomposition of management processes and information processing (Buryi, 2016) taking into account the management of quality metrics for organizational structures, including management of human groupings (Arsovski & Arsovski, 2008).

The global IT industry is characterized by constant growth. It is forecasted to reach $5 trillion in 2019. Although according to the report (CompTIA, 2019) the share of software is only 11% of the stated amount (Figure 1), which in absolute terms also significantly and largely determines the development vector of informatization of modern society along with the fundamental and applied research in computer science, with the development of new information technologies, the development of the information processing industry and the adjustment of relevant legal and business regulations (Yusupov & Musaev, 2017).

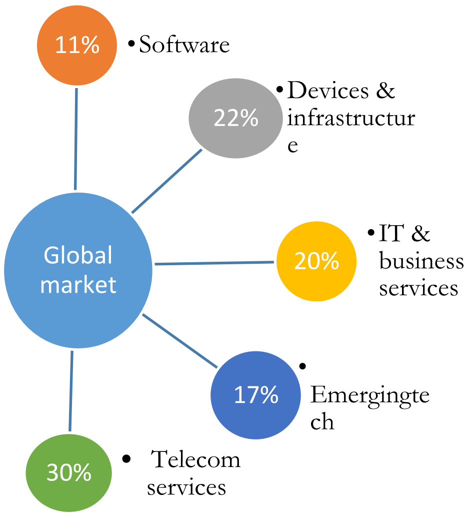

Figure 1. Structure of the global information technology industry

Therefore, it is necessary to develop a new approach to the organization of management processes, data retrieval and organization of tests for quality control programs, providing a comprehensive and interrelated approach to the technology of certification tests.

2.1. Ageneral structural approach

The organizational and technical structure of test complexes should be presented as a combination of three areas (areas of activity: 1 – technical-technological – A_T; 2 – metrological (as part of information, measurement and legal aspects) – A_M; 3 – practical usage (customers and consumers of measurement and communication information, data) – A_P (see Figure 2).

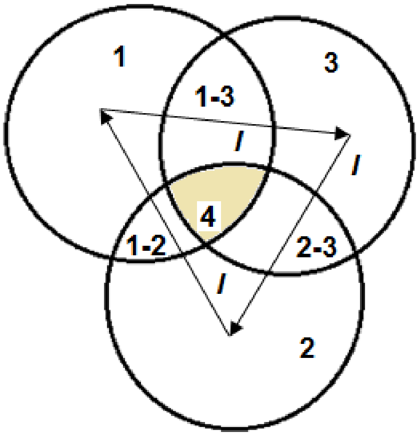

Figure 2. Structure of interaction of organizational, technical and information spheres in the course of testing

The intersection of the marked spheres can be viewed as separate subsystems. So, 4 – this area is connected with the process of obtaining of the conformity assessments of the product or service to the specified standard resulting from the intersection of the corresponding sets, which we can represent in a form of Cartesian product of sets:

$$A_{St} = A_T \times A_M \times A_P \,. \qquad (1)$$

Area 1-2 or$A_T \times A_M$ is characterized by testing means of measurement at the stage of their preparation for use. Region 1-3 or$A_T \times A_P$ corresponds to the stage of repair, calibration and maintenance of measuring instruments and testing. Area 2-3 or $A_M \times A_P$– activity on the formation of standard and metrological information for the preparation of the testing process (software testing).

The Figure 2 conditionally highlights the outline of information interaction (I) between these areas, which implies an informational exchange of data (test results) between subjects – participants in the processes.

It is known that protecting the interests of consumers, as well as public institutions based on product quality control, by ensuring their high consumer properties and increasing the objectivity of operational, ergonomic evaluations of product characteristics is the main purpose of certification of a design process, production of software.

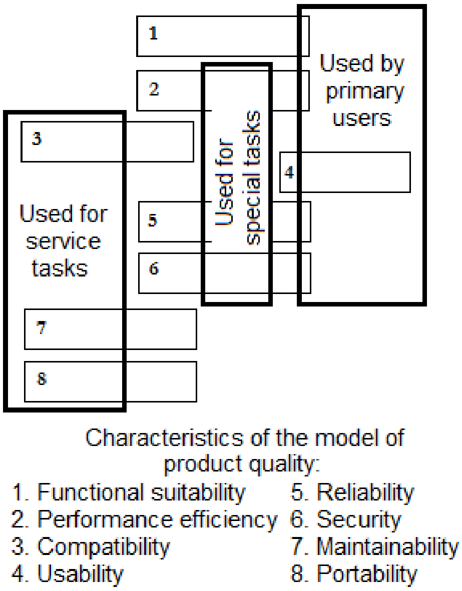

Figure 3. Classification of quality characteristics by usage

Figure 3 generally describes 8 main software quality indicators in accordance with ISO 9126, ISO 25000 standards and their implementation in the product quality models.

Depending on the target tasks carried out by the product with the installed software, its certain characteristics can be "active". So, for most user tasks, the employed characteristics are 1, 2, 4-6, which allow a user to simultaneously take into account the quality of software, hardware, computing environment and user ratings, that is, area 3 (see Figure 2). In product quality model for a service task, characteristics 3, 7, 8 are often used. Activation of quality characteristics 2, 5 and 6 is required for a number of special tasks, when a software installation is characterized by a single installation or an autonomous use (without the possibility of reinstallation, that is, in a limited functional environment). They are used when it is not possible to extend the life cycle of a product, or it is not possible to repair, replace individual structural elements during operation.

2.2. Subject area and data model

The conceptual scheme, which serves as the basis of the developed *conceptual model*, which is the essence of the question, requires specialists (knowledge) in many related areas involved in the development, design, manufacture, testing, quality control, and other areas specific to the life cycle of software tools. With all its initial roughness and inaccuracy, this model enables us to get a general idea of the researched area, to relate parts of the process, using information models in a form convenient for representing the research problem.

