Asian Journal of Information Technology 17 (3): 218-230, 2018
ISSN: 1682-3915

Qualitative Risk Analysis of Software Development

[1]Jamil Al-Azzeh, [2]Oleksandr Kovalenko, [2]Oleksii Smirnov, [2]Anna Kovalenko and [2]Serhii Smirnov
[1]Department of Computer Engineering, Faculty of Engineering Technology,
Al-Balqa Applied University, Salt, Jordan
[2]Department of Cybersecurity and Software Academic,
Central Ukrainian National Technical University, Kirovohrad, Ukraine

Abstract: In this study, a set of methods for the qualitative risk analysis of software development has been developed which has helped to resolve the contradiction that arises in the development of software and consists in neglecting software security vulnerability factors by software companies. A distinctive feature of the proposed method of qualitative risk analysis of software development is the consideration of operational risk factors, especially, the risk of failure to identify software vulnerabilities and evaluation of an arbitrary consistent non-contradictory finite set of "information bits".

Key words: Risk analysis, software development, methods, non-contradictory, evaluation, risk factors

INTRODUCTION

The analysis of the literature (Alharbi and Qureshi, 2014; Araszkiewicz, 2016; Boehm, 1988; Britkin, 2007; Dorensky, 2014a, b; Geymayr and Ebecken, 1995; Hijazi *et al*., 2014; Kovalenko, 2014; Kovalenko, 2016a-f; Krishnan, 2015; Li, 2014; Lysenko, 2014a-c; Moran, 2014; Nogin, 2007; Power, 2014; Schwindt and Zimmermann, 2015; Shapkin and Shapkin, 2005; Smirnov *et al*., 2013; Tavares *et al*., 2017; Tomanek and Juricek, 2016; Tomanek *et al*., 2014; Vishnyakov and Radaev, 2008; Zeng, 2010) and conducted studies have shown that the overall sequence of risk analysis most often includes the following actions:

- Identification of sources and causes of the risk of software development, stages and work under which the risk arises
- Identification of all potential risks typical to the project at hand
- Documentation of results and their subsequent prioritization
- Estimation of the level of individual risks and the risk of the project as a whole which determines its economic expediency
- Definition of the acceptable risk level of software development
- Development of measures for risk reduction

In accordance with the given algorithm, the risk analysis is divided into three directions that complement each other: qualitative (steps 1-3) and quantitative analysis (steps 4, 5) of software development risks as well as management (step 6). This study is devoted to the development of the method for qualitative risk analysis of software development.

The conducted research has shown that the method of qualitative risk analysis of the project is descriptive and it is a process aimed at identifying specific project risks as well as the causes that generate them with a subsequent analysis of possible consequences and development of countermeasures. In the process of qualitative risk analysis, the development of metrics responsible for determining the boundary indicators of the factors that symbolize the manifestation of the risk's occurs.

Problems of analysis and assessment of information activity risks: The general processes of globalization of economic, financial, social and information relations contributed to the development of risk management. However, global financial crises have shown insufficient attention to risk management by the majority of leadership representatives from various organizations including Ukrainian ones.

Currently, most organizations and enterprises of various forms of ownership are increasingly focusing on issues of analysis and estimation of risks. But

Corresponding Author: Jamil Al-azzeh, Department of Computer Engineering, Faculty of Engineering Technology, Al-Balqa Applied University, Salt, Jordan

despite this problems and issues related to the general theory and methodology of analysis, estimation and management of risks require adaptation to the approaches and provisions of modern management with taking into account new factors in the formation and development of technology, combining well-known "established" risk theory provisions with new progressive approaches of analysis and synthesis.

The analysis of literature (Moran, 2014; Tomanek *et al*., 2014; Araszkiewicz, 2016; Schwindt and Zimmermann, 2015; Hijazi *et al*., 2014; Li, 2014; Krishnan, 2015; Zeng, 2010; Britkin, 2007; Vishnyakov and Radaev, 2008; Shapkin and Shapkin, 2005; Boehm, 1988; Ishikawa, 1988; Nogin, 2007; Geymayr and Ebecken, 1995; Smirnov *et al*., 2013; Dorensky, 2013, 2014a, b; Lysenko, 2014a-c; Kovalenko, 2014, 2016a-f) has shown that, despite rather deep history of the development of the concept of "risk" and the attempts of a number of well-known researchers to concentrate their developments in the field of risk management of certain industries and activities the development of new promising scientific provisions in this area is somewhat "narrowed" by financial activities. At the same time, the widespread use of information technology in our research requires increased attention to this area and, consequently, a deeper coverage of the IT industry risk management issues.

The essence of any process, phenomenon or object (including the information component) is the activity that leads to the formation of results. In such area of activity as software development the end result in most practical cases is the fulfillment of customer's requirements and implementation of the developed product. Modern researchers (Moran, 2014; Tomanek *et al*., 2014; Breno *et al*., 2017; Araszkiewicz, 2016; Schwindt and Zimmermann, 2015; Hijazi *et al*., 2014; Li, 2014; Krishnan, 2015; Zeng, 2010; Britkin, 2007; Vishnyakov, 2008; Shapkin and Shapkin, 2005; Boehm, 1988; Nogin, 2007; Geymayr and Ebecken, 1995; Smirnov *et al*., 2013; Dorensky, 2013 often reduce the result of an estimated risk to a negative effect type forgetting that even the term "risk" itself originated from the french word "risque" or Italian "risico". It means the possibility or probability of occurrence of events with specific consequences as a result of certain decisions or actions. The expediency of such representation of concepts in risk theory is especially, emphasized by the consistent patterns that arise in the information relations during software development where the complexity and dynamics of interrelations the unclearness of external factors as well as heterogeneity in the structural and functional construction of systems allows the classification of the results of information activity to be extended to the type presented in Fig. 1.

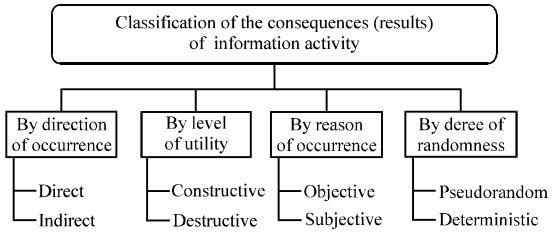

Fig. 1: Classification of the results of information activity

It should be noted that the objective result is the consequence of a purposeful and explicit execution of the process which is related to its essence. Subjective results are manifested when the process is carried out with an insufficient level of certainty and completeness of information. In practice in the IT industry, the majority of risks are associated with the subjective results of a realized step or an executed process.

