An Enhanced Risk Assessment Framework for Software Development in Multiple Projects Environment

Nagy Ramadan Darwish
Department of Information Systems and Technology
Institute of Statistical Studies and Research
Cairo University

Ahmed Ali Mohamed Warad
Department of Information Systems and Technology
Institute of Statistical Studies and Research
Cairo University

ABSTRACT

Risk management has an increasing importance in the domain of software project management because it has direct effects on the project success. Therefore, software organizations are challenged to achieve effective and efficient risk management. Measuring risks in software development is a main activity that must be performed in risk management. Risk measurement is one of the hot topics that need more efforts, especially in the environment of multiple software projects. Multiple projects environment include many projects that may share recourses. This paper conducts a structured literature review to find patterns or trends in software development for measuring project risk. It aims to reveal the aspects of risk measurement process, risk indicators, and different scales of software projects. This paper provides a proposed framework for risk assessment in multiple projects environment with shared recourses for different scales at organizations of various capacities.

General Terms

Risk assessment, Software development projects, Multiple Projects Environment.

Keywords

Software development projects, risk assessment, Indicator, Measure, Metric, large-scale project.

1. INTRODUCTION

Whereas project management addresses and manages risks through looking at the big picture in detail and planning what should happen and how to achieve success. Risk management focuses on unique risks that face each project through looking at potential problems and planning to solve them, as well as estimating, predicting and evaluating what could happen and trying to minimize the potential results through managing the risk [1]. Therefore, decision makers need risk management as well as project management in order to avoid the failure factors that may face the projects and that may lead to loss of effort and cost.

The potential problems or risks that are possibly found while managing the projects that may uses some shared resources. Therefore, the risk may be shared in multiple software projects in fact.

There are various types of risks in software projects including technical, management, financial, contractual and legal, personal, and other resources risks. However, it is a percentile value based on two different factors: organization size and project scale. Through discussion, we shall make this point clear.

This paper aims at conducting a literature review on the potential categories of risks in software development and proposing an enhanced risk assessment framework to integrate the recovery process for software projects. That would be achieved by an effective integration of risk management with project management in the environment that includes many projects, in order to help improve existing risk management practices.

This paper is organized as follows: the first section includes the introduction in hand and an overview of the risks in software development. The second section presents the related works for risk assessment. The third section deals with the risk management on software development projects. The fourth section presents an enhanced framework for risk assessment proposed in this work. Conclusion and references are given at the end.

2. RELATED WORK

There are many works reported the need for risk measurement practices in the domain of software project management. In the following the researchers briefly explain some of these works.

M. Esterman and K. Ishii [2] proposed a risk metric that combines the customization degree and dependence level between product components. When customization degree is high, it would be easy the reuse.

Sauer et al. [3] say that project size is an excellent risk indicator for IT projects, and he created a survey which verifies that there is a strong relationship between project size and its performance.

Kulk et al. [4] presented a set of quantifiable risk drivers for IT environments suggesting to measure the software. One of them considers aspects of changes in projects, such as in user requirements or changes in scope.

L. Turnbaugh [5] proposed a set of indicators, that it is important to define indicators using the schedule as the main source of information.

D. Gupta and M. Sadiq [6] proposed metrics called “Mission Critical Requirements Stability Risk Metrics (MCRSRM)”, which assesses the risk of the software product in terms of changes in requirements.

Izquierdo-Cortazar et al [7] and Hosseingholizadeh [8] showed proposals that could be helpful in reducing the unused codes due to turnover and that provides an analysis of the software product to reduce bugs just using information from source code. This kind of information could be helpful when the set of projects use the same programming language.

Singaravel et al [9] proposed a measurement of complexity in which complexity is related to the product. This paper only establishes that the complexity of a product is a risk indicator without illustrating the calculation method in detail. On the other hand, it is important to define different indicators considering also the complexity level of a project. If a project is complex, it is somewhat harder to implement because of having several types of information, such as project or product size, dependence level, function points, and so on.

Pajares and Paredes [10] proposed a metric called Cost Control Index, which is used to monitor the variability of cost during a project life cycle using Earned Value Management Analysis (EVM), they defined a measure called Schedule Control Index, which allows the schedule to estimate and monitor integrated to EVM analysis.

Wu [11] performed an empirical study, which shows that some risk factors are reduced when the organizational maturity is high. The level of maturity of an organization or process is used to indicate the organizational risk.

Souza et al [12] proposed metric to support the prioritization of test cases based on the risk exposure of requirements. They define the number of test cases and change requests as a source of information to measure technical risks once the test coverage works better when considering risk analysis before testing process.

Sultan et al [13] mentioned a number of errors. This kind of information seems to be relevant, especially to address product risks according to a database of errors in past similar products. He suggested a number of requirements such as information about product size.

Wang et al [14] suggested the combination between the probability of risk factors and concepts of Utility Theory, proposing a function called Utility of Risk Factor, which basically estimates the performance of a project.