The requirements for the data structure determined by their type and the adopted data model are reflected in the ISO 8000:2016(E) standard. These requirements are based on the data dictionary and the corresponding formal syntax. However, in order to develop databases for a specific subject area, it is advisable to provide a transition from a vocabulary approach to the interaction of individual concepts to their taxonomy, based on arbitrary connections and relationships between concepts while forming the information field of concepts (C) and terms (T). The next stage of development should be a strict specification of relations and relationships between the C and T sets that have an ontological nature, which makes it possible to realize hybridization (merging) of the concepts of databases and knowledge bases (Figure 4) due to the growth in the information presented in the semantic component.

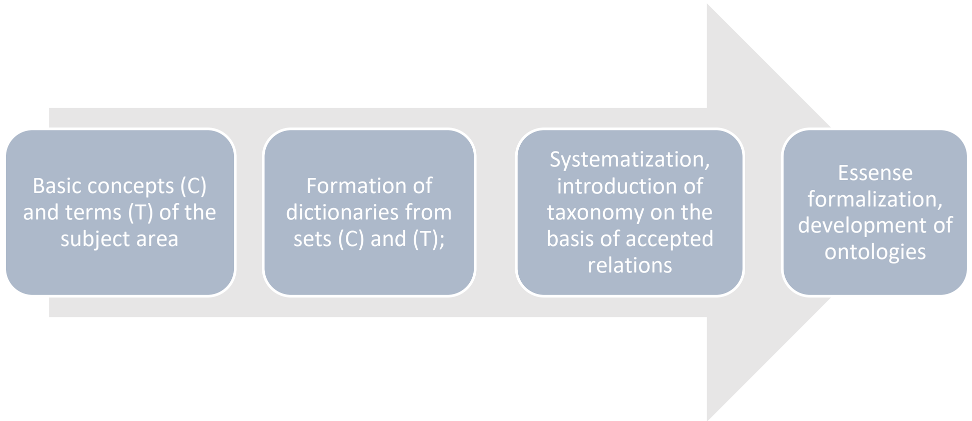

Figure 4. The growing role of semantic component in the database

The growth of the semantic component extends the capabilities of the distributed database models, ER-models (“essence”–“relation”), due to the greater possibilities of semantic relations and the logic of their construction, aggregation of connections and meta-properties of the relations themselves. An important factor in the advantages of usage of semantic links in databases is expressed in the simplification of forms for user query creation as well as in improving the speed of execution of received queries.

In order to form a database and a representation of relationships between individual entities we should introduce a concept of relationships, which traditionally has tabular form, where rows are entities, and columns are attributes or dimensions. Let us denote the set of relations and, accordingly, the set of attributes defined on them as follows:

$$\mathrm{R} = \{R_1, R_2, \dots, R_t\}, \mathrm{A} = \{A_1, A_2, \dots, A_g\}, \quad (2)$$

where $\boldsymbol{t}$ – the number of relationships or the power of a set of relationships R and $\boldsymbol{g}$ – number of attributes.

In this case, an attribute is a pair $A_i = \left(name(A_i): dom(A_i)\right)$, $(i = \overline{1, g})$, that includes the attribute name and the attribute domain as a set of its valid values.

The benefits in usage semantic links in databases are provided by:

- high level of data independence;
- simplification of forms for user query creation as well as in improvement of speed of execution of received requests;
- the possibility of using high-level programming languages in database design (Kul’ba & Sirotyuk, 2005).

2.3. The formalization of the objective of the study

Let us denote the object of testing and certification (software) in the form of a tuple of four sets that form a certain system Σ:

$$\Sigma = \langle T^{ST}, X, W, U, Q \rangle, \quad (3)$$

where $T^{ST} = \{T_1^{ST}, T_2^{ST}, \dots, T_H^{ST}\}$ – set of types of software tools (ST) (application programs, programs of service, operating systems, etc.), subjected to testing; $X = \{x_1, x_2, \dots, x_m\}$ – a set of condition of software tools; $W = \{w_1, w_2, \dots, w_N\}$ – a set of attributes (indicators of the quality of software tools) controllable during testing; $U = \{u_1, u_2, \dots, u_M\}$ – a set of testing models applied at different stages of software testing; $Q = \{q_1, q_2, \dots, q_N\}$ – a set of rules of transformation of conditions into attributes, that is $Q: X \to 2^{|W|}$, where $2^{|W|}$ – the number of possible subsets of a cardinality set, which is denoted as $|W|$.

The set of attributes for k-ith type of software tools forms the following vector:

$$\varphi_w(T_h^{ST}) = \left\{\varphi_{w_j}(T_h^{ST})\right\}, \quad j = \overline{1, N}, h = \overline{1, H}, \quad (4)$$

On the other hand, the value of attribute w_j is determined by measured parameters specific to testing of the h-th type of software tools, $\varphi_{w_j}(T_h^{ST})$ that is there is there exists a membership function reflecting a set of real values of the numerical axis on a set of quality attributes in a corresponding measurement scale.

It is possible to write the following notation for quantitative (real) indicators of various forms of features of quality:

$$w_j = \varphi(y_j) \in \left[y_{j\_}, y_j^{-}\right], \ j = \overline{1, N},$$

where $y_{j\_}$ and y_j^{-} - are respectively the lower and upper bounds of the interval of values of the j-th attribute.