The necessary information is obtained due to the availability of clear and specific (standardized, tested, regulated, etc.) assets, tools, methods and techniques the implementation of which is associated with resource costs as well as the lack of reliable data about the purpose and essence of the process under study.

Thus, it can be noted that all risks in the development of software with more or less admission can be considered a subjective result of the process which is associated with the lack of quantitative or qualitative information about the process as well as its uncertainty. Afore-referenced factors can be considered the main reason that generates and accompanies risks during their entire life cycle. Each risk of a software development cycle can be associated with one of the following: data, human and system. At the same time, it is necessary to take into account the degree of influence and responsibility of the results of risk analysis for different software development methodologies.

The analysis of literature (Moran, 2014; Tomanek *et al*., 2014; Araszkiewicz, 2016; Schwindt and Zimmermann, 2015; Hijazi *et al*., 2014; Li, 2014; Krishnan, 2015; Zeng, 2010; Britkin, 2007; Vishnyakov, 2008; Shapkin and Shapkin, 2005; Boehm, 1988; Nogin, 2007; Geymayr and Ebecken, 1995; Smirnov *et al*.,

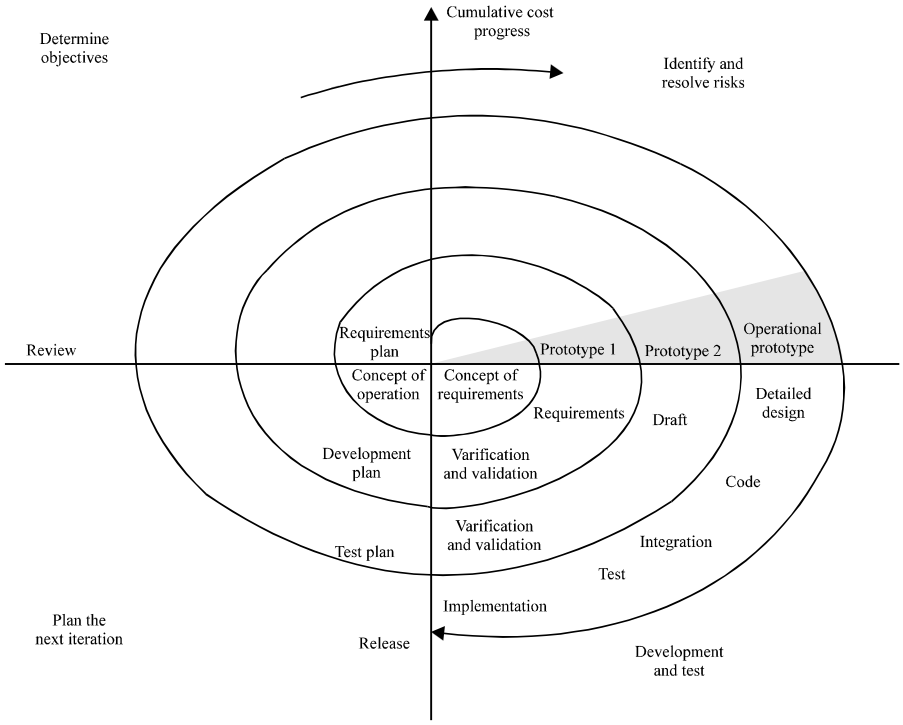

Fig. 2: Spiral model of software development

2013; Dorensky, 2014a, b; Lysenko, 2014a-c; Kovalenko, 2014, 2016a-f) has shown that, currently, there are many $R = \{x_1, ..., x_n\}$ different models of software development. It should be noted that the choice of a process model in the project implementation has a significant impact on the results of risk analysis assessment and management. For example, it is known from the literature (Vishnyakov, 2008) that one of the widely used software development models is the spiral model. Proposed in 1988 by the American specialist Barry Boehm (Boehm, 1988), this methodology is guided by incremental developments based on risks (Fig. 2).

As can be seen from Fig. 2, more than 15% of the IT-project management time is spent on risk analysis and estimation. It should be noted that at each turn of the "spiral" this task has its own peculiarities and limitations that affect the process of risk management in the system.

The analysis of literature (Moran, 2014; Tomanek *et al*., 2014; Araszkiewicz, 2016; Schwindt and Zimmermann, 2015; Breno *et al*., 2017; Hijazi *et al*., 2014; Li, 2014; Krishnan, 2015; Zeng, 2010; Britkin, 2007; Vishnyakov, 2008; Shapkin and Shapkin, 2005; Boehm, 1988; Nogin, 2007; Geymayr and Ebecken, 1995; Smirnov *et al*., 2013; Dorensky, 2014a, b; Lysenko, 2014a-c; Kovalenko, 2014, 2016a-e) has shown that modern researchers in their majority identify five major risks: errors inherent in the schedule, the emergence of new requirements, the change of staff, the decomposition of the specification, low productivity.

The conducted researches have shown that this position is controversial, since, it does not take into account a number of important aspects of software development. The analysis of regulatory documentation of a number of well-known software companies has shown that, at the risk estimation stage as a rule, risks associated with possible errors in models, algorithms information processing programs that are used to develop management decisions are not taken into account and security risks are neglected (possible errors affecting the vulnerability of the software). This often leads to errors and accordingly, unjustified losses (temporary, economic, image, etc).

Thus, the conducted studies have shown that despite the importance of solving the problem of risk management in software development at the moment there is no well-formed, standardized methodological base for describing this process. Currently we can see:lack of a single, comprehensive and systematic approach to the problem of risks in software development, lack of

clarity and transparency in understanding the final results of risks impact and their inadequate accounting in software development, significant discrepancies in understanding the methods of analysis, evaluation and risk management and insufficient accounting of important factors that arise as software development tools and technologies are improved.

The analysis of literature (Moran, 2014; Tomanek *et al*., 2014; Araszkiewicz, 2016; Schwindt and Zimmermann, 2015; Hijazi *et al*., 2014; Li, 2014; Krishnan, 2015; Zeng, 2010; Britkin, 2007; Vishnyakov, 2008; Shapkin and Shapkin, 2005; Boehm, 1988; Nogin, 2007; Geymayr and Ebecken, 1995; Smirnov *et al*., 2013; Dorensky, 2014a, b; Lysenko, 2014a-c; Kovalenko, 2014, 2016a-e) and conducted studies have shown that the overall sequence of risk analysis most often includes the following actions:

- Identification of sources and causes of the risk of software development, stages and work under which the risk arises
- Identification of all potential risks typical to the project at hand
- Documentation of results and their subsequent prioritization
- Estimation of the level of individual risks and the risk of the project as a whole which determines its economic expediency
- Definition of the acceptable risk level of software development
- Development of measures for risk reduction

In accordance with the given algorithm, the risk analysis is divided into three directions that complement each other: qualitative (steps 1-3) and quantitative analysis (steps 4, 5) of software development risks as well as management (step 6).