The membership function is represented in the form of the following expression (5) with the qualitative assignment of features, using linguistic variables, for example, “High”, “Medium”, “Low”, employed in expert assessment, which can be recalculated in a base interval [0,1], or can be represented by a two-element set, in accordance with the "No" or "Yes" responses received from the experts:

$$\varphi_{w_j}(T_h^{ST}) \to [0,1] \oplus \{0,1\}, j = \overline{1,N}, \qquad (5)$$

where symbol $\oplus$ means strict disjunction.

The model of measurements of the tested software features (properties) is set in a form of a mapping of controlled software tools to the space of measured parameters:

$$\Psi: T_h^{ST} \to \{u_{h1}(w_1^d, w_1^s), u_{h2}(w_2^d, w_2^s), \dots, u_{hl}(w_j^d, w_j^s), \dots, u_{hL}(w_N^d, w_N^s)\},$$
$$j = \overline{1,N}, \qquad l = \overline{1,L}, \qquad h = \overline{1,H},$$

where $u_{h1}(w_1^d, w_1^s)$ is the function characterizing the model of testing the first quality attribute for the h-th type of software tools, in the form of the correspondence of the measured attribute (data) w_1^d to its standard (required) value w_1^s.

We may denote a set of features of quality in the form of requirements for a certain type of software tools as a standard $[1 \times N]$ size vector w^s in the form of:

$$w_h^s = \{w_{hj}^s\}, j = \overline{1,N}$$

or

$$w_h^s = \{w_{h1}^s, \dots, w_{hN}^s\}^{\mathrm{T}}, \qquad (6)$$

where w_{hj}^s is the required (reference) level of the j-th feature, $h = \overline{1,H}$ – is the type of software tools, and "T" is the sign of the vector transposition.

The notion of proximity of quality features to existing standards and norms (see Buryi, 2019) is quite important in a matter of solving the task of controlling and measuring features of quality from the point of view of algebra of creation of a signature (a set of operations on a set of data and set of relations with respect to internal relations) within the measuring problems solved (for multi-attribute comparison of objects, when vector attributes of quality are compared in the space of qualitative attributes).

For a set of possible states X of the system Σ a set of identification features (characteristics) is formed:

$$I_C = \{I_{C1}, I_{C2}, \dots, I_{Cm}\}, \qquad (7)$$

where I_{Ci} are the indicators for the i-th ($i = \overline{1,m}$) state of the system Σ and the number of states of the latter with the task in question is quite small, for example:

1) the tested software (ST) tools fully comply with the stated requirements and have been successfully tested;
2) the tested ST require improvement in order to obtain the level of certification required by the customer;
3) the tested ST do not satisfy some secondary quality indicators and can be certified for a limited use – for less critical tasks (at the request of the customer), etc. For a typical case, these are the first two states when decisions are made to issue or deny a certificate of conformity.

The identification attributes I_{Ci} from (7) characterize the fact of the presence of a specific quality indicator (attribute) in the i-th state, which is represented by a Boolean vector $b = (b_1, b_2, \dots, b_N)$, in the form of a set of numbers, each of which takes the value "1", when the attribute (quality indicator) is executed and goes to processing, or the value "0" – otherwise. Let us denote this as follows:

$$I_{Ci}: w_i \bigwedge_{j=\overline{1,N}} (b_{i1}, \dots, b_{ij}, \dots, b_{iN}), \quad (8)$$

where $w_i = \{w_{ij}\}$ and b_{ij} with $j = \overline{1,N}$, respectively, are the totality of quality attributes and Boolean values for the i-th state. Expression (8) enables us to form a set of features of quality corresponding to a given testing object.

We will identify the system Σ with an identification matrix, for which we should introduce the following definition.

The matrix of identification features (IF) M_{IF}^{h} of size $[m \times N]$ formed for each software tool of type $h, h = \overline{1,H}$ from the set T^{ST} (3), has a matrix, in which rows are identification marks (7), and the number of rows is determined by the number of states of system Σ.

Various forms of representation of individual features of quality, measured during testing both in absolute scales and in relationship scales with different bases, lead to the necessity of comparing feature vectors of the form (4) for individual components. When comparing the measured j-ith indicator w_j^d with the corresponding standard value w_j^s, the preference of the first is recorded as $w_j^d \succ w_j^s$, and, correspondingly as $w_j^d > w_j^s$ for numerical data notation. The numerical characteristic of preference in this case is the relation $\nu_j = w_j^d / w_j^s$.

Let us denote the utility function Φ in a form of a result of a multicomponent comparison in the process of expert evaluation of each measured indicator compared to its reference value (Fishburn, 1974):

$$\Phi(\nu, \beta) = \prod_{j=1}^{N} \delta_j \nu_j^{\beta_j}, j = \overline{1,N}, \quad (9)$$

where β_j – is the parameter of the expert's preference according to the results of the testing of the j-th attribute, with restrictions $\beta_j > 0$, $\sum_{j=1}^{N} \beta_j = 1$ for all $j = \overline{1,N}$; δ_j – the indicator parameter that takes the value “1” on $w_j^d \succcurlyeq w_j^s$ and the value “0” otherwise.

The function $\Phi(\nu, \beta)$ of the form (9) is the weighted geometric average of the results of comparisons of indicators during their sequential testing. It is characterized by the properties of multiplicativeness, monotonicity and homogeneity according to the vector of features (4) as well as properties of invariancy with respect to the measuring scales of controlled features of quality.

Therefore, as result of sequential testing of software tools of a given type it is concluded that the controlled ST meets or fails to meet the test documentation requirements. And the utility function of the form (9) will be considered as a generalized quality characteristic of software tools with fixed preferences when analyzing data.

In order to organize the control of a batch of software tools of a given type and volume, we will determine a generalized quality indicator for all batch elements identifying a sample characterized by the maximum level of utility:

$$T^{ST^*}: \underset{f}{\operatorname{argmax}} \left(\Phi_f(\nu_j, \beta_j) \mid j = \overline{1,n}, \; f = \overline{1,F}, \sum_j \beta_j = 1 \right), \quad (10)$$

While preparing and planning software testing, a set $U_p \subset U$ of testing models is developed and the scales of attributes are normalized to the form of piecewise linear monotonic utility functions. The analysis of technical, topological and informational structures during assessing the capabilities of the testing base of the laboratory (certification authority) for conducting complex testing of software in accordance with the technical task allows us to draw preliminary conclusions about the consistency and reliability of the planned works.

3. Application of the model

Cognitive maps are useful for creating (developing and designing) complex software systems. The advantage of the cognitive approach is that for a specific subject area it is possible to identify indicators (factors) that reflect the diversity of technological processes at the stages of the software product life cycle as well as take into account the economic and social aspect of the considered task.

The main emphasis of cognitive modeling is made on weakly structured scenarios, when it is necessary to take into account elements and technological features of various nature, organizational structures, social, economic, cultural and legal factors in the circuits of information interaction of management subsystems. This list can be significantly extended depending on the object being analyzed, provided the investigator has a priori information on the structure of his external and internal links.

The main feature of semi structured tasks is that additional information is required on the interaction (internal influence) of individual subsystems and on factors in related branches of knowledge that directly or indirectly affect the target factor (the studied process – improvement of software quality).

Building a formalized model of management of metrological activities allows not only to perform its analysis for optimization, but also to identify material and information flows, to identify processes that are subject to management within the quality management system ISO 9001, see, for example, (Vujovic & Krivokapic, 2009).

The practice of developing methodological tools in terms of quality models is aimed at obtaining software quality assessments (Laporte and April, 2018) depending on the metrics used to represent the analyzed software characteristics (Ortega et al., 2003), (Lochmann, 2013), (Dordevic et al., 2015), (Koteska et al., 2018) informational links, both internal and external (for example, at the level of a typical interface), as well as ensuring the required level of program security and protection (Salva & Regainia, 2019).

Cognitive models are increasingly used to evaluate a number of factors influencing the process of creating (developing and designing) complex software systems, as well as processes of co-management and application, (Abramova et al., 2010), (Kulivets, 2011), (Kokkinos et al., 2018).

Cognitive maps are characterized by structure, clarity, simple and understandable interpretation due to a causal relationship between the analyzed factors.

Table 1 presents a qualitative and quantitative approach in solving research problems. Each of those approaches relies on its own methods, models, and algorithms (Novikov, 2010).

Table 1. Tools for analysis of development of situation

Purpose / method	**Tool**	
	Qualitative analysis	Quantitative analysis
Description of situation	Cognitive maps	Differential or difference equations
Analysis and centralized control of situation	Simulation	Optimal control theory
Interaction analysis for parties interested in development of situation	“Cognitive games”	Dynamic games
Multidimensional scaling	Methods of classification, typological and systematization of data	Direct measurements and numerical representation of data
Factoranalysis (FA) and modeling	Expert analysis, theory of behavior, hierarchical FA	Determined FA, stochastical FA

Cognitive modeling, in our opinion, provides a wider view at the problem of improving the software quality. During ST certification, its quality control is carried out. This step gives an estimate both to the product itself and to the technology for its development, design and usage. Cognitive maps can additionally take into account factors not only of the technological level of production, but also the technology of product implementation, the economic aspects of market realization, and social perception. This ensures timely understanding and implementation of new market demands, technological "breakthroughs" in related areas of technology providing users with both modern service capabilities and advanced technological solutions

Let us denote a fuzzy graph describing a fuzzy cognitive map in the following form:

$$G = \langle B, R, P \rangle, \qquad (11)$$

where the components of a tuple are: $B = \{1,2,\dots,n\}$ – a set of vertices describing the simulated scenarios (concepts) of the graph; $R \subseteq B \times B$ – a set of arcs (causal relationships between concepts), moreover $R = \{r_{ij} | r_{ij} \in R; i,j = \overline{1,n}\}$, where n is the number of vertices of the graph, and the element r_{ij} indicates the influence of the i-th factor on the j-th factor; P – a set of weights (characteristics) of links.

According to (Gray et al., 2014), cause-and-effect relationships between objects in the form of weighted arcs in the map structure, originate from constructivist psychology, which is the most adequate to an interactive interaction, both interpersonal and collective experience.

It worth mentioning that in cognitive models there are positive and negative causal relationships. For the first ("+") is an increase in the concept (factor) – "cause" leads to an increase in the concept - "consequence". For the second ("–") - an increase in the concept - "cause" leads to a decrease in the concept - "consequence". The stability of the cognitive model depends on the nature of the feedback between the vertices of the cognitive map graph.