Let's investigate the methods of qualitative and quantitative analysis of software development risks in more detail.

MATERIALS AND METHODS

The conducted research has shown that the method of qualitative risk analysis of the project is descriptive and it is a process aimed at identifying specific project risks as well as the causes that generate them with a subsequent analysis of possible consequences and development of countermeasures. In the process of qualitative risk analysis, the development of metrics responsible for determining the boundary indicators of the factors that symbolize the manifestation of the risk(-s) occurs.

Identification of sources and causes of the risk of software development, stages and work under which the risk arises: Considering the first item of the above list of actions for qualitative and quantitative risk analysis, we note that the initial data for identifying and describing the characteristics of risks can be taken from different sources:

- Knowledge base of the organization
- Information from open sources, scientific papers
- Marketing analytics
- Survey of experts, etc

A number of well-known researchers ((Moran, 2014; Tomanek *et al*., 2014; Araszkiewicz, 2016; Schwindt and Zimmermann, 2015; Hijazi *et al*., 2014; Li, 2014; Krishnan, 2015; Zeng, 2010; Britkin, 2007; Vishnyakov, 2008; Shapkin and Shapkin, 2005; Boehm, 1988; Nogin, 2007; Geymayr and Ebecken, 1995; Smirnov *et al*., 2013; Dorensky, 2014a, b; Lysenko, 2014a-c; Kovalenko, 2014, 2016a-e) having carried out research, identified the most common risks in software development. For example, DeMarco and Lister (Moran, 2014; Shapkin and Shapkin, 2005; Nogin, 2007) list their five most important sources of risk for any software development project:

- Flaws in scheduling
- Staff turnover
- Inflation of requirements
- Specifications
- Low productivity

It can be noted that this list has a generalized nature which greatly complicates the metric evaluation of the given list. Boehm (1988) in his research expands the list to 10 most common risks of a software project:

- Shortage of specialists
- Unrealistic timing and budget
- Implementing inappropriate functionality
- Development of an incorrect user interface
- Gold plating, perfectionism, unnecessary optimization and honing of details
- Continuous flow of changes
- Lack of information about external components that determine the environment of the system or are involved in integration

- Deficiencies in the work carried out by external (with respect to the project) resources
- Insufficient performance in the resulting system
- "Gap" in the qualifications of specialists from different fields of knowledge

However, this list is also not complete and unstructured. This makes it difficult to estimate the mutual influence of the given risks on each other. The risks were estimated in detail and classified by Moran (2014), Tomanek *et al.* (2014), Araszkiewicz (2016), Schwindt and Zimmermann (2015), Hijazi *et al.* (2014), Li (2014), Krishnan (2015), Zeng (2010), Britkin (2007), Vishnyakov (2008), Shapkin and Shapkin (2005), Boehm, (1988), Nogin (2007), Geymayr and Ebecken (1995), Smirnov *et al.* (2013), Dorensky (2014a, b), Lysenko (2014a-c) and Kovalenko (2014, 2016a-e). In accordance with these studies, the risks are classified according to the following characteristics:

- Environment (internal, external risk)
- Nature (economic, technical, technological)
- Industry (project, process and product risk)
- Level (from critical to insignificant risk)
- Branch of impact (risk of not fulfilling the project budget, plan and/or quality)
- Risk management link (risk of a separate process, project risk, company risk)

However, such a classification emphasizes projects for the development of software systems that are not related to the processes of their further implementation and adaptation in the conditions of a specific organization and operation in the context of possible external malicious influences. Therefore, it is expedient to consider separately:

- The organizational risks which are related to the fact that the project will cause such changes in the structure and business processes of the company that offset the planned benefits
- The operational risks associated with uncontrolled growth in operating costs of the system
- The social risks associated with inadequate behavior of project participants
- Operational risks associated with possible future financial, image and other losses in case of potential project vulnerabilities

Method of structural identification of risks of software development: Using the results of the research of the above researchers (Moran, 2014; Tomanek *et al.*, 2014; Araszkiewicz, 2016; Schwindt and Zimmermann, 2015; Hijazi *et al.*, 2014; Li, 2014; Krishnan, 2015; Zeng, 2010; Britkin, 2007; Vishnyakov, 2008; Shapkin and Shapkin, 2005; Boehm, 1988; Nogin, 2007; Geymayr, 1995; Smirnov *et al.*, 2013; Dorensky, 2014a, b; Lysenko, 2014a-c) expert opinions, marketing data as well as knowledge bases of such well-known companies as EPAM systems and Nix Solutions Ltd., we can identify the risks of software development and present the result in the form of a structural classification scheme in Fig. 3. As can be seen from Fig. 3, the main risks of software development can be represented in the form of an aggregate of sets of organizational Z = {Id 1, ..., Id 5}, managerial U = {Id 6, ..., Id 9}, operative Y = {Id 10, ..., Id 15}, technological, T = {Id 16, ..., Id 20}, operational E = {Id 21, ..., Id 24}, social C = {Id 25, ..., Id 27} and legal W = {Id 28, Id 29} risks.

A distinctive feature of the presented classification is the consideration of operational risks. These risks are especially, important under the conditions of an increased level of cybercrime when neglect of software vulnerabilities can lead to operational problems and often to impossibility of software operation (crash).

In addition, under the conditions of the Ukrainian legal field, there are individual cases of inadequacy and inconsistency of the actions of the state apparatus officials with the legal norms.

The experience of a number of well-known software companies (Nix Solutions Ltd., etc.) has shown that this risk factor should be taken into account when developing software, along with the factor of possible changes in Ukrainian legislation.

The influence of the risks (Fig. 3) on the main factors of success of the development, implementation and long-term operation of the software is illustrated in Fig. 4.

As can be seen from this Fig. 4, most of the considered risks of software development (organizational, operative, managerial, etc.) can have a direct impact on both the software development process and the process of its operation. At the same time for example, operational risks do not have a direct impact on the software development process. But neglecting these risks often leads to the failure of the software operation and the loss of future orders and projects (idleness of software developers). It is this factor that causes the connection between the blocks "Failure during operation of software" and "Failure during software development".