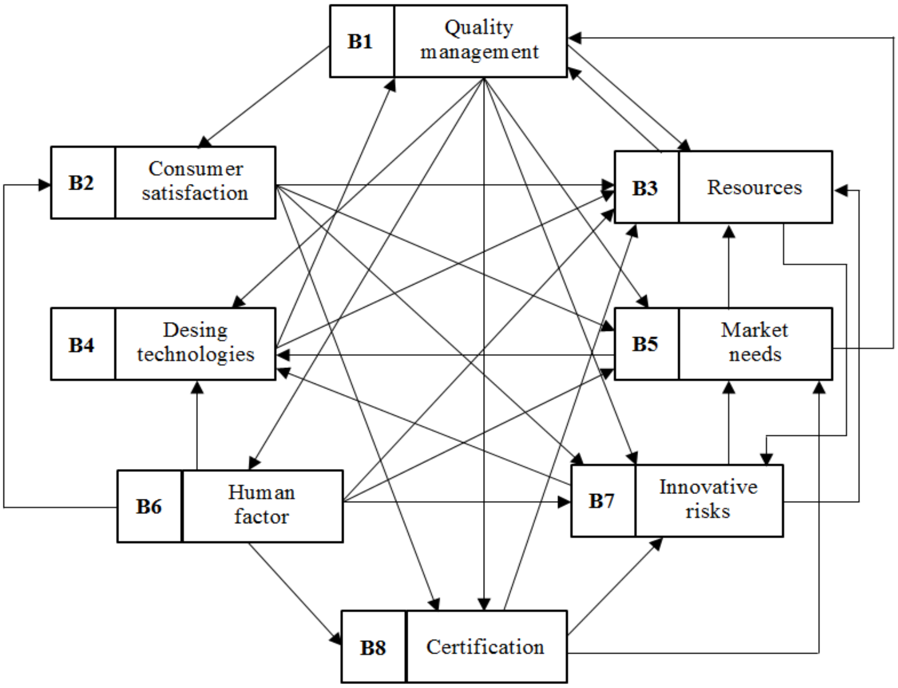

Figure 5. Structure of fuzzy cognitive map for factor estimation of impact on quality of software tools

A discussion of the logic of constructing a fuzzy cognitive map boils down to the following (see Figure 5):

- ensuring a high level of quality of software tools (concept **B1**) is based on resources (**B3**) (informational, financial, and others);
- the reduction of innovation risks (**B7**) is ensured by the results of certification activities (**B8**), the qualifications of service personnel and, in general, the human factor (**B6**) as well as the level of quality of the developed software tools (**B1**);
- consumer satisfaction (**B2**) is a positive factor for the concept of resources (**B3**) in the form of payment for services and is an
- indirect incentive for the software market development (**B5**);
- design technologies (**B4**) have a positive effect on the quality of software products (in development, in testing) requiring an expenditure of resources for modernization and development (negative impact on (**B3**);
- human factor (**B6**) contributes to the growth of technology; such factor is in demand in the market, provides high-quality interaction with ordering organizations (**B2**), has a positive effect on certification tests (**B8**), reduces innovative risks (**B7**) due to intellectual capital, requires investment in training specialists – the connection with the concept (**B3**) is negative.

4. Results

Formation of a matrix of relationship values characterizing the influence of interacting concepts (factors) on each other. In this case, the methodological apparatus may be comprised of the following: statistical methods of data analysis, the results of expert surveys, analytical reviews of the information and analytical database of the subject area, methods of decision-making theory, assessment theory, normative legal acts, etc. The result of this stage is a cognitive matrix of mutual influences in the form of Table 2.

Table 2. Cognitive matrix of mutual influences

Factors	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	**B7**	**B8**
B1	0	0,75	0,4	0,1	0,15	0,25	-0,6	0,9
B2	0	0	0,9	0	0,3	0	0,1	0,45
B3	0,4	0,5	0	0	0	0	-0,3	0
B4	0,8	0	-0,8	0	0	0	0	0
B5	0,05	0,05	0,6	0,2	0	0	0	0
B6	0	0,2	-0,3	0,4	0,3	0	-0,35	0,3
B7	0	0	0,65	0,3	0,05	0,02	0	0
B8	0	0,05	0,12	0	0,65	0	-0,25	0

The dependencies represented in figure 6:

$$B(0) = [0{,}55; 0{,}1; 0{,}4; 0{,}5; 0{,}2; 0{,}65; 0{,}6]^{\mathrm{T}}.$$

were created as a result of calculation of dependencies of concepts within the time interval with respect to the weights of influence, both with positive and negative feedback links (see Figure 5) given a vector of initial values for concepts.

The initial state vector of cognitive map factors tends to increase if the direction of the process movement occurs only along arcs with a "+" sign, but it can also represent an oscillatory (unstable) process when alternating the sign of influence weights on the factors of the model.

Such an approach allows:
1) to identify management factors that provide the required scenarios for a development of a situation, its assessment and elimination of possible negative effects from the impact of destructive factors affecting the quality of software;
2) to obtain operational assessments of the influence of a variety of factors of external and internal conditionality on the target concept, including them in the cognitive model based on the existing technological, communicative, market, market situation obtained during the preliminary or ongoing monitoring of the subject area.

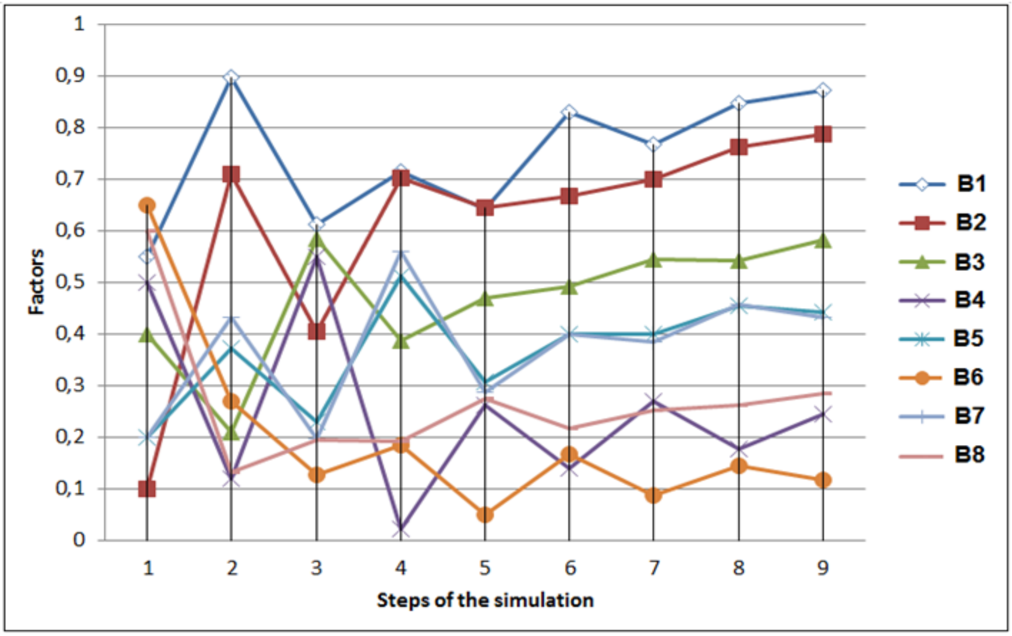

Figure 6. Results of calculation of values of Factors for fuzzy cognitive map by tacts of the model time

The following assumptions were made while carrying out the development of a model for assessing the consistency of expert opinions in the process of analysis of the evaluation results of the quality attributes of software products during their certification tests.

The resulting expert assessments are subjective and are determined by the level of qualifications of experts, their preferences and priorities. The *m* number of experts participate in the tests, each of whom gives an assessment of each attribute from their total number N in the following form:

$$x_{ij} = \begin{cases} 2, \text{IF } w_{ij}^{d} > w_{i}^{s}; \\ 1, \text{IF } w_{ij}^{d} = w_{i}^{s}; \\ 0, \text{IF } w_{ij}^{d} < w_{i}^{s}, \end{cases} \qquad (12)$$

where each element x_{ij}corresponds to the *i*-thfeature obtained from the *j*-th($j = \overline{1,z}$) expert.

For each *i*-th attribute, the quality of expert opinion, taking into account the expression (12) can be represented as a sum:

$$z_i = z_{di} + z_{=} + z_{si} \,,$$

where z_{di} – is the number of experts, which determined that the quality of the controlled indicator is noticeably higher than that of the standard; z_{si} – the number of experts who gave a preference for this indicator to standard value; $z_{=}$ – the number of experts who determined that the indicator of the *i*-th quality indicator of the test sample is equal to the reference quality level. This way we may consider that the sum $(z_{di} + z_{=})$ determines the number of positive results of evaluation of the quality attribute. The *Expected value* for the *i*-th quality attribute may be finally denoted as:

$$\bar{x}_i = 1 + \frac{z_{di} - z_{si}}{z_i}. \qquad (13)$$

Table 3 presents the results of the evaluation of quality attributes - D1 – D14; five experts E1–E5; total scores for each sign; average scores for each feature and average scores for each expert.

Table 3. Results of expert evaluation

Indicator	E1	E2	E3	E4	E5	Total assessment	Average assessment
D1	1	2	2	2	2	9	1,80
D2	0	2	1	2	2	7	1,40
D3	2	2	0	1	2	7	1,40
D4	2	2	2	2	1	9	1,80
D5	1	2	2	0	2	7	1,40
D6	2	2	2	2	2	10	2,00
D7	2	0	2	2	0	6	1,20
D8	2	2	2	2	2	10	2,00
D9	1	1	0	1	1	4	0,80
D10	2	1	2	2	2	9	1,80
D11	1	0	1	1	2	5	1,00
D12	2	1	0	1	2	6	1,20
D13	1	0	1	1	0	3	0,60
D14	2	1	0	0	2	5	1,00
Average of the Experts	1,500	1,286	1,214	1,357	1,571	97	
Normalization	0,216	0,186	0,175	0,196	0,227	1,000	

It is required to formulate an opinion regarding all quality indicators, including those indicators that have received "0" marks.

Let's a competence vector by the method of successive approximations. In order to do this, we set equal weights for all experts as the initial value of the competence vector–$K(0)$. Only the total number of experts involved and the normalization condition determines the weights that is:

$$K = \left(K_1 \dots K_j \dots K_m\right)^{\mathrm{T}}; \sum_{j=1}^{m} K_j = 1.$$

An argument for the$K(t)$ vector can be either a time parameter, or any iteration label, for example, a modeling step.

In *step 1* of the iterative process, the intermediate vector of the weights is determined from the matrix equation as

$$G(1) = Z^{\mathrm{T}} Z\, K(0), \qquad (14)$$

whereZ^{T} is the matrix of the size $(z \times N)$ of the normalized estimates (Table 3);z– is the number of experts;N– is the number of analyzed attributes of quality.

For the vector $G(1) = \left[g_1^{[1]}, \dots, g_j^{[1]}, \dots, g_z^{[1]}\right]^{\mathrm{T}}$ let us determine the total value across all of its components, after which we obtain the j-th components of the competence vector for *step 1*:

$$K_j(1) = g_j^{[1]} \Big/ \sum_{j=1}^{m} g_j^{[1]} . \qquad (15)$$

The vector $K(1)$ is formed, therefore, as a result of the calculation carried out by expression (15).

The *step 2* of the iterative process following to the formula (14) takes into account $K(1)$, determines $G(2)$, after which the process is repeated and, using (15), we calculate the competence vector $K(2)$. An indicator of the completion of the iterative process is the limit on the maximum magnitude of the increment modulus– V_K, that is

$$\max_j \left| K_j(t) - K_j(t-1) \right| < V_K.$$

for any component of the competence vector.

Figure 7 illustrates the changes in the components of the competence vector.