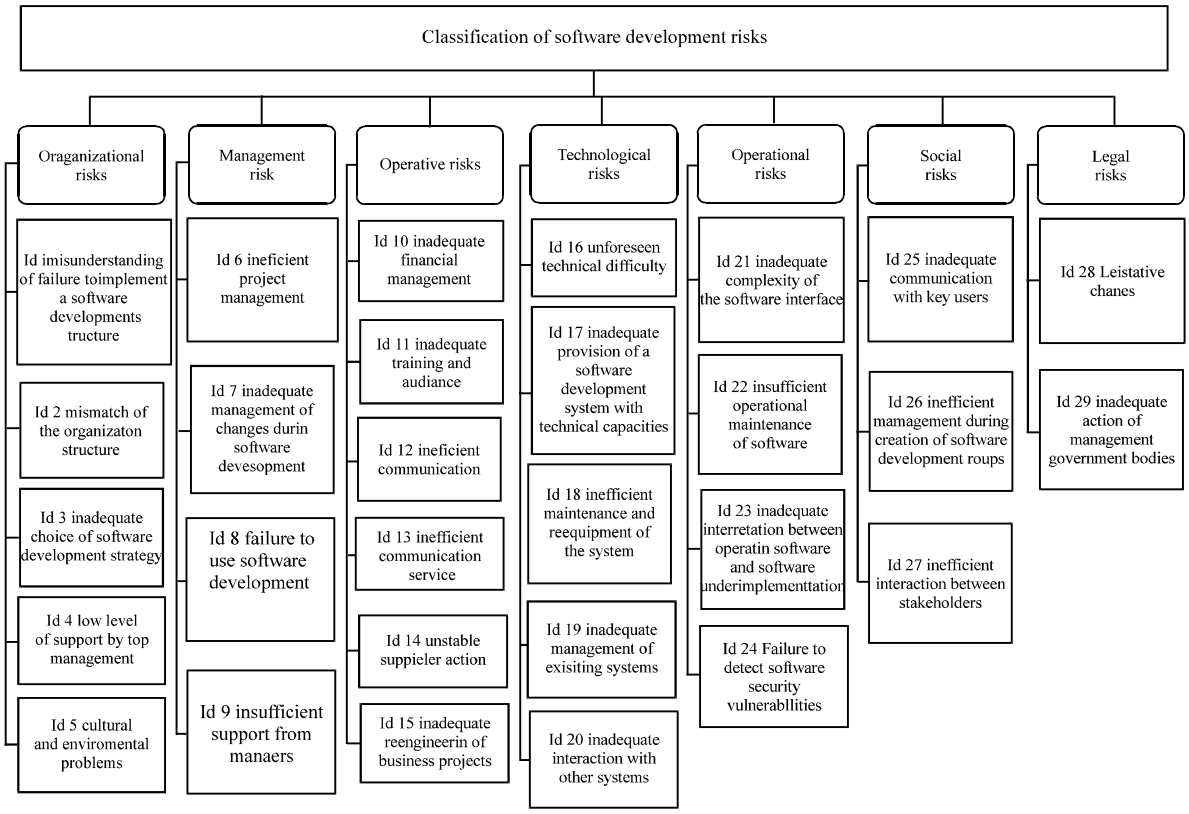

Fig. 3: Classification of software development risks

It should be noted that the factors shown in Fig. 4 sufficiently describe the list of possible risks of software development. However, they do not give an idea about mutual influence and consequently, possible change of the end result. In addition, the above-mentioned sets of software development risks affect the end result to varying degrees. Therefore, it is expedient to carry out procedures of rating and allocating the most prioritized software development risks as the next step of identifying software development risks.

However, despite this in general, we can allocate a set of risks that directly affect the software development process: MR = {Z, U, Y, C, T, W}and a set of risks that directly affect the software operation process: ME = {Z, U, Y, C, T, W, E}, (Id 9, Id 10, Id 15, Id 29∉ME).

The conducted researches have shown that to solve the problem of determining the mutual influence of risks it is expedient to use the analysis tool of cause-effect relationships between various factors and risks, developed by Kaoru Ishikawa (Ishikawa diagram). In accordance with the well-known Pareto principle from the set of potential causes (causal factors, according to Ishikawa) that generate problems (consequences), only two or three are the most significant and their search should be organized. To do this, one should:

- Collect and systemize all causes that directly or indirectly affect the problem under study
- Group these causes by semantic and cause-and-effect blocks
- Rate them inside each block
- Analyze the resulting picture

Therefore, this tool allows clarifying and taking into account all the significant factors that affect the result of software development. The use of the Ishikawa diagram allows finding out the reasons for any problems in the organization or for example, the reasons for the occurrence of operational “bugs” of software. With that the Ishikawa diagram has some advantages:

- It helps visualizing the relationship between the obtained result and the underlying causes
- It allows analyzing the chain of factors that affect the problem

The main steps of the Ishikawa diagram formation algorithm are given in Fig. 4. Using the proposed