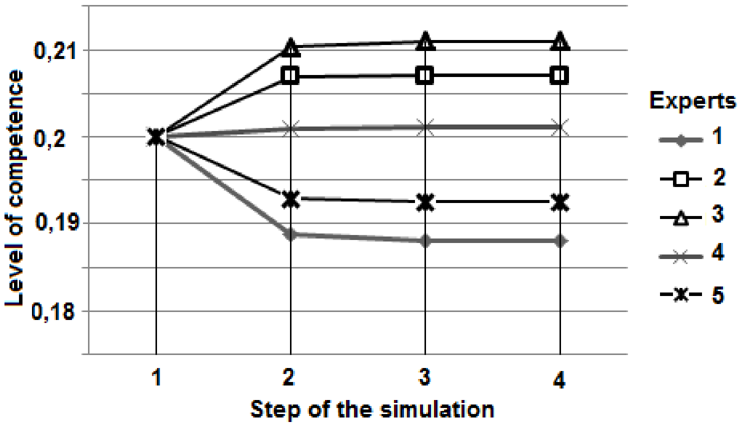

Figure 7. Dynamics of changes in the components of the vector competencies depending on the iteration step

5. Conclusions

Thus, the main directions of development and improvement of information support for the quality control of software products can be formulated as:

- a comprehensive presentation of the issues of structuring of information within the quality control systems as well as relevant
- databases for recording results of program testing used to formalize the research task in the form of a quality control process management model during certification tests of software;
- cognitive modeling allows to identify the main factors influencing the quality management process and the causal relationships between them.

Analysis of the obtained simulation results allows one to set boundaries of possible actions and determine promising directions for finding solutions of the target factor – improvement of quality of software tools, through information and methodological support of technology certification of software products and identify the management factors that provide the required scenarios for the development of the situation (see expression (12) for an example of linear interpretation) to assess the situation and eliminate possible negative consequences from the impact of destructive factors affecting the quality of software.

References:

Abramova, N., Avdeeva, Z., Kovriga, S., & Makarenko, D. (2010). Subject-formal methods based on cognitive maps and the problem of risk due to the human factor. *Cognitive Maps, Karl Perusich (Ed.)*, 35-62. doi: 10.5772/7118

Abd-Elrahman, El-Borsaly, A. & Hassan, S. (2020). The Impact of Service Quality on Organizational Performance in the Mobile Telecommunications Sector in Egypt. *Proceedings on Engineering Sciences, 2*(1), 93-104. doi: 10.24874/PES02.01.010

Aleksic, A., Puskaric, H, Tadic, D., & Stefanovic, M. (2017). Project management issues: vulnerability management assessment. *Kybernetes, 46*(7), 1171-1188. doi: 10.1108/K-08-2016-0218

Amin, F. M. H., & Salih, N. K. (2017). New model to achieve software quality assurance in E-learning application. *International Journal of Computer Science Issues, 14(3)*, 65-69. doi: 10.20943/0121703.6569

Arsovski, Z., & Arsovski, S. (2008). The Quality Metrics of Information Systems. *International Journal for Quality Research, 2*(2), 93-100.

Arsovski, Z., Petrovic, D. R., Milanovic, I., Rankovic, V., & Kalinic, Z. (2012). Measuring the data model quality in the e-supply chains. *International Journal for Quality Research, 6*(1), 47-53.

Barabanov A., Markov A., Fadin A., & Tsirlov V. (2017). Statistics of software vulnerabilities detection during certified testing. *Voprosykiberbezopasnosti, 2*(20), 2-8.doi: 10.21581/2311-3456-2017-2-2-8 (in Russian).

Boehm, B. A. (2006). A view of 20th and 21st century software Engineering // Proceedings of the 28th international conference on Software engineering, ICSE'06, May 20-28, 2006. – Shanghai, 12-29. doi: 10.1145/1134285.1134288

Boiral, O. (2012). ISO 9000 and organizational effectiveness: A systematic review. *Quality Management Journal, 19*(3), 16-37. doi: 10.1080/10686967.2012.11918071

Buryi, A. S. (2016). *Fault-tolerant distributed systems of the processing of information*, Hot line – Telecom, Moscow.

Buryi, A. S. (2019). Quality control in the process software certification, *Legal informatics, 1*, 46-55.doi: 10.21681/1994-1404-2019-1-46-55.

Buryi, A. S., Polous, A. I., & Shlyakonov, V. A. (1998). A functional diagnostic method for program-controlled objects. *Automation and Remote Control*, *59*(4), Pt 2, 599-602.

CompTIA (2019). IT industry outlook 2019, [online]. Retrieved from https://www.comptia.org/resources/it-industry-trends-analysis (online 15 May, 2019).

Dordevic, A., Nestic, S., Stefanović, M., Tadic, D., Arsovski, S., Doljanica, S., & Misic, M. (2015). New service development metric in medium organizations, *9 th International Quality Conference, June 2015, Center for Quality*, 203-210.

Dugalic, B., & Mishev, A. (2012). ISO software quality standards and certification.*BCI'12, September 16-20*, 113-116.

Eda, R., & Do, H. (2019). An efficient regression testing approach for PHP Web applications using test selection and reusable constraints. *Soft Quality Journal*, *27*(1).doi: 10.1007/s11219-019-09449-2

Fishburn, P. C. (1974). Exceptional Paper – Lexicographic Orders, Utilities and Decision Rules: A Survey. *Management Science, 20*(11), 1442-1471. doi: 10.1287/mnsc.20.11.1442

Fonseca, L., & Domingues J. P. (2017). ISO 9001: 2015 edition-management, quality and value. *International Journal of Quality Research, 11*(1), 149-158. doi: 10.18421 / IJQR11.01-09

Gray, S. A., Zanre, E., & Gray, S. R. J. (2014). Fuzzy cognitive maps as representations of mental models and group beliefs: theoretical and technical issues. *In Fuzzy Cognitive maps for Applied Sciences and Engineering - From fundamentals to extensions and learning algorithms Ed: Papageorgiou, E. I. Heidelberg: Springer*, 29–48.