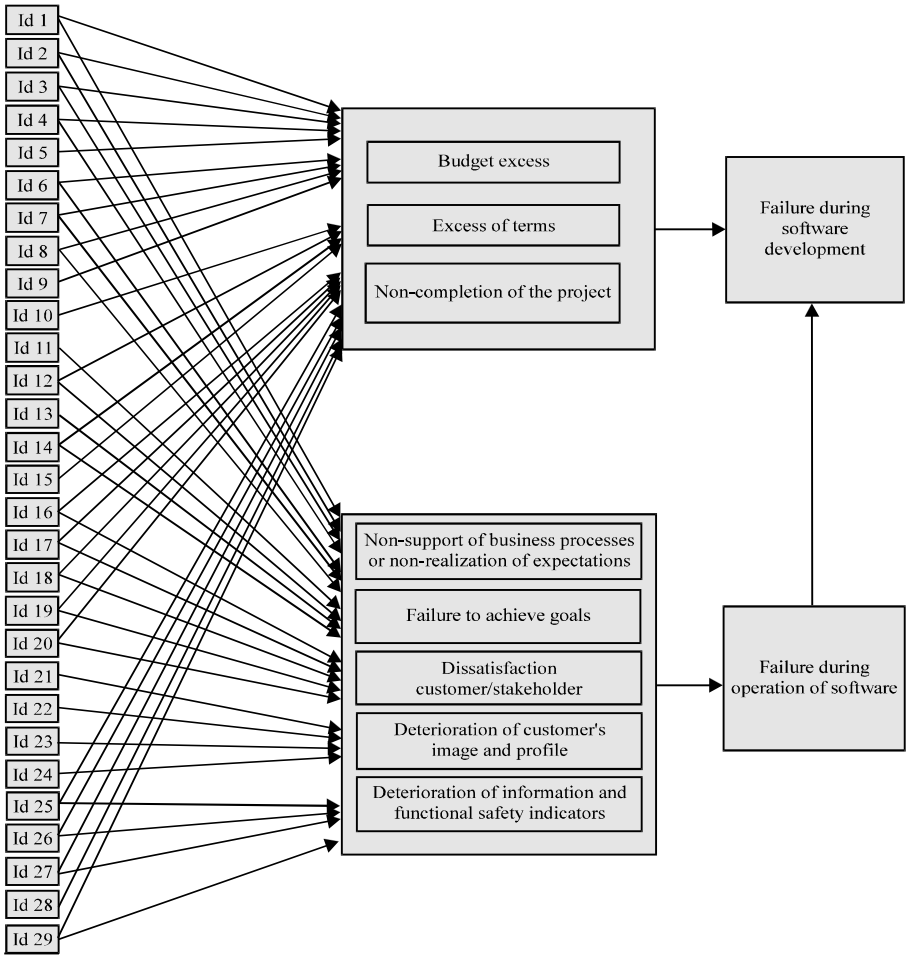

Fig. 4: The impact of risks on the main factors of development success, implementation and long-term operation of software

algorithm and taking into account the risks described above in Fig. 3 and 4, the Ishikawa diagram can be represented in the form of Fig. 5.

Representation of the Ishikawa diagram gives an opportunity to get more detailed information about the possibility of mutual influence between different types of risks which will also provide clarifying data for quantitative risk analysis. However, the diagram cannot solve the problem of choosing the most prioritized risks.

In the dissertation it is proposed to use the mathematical apparatus of multicriteria optimization based on the local geometry of the Pareto set to solve this problem.

The analysis of literature (Moran, 2014; Tomanek *et al*., 2014; Araszkiewicz, 2016; Schwindt and Zimmermann, 2015; Hijazi *et al*., 2014; Li, 2014; Krishnan, 2015; Zeng, 2010; Britkin, 2007; Vishnyakov, 2008; Shapkin, 2005; Boehm, 1988; Nogin, 2007; Geymayr and Ebecken, 1995; Smirnov *et al*., 2013; Dorensky, 2014a, b; Lysenko, 2014a-c; Kovalenko, 2014, 2016a-e) has shown that there are at least three enunciations of multicriteria optimization based on the local geometry of the Pareto set:

- Local: find one Pareto-optimal solution (closest to the given initial point)
- Global: find a finite set of Pareto-optimal solutions that sufficiently describes (covers) the true Pareto front
- Interactive: find the Pareto-optimal solution that best suits the preferences of the Decision-Maker (DM)

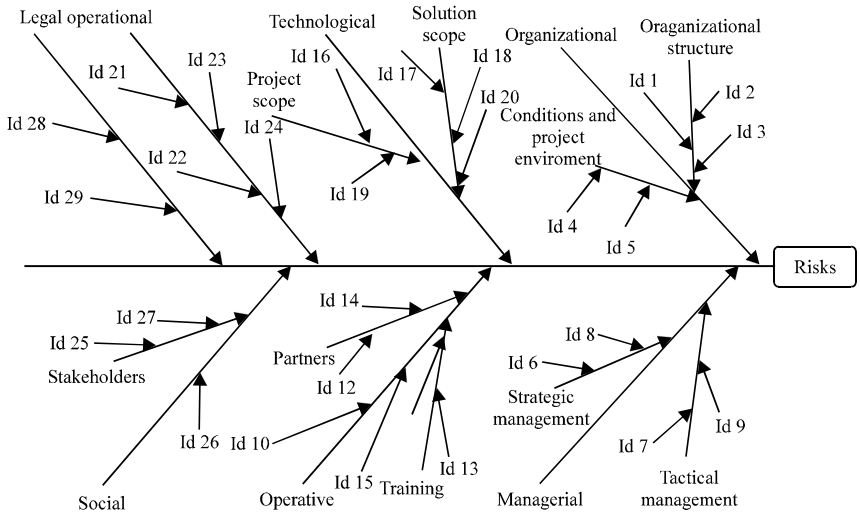

Fig. 5: Decision tree of the software development risk diagram

The conducted researches have shown that in the processes that are built on the principles of constant communication between the participants and use of "brainstorms" with expert opinions, it is expedient to use the interactive enunciation of multicriteria optimization.

In these conditions, the abstract problem of choosing the most important risks of software development from the available initial set X of possible (permissible) variants (solutions) can be enunciated as follows.

Let's denote the set of all predetermined risks of software development as S (X). Obviously, S (X)∈X. Thus, in the choice problem there is given a set X containing at least two elements and it is required to find some nonempty subset S(X). It is assumed that the choice is made by the DM whose role can be taken by an individual or a team of developers. To ensure that the choice made is the most appropriate to the achievement of the existing goal (i.e. is "the best" or "optimal" for the given DM), it is necessary to take into account the opinion of experts in the selection process.

The conducted researches have shown that currently, there are many approaches to the consideration of the expert opinion (the hierarchy analysis method (Nogin, 2007) implemented in the expert choice software, the "artificial" preference (Nogin, 2007) method, etc). However, all of them have significant drawbacks, the main of which is that despite the diversity and detailed study of hierarchies and "artificial" relationships, they can be extremely seldom considered as satisfying for a particular DM in full. A typical example that confirms this fact is the neglect of the vulnerability analysis of developed software (insufficiency or complete absence of pen-testing).

Therefore, to solve the problem of selecting the most prioritized risks (narrowing the Pareto set), it is suggested to use "information bits". To do this, let's consider arbitrary risk estimations for software development $y' = (y_1, ..., y_m)$ and $y'' = (y_1, ..., y_m)$ that belong to the set of Pareto-optimal vectors $f(P_f(X))$. By the definition of the Pareto set, there must be two such nonempty subsets of criteria numbers A, B⊂I = {1, 2, ..., m} that:

$$y_i' > y_i'', \quad y_i' - y_i'' = w_i > 0, \quad \forall i \in A \tag{1}$$

$$y_j'' > y_j', \quad y_j'' - y_j' = w_j > 0, \quad \forall j \in B \tag{2}$$

$$y_s'' = y_s', \quad \forall s \in I \setminus (A \cup B) \tag{3}$$

According to conditions (Eq. 1-3), the first vector exceeds the second by components of the criteria group A while the second exceeds the first by components of the criteria group B. For the remaining components (if any), the two indicated vectors coincide. The narrowing of the Pareto set, i.e., the removal of some Pareto-optimal vectors, usually occurs on the basis of a comparison. It's easier for a human to compare pairs. If while comparing a fixed pair of Pareto-optimal vectors y' and y" of the form (Eq.1-3), the DM "rejects" one of these vectors (for example, the second one) it means that for DM the first vector is, so, y'≻y" where ≻ is the preference relation that is defined on the whole criteria space $\Re^m$ and coincides on the set Y with relation ≻y.