Ivanyuk, V. A., Andropov, K. N., & Tsvirkun, A. D. (2014). Methodology comprehensive prediction of assets and risks. *Fundamental research, 12*-5, 1032-1035. (in Russian)

Kakouris, A. P., & Sfakianaki, E. (2019). Motives for implementing ISO 9000 – does enterprise size matter? *International Journal of Productivity and Performance Management*, *68*(2), 447-463.doi: 10.1108/IJPPM-03-2018-0096

Klir, G. J. (1985). *Architecture of systems problem solving*. Plenum press, New York.doi: 10.1002/sres. 3850040309

Kokkinos, K., Lakioti, E., Papageorgiou, E., Moustakas, K., & Karayannis, V. (2018). Fuzzy Cognitive Map-Based Modeling of Social Acceptance to Overcome Uncertainties in Establishing Waste Biorefinery Facilities. *Frontiers in Energy Research, 6*: 112. doi: 10.3389/fenrg.2018.00112

Koteska, B., Mishev, A., & Pejov, L. (2018). Quantitative measurement of scientific software quality: definition of a novel quality model. *International Journal of Software Engineering and Knowledge Engineering, 28*(3), 407-425. doi: 10.1142/S0218194018500146

Kul'ba, V. V., & Sirotyuk, V. O. (2005). Optimal design of object-oriented databases in automatic information-control systems. I. Analysis of problem domains. *Automation and Remote Control, 66*(9), 1486-1495.

Kulivets, S. G. (2011). Modeling of conflict situations with uncoordinated beliefs of the agents involving games based on linear cognitive maps. *Automation and remote control, 72*(7),1568-1577.

Laporte, C. Y., & April, A. (2018). *Software quality assurance*. Wiley-IEEE Computer Society, Inc.[online]. Retrieved from https://www.researchgate.net/publication/320853147_ Software_Quality_ Assurance (online 15 May, 2019).

Lochmann, K. (2013). Defining and assessing software quality by quality models // Thesis was on 31.07.2013 at the Technical University of Munich. [online]. Retrieved from http://mediatum.ub.tum.de/doc/1169637/955300.pdf (online 19 May, 2019).

Lomakin, M. I., & Glushakova, E. V. (2015). Quality assessment of products as a tool to reduce information asymmetry. *Competence, 1(122)*, 46-50. (in Russian)

Lovtsov, D. A., Bogdanova, M. V., & Loban, A. V. (2018). Informatively-mathematical providing of legal regulation of the turnover of results of intellectual activity. *Legal informatics, 4*, 15-23.doi:10.21681/1994-1404-2018-4-15-23

Novikov, D. A. (2010). "Cognitive games": a linear impulse model. *Automation and Remote Control*, *71(4)*, 718-730. doi: 10.1134/S0005117910040120

Ortega, M., Pérez, M. & Rojas, T. (2003). Construction of a Systemic Quality Model for Evaluating a Software Product. *Software Quality Journal, 11*(3), 219-242.doi: 10.1023/A:1025166710988

Porter, M. E., & Heppelmann, J. E. (2015). How Smart, Connected Products are Transforming Companies. *Harvard Business Review*, *10*, 97-114.

Roco, M. C., & Bainbridge, W. S. (2013). The new world of discovery, invention, and innovation: convergence of knowledge, technology, and society. *Journal of Nanopartice Research, 15*: 1946. doi: 10.1007/s11051-013-1946-1

Sakovich, N. (2018). Top five information technology trends to watch in 2019, *SaM Solutions*, [online]. Retrieved from https://www.sam-solutions.com/blog/top-five-information-technology-trends-to-watch-in-2019/ (Accessed 20 June 2019).

Salva, S., & Regainia, L. (2019). An approach for guiding developers in the choice of security solutions and in the generation of concrete test cases. *Software Quality Journal*, *27*(2), 675-701. doi: 10.1007/s11219-018-9438-2

Sokolov, B. V., & Yusupov, R. M. (2006). Risk integrated modeling for control in complex technical-organizational systems. *Journal of automation and information sciences*, *38*(1), 36-57.

Vujovic, A., & Krivokapic, Z. (2009). Forecasting and analysis of trends in area of quality management system. *International Journal for Quality Research*, *3*(4), 299-307.

Yusupov, R. M., & Musaev, A. A. (2017). Effeciencyof information systems andtechnologies: featuresof estimation. *SPIIRAS Proceedings*, *51*(2), 5-34. doi: 10.15622/sp.51.1

Alexey S. Buryi
Russian scientific and technical centre of information on standardization, metrology and conformity assessment,
31, Bldg 2, Nahimov Avenue,
Moscow, 117418
Russia
a.s.burij@gostinfo.ru

Mikhail I. Lomakin
Russian scientific and technical centre of information on standardization, metrology and conformity assessment,
31, Bldg 2, Nahimov Avenue,
Moscow, 117418
Russia
m.i.lomakin@ yandex.ru

Alexander V. Dokukin
Russian scientific and technical centre of information on standardization, metrology and conformity assessment,
31, Bldg 2, Nahimov Avenue,
Moscow, 117418
Russia
a.v.dokukin@gostinfo.ru

Evgeny V. Morin
State regional center for standardization, metrology and testing in Moscow and Moscow region,
31, Bldg 1, Nahimov Avenue,
Moscow, 117418
Russia
morin.evg.vas76@mail.ru

Anatoly A. Strekha
Russian scientific and technical centre of information on standardization, metrology and conformity assessment,
31, Bldg 2, Nahimov Avenue,
Moscow, 117418
Russia
a.a.streha@gostinfo.ru

Mikhail I. Zlydnev
Russian scientific and technical centre of information on standardization, metrology and conformity assessment,
31, Bldg 2, Nahimov Avenue,
Moscow, 117418
Russia
a.a.streha@gostinfo.ru