The correlation y'≻y" specifies an "information bit" about the relation of strict preference which indicates the willingness of the DM to compromise he agrees to go for

losses by all the criteria of group B by the amount of w_i in order to receive increases by amount of w_j by the criteria of group while retaining the values of all the other criteria.

The presence of this "information bit" allows narrowing the Pareto set by one vector y". In order to achieve greater narrowing, it can be assumed that y'≻y" holds not only for a given pair of vectors but also for all those vectors that satisfy conditions (Eq. 1-3) with constant values of w_i and w_j.

In this case, it is suggested to say that the criteria group A is more important than group B. With the specified action expansion of the "information bit" one can expect a more apparent narrowing of the Pareto set, although it is often insufficient for the final choice. In such cases, it is expedient to impose additional requirements on the preference relation, so that, the action of the "information bit" in the narrowing of the Pareto set turns out to be more effective. These requirements (without the exclusion axiom) are formulated by Moran (2014), Tomanek *et al.* (2014), Araszkiewicz (2016), Schwindt and Zimmermann (2015), Breno *et al.* (2017), Araszkiewicz (2016), Schwindt and Zimmermann (2015), Hijazi *et al.* (2014), Li (2014), Krishnan (2015), Zeng (2010), Britkin (2007), Vishnyakov (2008), Shapkin (2005), Boehm (1988), Nogin (2007), Geymayr (1995) and Smirnov *et al.* (2013). Later it was established that they represent a further strengthening of the system of the two axioms mentioned earlier which guarantee the fulfillment of the Edgeworth-Pareto principle:

- Axiom 1: (exclusion axiom)
- Axiom 2: relation ≻ is defined on the whole criteria space and is transitive on it
- Axiom 3 (coordination axiom): of the two vectors that differ from each other by a single component, a vector with a large component is preferable for a DM

Axiom 4 (invariance axiom). The preference relation is invariant in relation to a linear positive transformation (i.e., it is linear).

Let one criterion (or a group of criteria) be more important than another criterion (another group of criteria), if there is some condition Ξ, that contains certain information about the preference relation of the DM. Hence, it is clear that one can manage without defining the importance of criteria while directly operating the condition Ξ in the decision-making process. In order to use, the importance definition based on the "information bit" and using correlations (Eq. 1-3) as the condition Ξ, first it is necessary to explain this importance definition to the DM and to make sure that he "gets" it. Then, to identify the preferences of the DM ask him a question in the "importance language": "Is the criteria group A more important than group B with parameters w_i and w_j (for i∈A and j∈B).

It is known from (Nogin, 2007) that a binary relation ≻ defined on a vector space $\Re^m$ is said to be conical if there exists such a cone K⊂ $\Re^m$ that the correlation y'≻y" holds if and only if y'-y"∈K.

Axiom 5; Any binary relation ≻ which is defined on a vector space $\Re^m$ and that satisfies axioms 2-4 is conical with an acute convex cone (without the origin) which contains all vectors with nonnegative components. Conversely, every conic relation ≻ with the indicated cone satisfies axioms 2-4.

Axiom 5 opens the possibility of using the apparatus of convex analysis and constructing a meaningful mathematical theory to take into account a different set of "information bits". The simplest case of a single "bit" is considered in the following statement, the proof of which is based on facts from the duality theory of convex analysis.

Axiom 6; Suppose that axioms 2-4 are fulfilled and there is an "information bit" about the preference relation ≻. Then for any set of selectable variants S(X) which satisfy the Axiom 1 the inclusions S(X)⊂ $P_g(X)$⊂P_f are valid and the "new" vector criterion g can be formed from the functions f_i for all i∈I/B:

$$g_{i,j} = w_j f_i + w_i f_j = \text{for all } i \in A, j \in B \tag{4}$$

or from the functions f_i for all i∈I/B:

$$f_0 = \min_{i \in A} \frac{f_i}{w_i} + \min_{j \in B} \frac{f_j}{w_j} \tag{5}$$

An important feature of Axiom 6 is the absence of any requirements for the set X and the vector criterion f-these objects can be arbitrary. Restrictions are imposed only on the behavior of the DM in the decision-making process and they are expressed in the form of Axioms 1-4.

Axiom 6 shows the upper bound $P_g(X)$ for an unknown set of selectable variants S(X) which is more precise than the Pareto set $P_f(X)$. The estimate itself is a set of Pareto-optimal variants but with respect to the "new" vector criterion g.

In order to form g, all components of the criteria group B should be deleted from the "old" vector criterion f and one non-linear criterion f_0 of the form (Eq. 5) or |A|·|B| "new" linear criteria of the form (Eq. 4) should be added where, |L| is the number of elements of the finite set L.

The variant with a non-linear function f_0 of the form (Eq. 5) can be used for quantitative criteria the values of which are measured in the relationship scale whereas option (Eq. 4) also allows the use in the interval scale. The nonlinear function f_0 of the form (Eq. 5) can be used to study the case where one criteria group is more important than the other where in contrast to the above axiomatics, a transitive closure operation of the binary relation and some other assumptions are used.

As studies have shown, taking into account several "information bits" should contribute more to the narrowing of the Pareto set. However, there may be a situation where a number of "information bits" will have a contradictory meaning and their use will be impossible. Therefore, the important task is the choice of noncontradictory "information bits". Within the framework of the dissertation a set is called consistent when it "generates" an irreflexive relation.

The construction of an upper bound for an unknown set of selectable vectors S(Y) = f(S(X)) in the form of a set $\overline{P}(Y)=f(P_g(X))$ in the presence of an arbitrary noncontradictory finite set of "information bits" in the case of a finite set Y is reduced to sequential verification of the relation

$$y' \succ_m y'' \quad (6)$$

for all pairs of admissible vector estimates y', y"∈Y where $\succ_m$ is a binary relation which is constructed on the basis of the available noncontradictory set of "information bits".

Thus, the method which differs from those known for constructing a software development risk analysis "from above" in the form of a set in the presence of an arbitrary, noncontradictory, finite set of "information bits" for structural identification of software development risks, has got further development. Using the above method, we will analyze the rank of software development risks.

Study of the developed method of structural identification of risks: Once the risks of software development are identified and included in the risk register, it becomes necessary to analyze and rank them separately for each process/project goal (for example, for a functional scope, time or other resources) and construct a probability and impact matrix (Zeng, 2010). The risk rank allows quick management of the response to risks located in different areas of the matrix. The areas of the matrix play the role of priorities.

As it was pointed out, the decision on the risk rank is influenced by the priorities of the DM that are largely formed by expert reviews or by results of "brainstorms" (typical for flexible software development models).

Taking these factors into account, we will construct a matrix of qualitative risk rank analysis of software development in accordance with the data in Fig. 4 and expert reviews by specialists from several well-known software companies (Nix Solutions Ltd., EPAM Systems) (Moran, 2014; Tomanek *et al*., 2014; Araszkiewicz, 2016; Schwindt and Zimmermann, 2015; Hijazi *et al*., 2014; Li, 2014; Krishnan, 2015; Zeng, 2010; Britkin, 2007; Vishnyakov, 2008).

Table 1 presents results of the qualitative risk rank analysis of software development. It should be noted that the areas of the matrix play the role of priorities. For example, risks located in the high-risk area (highlighted in dark gray and make up the set D) of the matrix, need preventive operations and an aggressive response strategy. For threats located in the low-risk area (highlighted in white and make up the set G) preventive operations may not be necessary if all the content of the activity is kept under control. In turn, many medium-risk threats (highlighted in light gray (set F)) require a mandatory management and response strategy.

As can be seen from Table 1, the main part of organizational, operative, managerial and operational risks is in the "shaded" area. This indicates the importance of taking these risks into account (especially in today's flexible software development models).

It should be noted that many risks (for example, Id 18 and Id 20) at the beginning of a certain activity may be in a low-ranking area and move to borderline or into more critical areas when software development gets to its milestones. At the same time, a number of existing risks, regardless of the initial rank level, can move to a more "critical" area (for example, Id 23 and Id 24, etc).

Thus, the proposed apparatus for identification and qualitative risk rank analysis of software development allows narrowing the set of important risks down to 17% and accordingly, prioritized management decisions.

RESULTS AND DISCUSSION

Documentation of results and their subsequent prioritization: The next stage in qualitative risk analysis is the documentation process. The risk analysis process should be documented throughout the life cycle of the entire project/process.The volume of documentation and

Table 1: Results of the qualitative risk rank analysis of software development

Qualitative analysis of the probability of harm	The severity of the consequences after causing harm				
	Very high severity	High severity	Average severity	Low severity	Minor severity
High probability	Id 1	Id 6, 7, 24	Id 15	Id 23, 27	Id 25
Average probability	Id 19	Id 3, 10, 16	Id 4, 8	Id 21	Id 22
Low probability	Id 9	Id 2, 17	Id 12, 14	Id 18	Id 28
Small probability	Id 26	Id 11	Id 20, 29	Id 13	Id 5

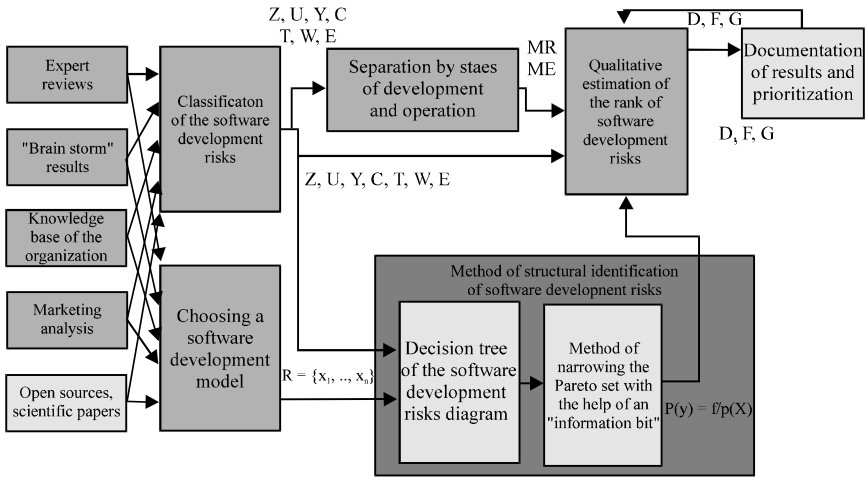

Fig. 6: General structure of the method of qualitative risk analysis of software development

its form that contains the results of the analysis depends on the specific objectives of the risk analysis that was carried out. The analysis of the documentation of well-known software companies has shown that in the final document it is expedient to record the following data:

- Title page
- List of participants in the process of qualitative risk analysis of software development
- Annotation
- Content (table of contents)
- Goals and objectives of the conducted qualitative risk analysis of software development
- Description of the analyzed object
- Method for qualitative risk analysis of software development-initial assumptions and limitations that define the limits of risk analysis
- Description of the methods of analysis used and the rationale for their application
- Initial data and their sources
- Identification results
- Results of qualitative risk analysis
- Analysis of uncertainties in the results of the risk estimation
- Recommendations for dealing with risks
- Conclusion
- List of sources of information used

Considering all the above stages of risk estimation and analysis of software development the overall structure of the method can be represented in the Fig. 6.

Thus, as a result of the studies conducted on the basis of classification and structural identification of software development risks, a method for qualitative risk analysis of software development has been developed. A distinctive feature of the developed method is the consideration of operational risk factors, especially, the risk of failure to reveal software vulnerabilities and the evaluation of an arbitrary, noncontradictory, finite set of "information bits". This will allow to narrow the number of important risks up to 17% and reduce the possible financial and image losses of software development organizations.

Only after the experience and the necessary array of data are accumulated it is expedient to proceed from qualitative risk analysis to quantitative analysis. Moreover, attention should be focused on those risks that were included in the high-risk area in the process of qualitative classification (especially with a high severity and probability of harm).

CONCLUSION

In this study, one of the contradictions that arise in the development of software has been identified and

solved and it consisted in neglecting software security vulnerability by software companies. As a solution to this problem, the use of developed methods for qualitative risk analysis of software development is proposed.

In the course of solving the task at the first stage, a method for qualitative risk analysis of software development has been developed. Its distinctive feature is the consideration of operational risk factors, especially, the risk of failure to reveal software vulnerabilities and the evaluation of an arbitrary, noncontradictory, finite set of "information bits". This will allow to narrow the number of important risks up to 17% and reduce the possible financial and image losses of software development organizations.

One of the main components of the method is the technique of structural identification of software development risks which differs from those known for constructing a software development risk analysis "from above" in the form of a set in the presence of an arbitrary, noncontradictory, finite set of "information bits".

Application of the advanced method of "Failure tree analysis" will allow improving the accuracy of quantitative risk analysis of software development up to 22%. At the same time, the application of the method of estimating the net present value of the software development project allows to view the project as a complex with taking into account security considerations and software vulnerability testing, using tools that allow overcoming the complexity, uncertainty and long-term nature of projects.

REFERENCES

Alharbi, E.T. and M.R.J. Qureshi, 2014. Implementation of risk management with SCRUM to achieve CMMI requirements. Intl. J. Comput. Network Inf. Secur., 6: 20-25.

Araszkiewicz, K., 2016. Building information modelling: An innovative way to manage risk in construction projects. Intl. J. Contemp. Manage., 14: 23-40.

Boehm, B.W., 1988. A spiral model of software development and enhancement. Computer, 21: 61-72.

Britkin, A.I., 2007. [Risks associated with the introduction of technology in software development projects (In Russian)]. Socio Econ. Tech. Syst. Res. Des. Optim., 8: 2-2.

Dorensky, O., 2014a. [Development of the theoretical bases of logical domain modeling of a complex software system (In Russian)]. Intl. J. Comput. Eng. Res., 4: 19-23.

Dorensky, O.P., 2014b. [Formalization of the process of changing the states of software objects of complex systems on the basis of the formal apparatus of fools of Moore's automats (In Russian)]. Commun., 3: 27-31.

Geymayr, J.A.B. and N.F.F. Ebecken, 1995. Fault-tree analysis: A knowledge-engineering approach. IEEE. Trans. Reliab., 44: 37-45.

Hijazi, H., S. Alqrainy, H. Muaidi and T. Khdour, 2014. Risk factors in software development phases. Eur. Sci. J., 10: 113-132.

Kovalenko, A.V., 2014. [Problems of recognition of situations in ERP systems (In Russian)]. Syst. Inf. Tech., 4: 161-164.

Kovalenko, A.V., 2016a. [Method of quantitative assessment of software development risks (In Russian)]. Collect. Sci. Works Kharkiv Univ. Air Forces, 2: 128-133.

Kovalenko, A.V., 2016b. [Method of risk management software development (In Ukrainian)]. Syst. Manage. Flowering, 2: 93-100.

Kovalenko, A.V., 2016c. [Methods for qualitative risk analysis of software development (In Russian)]. Syst. Inf. Tech., 5: 153-157.

Kovalenko, A.V., 2016d. [Problems of analysis and risk assessment of information activities (In Russian)]. Syst. Inf. Tech., 3: 40-42.

Kovalenko, A.V., 2016e. [The method of qualitative analysis of software development risks (In Russian)]. Sci. Technol. Powers Energetic Forces Ukraine, 2: 150-158.

Kovalenko, A.V., 2016f. [Use of pseudo-Boolean methods of bivalent programming for software risk management (In Russian)]. Control Syst. Tamping, 1: 98-103.

Krishnan, M. S., 2015. Software development risk aspects and success frequency on spiral and agile model. Intl. J. Innovative Res. Comput. Comm. Eng, 3: 122-129.

Li, Z.G., 2014. The Risk Management of whole Life Cycle of it Outsourcing Project. In: Advanced Materials Research, Xu, P., S. Hongzong, Y. Wang and P. Wang (Eds.). Trans Tech Publications, Zurich, Switzerland, pp: 4057-4060.

Lysenko, I.A., 2014 (b). [Investigation of software development process of infotelecommunication systems (In Russian)]. Syst. Technol., 4: 103-106.

Lysenko, I.A., 2014 (a). [Investigation of testing levels of software for infotelecommunication systems (In Russian)]. Sci. Technol. Powers Energetic Forces of Ukraine, 4: 79-81.

Lysenko, I.A., 2014 (c). [Investigation of the algorithm for identifying the type of unaccounted test cases in the design of test sets (In Russian)]. Commun., 2: 153-156.

Moran, A., 2014. Agile Risk Management. In: Agile Risk Management, Moran, A . (Ed.). Springer, Cham, Switzerland, ISBN:978-3-319-05007-2, pp: 33-60.

Nogin, V.D., 2007. Decision-Making under many Conditions: Educational and Methodical Benefits. Publishing House UTAS, St. Petersburg, Russia, Pages: 104.

Power, K., 2014. Impediment Impact Diagrams: Understanding the Impact of Impediments in Agile Teams and Organizations. Proceedings of the Conference on Agile Conference (AGILE), July 28-August 1, 2014, IEEE, Kissimmee, FL, USA., pp: 41-51.

Schwindt, C. and J. Zimmermann, 2015. Handbook on Project Management and Scheduling. Vol. 2, Springer, Cham, Switzerland, Pages: 1400.

Shapkin, A.S. and V.A. Shapkin, 2005. [Theory of Risk and Modeling of Risk Situations: A Textbook]. Dashkov and K Publishing and Trading Corporation, Moscow, Russia, Pages: 880 (In Russian).

Smirnov, O.A., O.V. Kovalenko and E.V. Meleshko, 2013. [Software Engineering: Tutorial]. RVL KNTU, Russia, Pages: 409 (In Russian).

Tavares, B.G., C.E.S. Da Silva and A.D. De Souza, 2017. Risk management in scrum projects: A bibliometric study. J. Commun. Software Syst., 13: 25-41.

Tomanek, M. and J. Juricek, 2016. Project risk management model based on PRINCE2 and SCRUM frameworks. Intl. J. Software Eng. Appl., 6: 81-88.

Tomanek, M., R. Cermak and Z. Smutny, 2014. A conceptual framework for web development projects based on project management and agile development principles. Proceedings of the 10th European Conference on Management Leadership and Governance (ECMLG), November 13-14, 2014, Zagreb, Republic of Croatia, pp: 550-558.

Vishnyakov, Y.D. and N.N. Radaev, 2008. [General Theory of Risks: Textbook]. Publishing Center Academy, Moscow, Russia, Pages: 368 (In Russian).

Zeng, Y., 2010. Risk management for enterprise resource planning system implementations in project-based firms. Ph.D Thesis, University of Maryland, Maryland, USA.