Fiedler [15] discusses risk factors related to resistance due to changes in organizations. In this paper, some indicators refer to job satisfaction as well as the rate of sick leave and turnover.

Sharif [16] discussed and studied the existing software risk assessment from the previous studies during the last ten years and focused on the gap of SMSDP risk assessment in the research field, and the need to resort to different directions to solve that problem.

Basri [17] studied and analyzed the software risk assessment in the context of SME companies. That paper aims at identifying the impact of factors according to the analyzed approaches. In addition, this paper proposed a risk assessment conceptual framework for SMEs software development projects to be more effective and integrated.

Jr [18] presented an application of the metric “Risk Points” and its variations in a multiple software development projects to evaluate the proposed metrics as a decision-support tool and monitoring risk during the life-cycle of the project.

Menezes, Júlio et al [19] developed a systematic mapping study to define and propose indicators to support risk assessment activities– in environments of multiple software projects to provide a set of categorized indicators, that allow providing initial ways in the designing of this model through the availability and use of information that indicate risks.

Shehzad, Awan, Lali, and Aslam [20] conducted an extensive systematic literature review to achieve high validity to identify software risk list for outsourced and in-house development. The aim of this research is to explore the fundamental reasons of software failure. The authors have found new patterns to identify the causes of failure in software projects and have identified thirty-seven different risks of in-house and thirty-nine risks of outsourced software projects. It also assists them to introduce better mitigation techniques for each identified risk. This study is considered an asset to improve the paradigm of risk management in software development.

Vahidnia, Tanrıöver, and Askerzade [21] developed a risk assessment method and a prototype tool originally based on data that gather further data as the tool. A long with a risk prioritization and risk matrix, and a fuzzy method-based tool to facilitate the risk assessment by factors and their consequences in form of failure mode analysis. The method utilizes fuzzy logic to provide the practitioners with predicted scores for potential failure types and the aggregated risk score for the project. The authors have conducted a case study for the project risk assessment in a small software organization. The research aims at presenting a method and a prototype tool to support software development practitioners and teams with risk assessment processes in a small software organization.

Hsieh, Hsu, and Lin [22] developed a fuzzy risk impact rating (FRIR) for determining the total project risk exposure level for an NSD project according to risk attributes associated with the project using fuzzy weighted average. It discussed fuzzy logic-based risk evaluation model and how it can efficiently aid managers in dealing with issues like ambiguity, imprecision, and complexity in NSD risk evaluation.

3. RISK CATEGORIES AND THE PROJECT SCALE

The scale of the project can be estimated according to two aspects: (1) the number of requirements and (2) the number of interfaces. Both of them combined with the information about software, such as presented in Function Points. When software is bigger, there are more associated risk factors. Thus, a bigger system could require an increased attention. The large-scale means software requirement complexity level and requirement dependence, difficulty in implementing requirements set, especially if it is considered as a priority, in an environment of multiple projects, that is more reusable components and the sharing of software resources in the environment. The second kind may allow the review of related factors, such as quality of specification of external and internal interfaces, difficulty level to integrate components and maintenance.

Note, the number of programming languages indicator is more related to the number of interfaces of the product. Therefore, development of software products that include mixed technologies, environments or frameworks needs to use different programming languages. This indicator allows more understanding about interfacing points within the product, collaborating for optimizing the software development activities. Test process enables early identification of bugs, so it prevents bigger problems, due to the bug. In this case, problems are more easily identified.

Measuring how much critical or complex is a project may allow to take decisions about the feasibility of a project.

Critical projects need more careful management rather than others. Once it is supposed to have many restrictions, especially of cost and time, project management indicators like CPI (Cost Performance Index) or SPI (Schedule Performance Index) or product metrics like “number of transactions” can also help in identifying risks.

Regarding complexity, the same happens: if the project is difficult to develop. In order to reduce the levels of complexity, the researchers can consider measures of complexity: size of a Work-Breakdown Structure, Number of Dependence between projects or systems and Function Points or Story Points for Scrum-based projects.

Risk Points [19] aim at assessing the complexity of a software project based on a number of identified aspects to measure the risk level of a project. Financial Feasibility is helpful to perform an ante project risk assessment, verifying if the estimated cost is really feasible.

This indicator also is helpful to reallocate resources to projects considered as a priority for the organization. External dependence level brings information about project external dependencies, such as products or services. It helps in minimizing or avoiding risks, especially related to outsourcing. Business value tries to identify whether there are similar products on the market or not.

The idea is to help stakeholders to maintain the project value to an acceptable level. Indicators about a client are described in Client Experience Level and Client Participation Level. They indicate risks about the client involvement and experience, so that it would be helpful in mitigating risks due to the client, enabling action plans in order not to prejudice the project.

Multiple Project Management [18] currently is a reality in software development environments. In the case of software projects, some features are emphasized, such as persistent changes in levels of scope or product, software complexity and aspects related to human resources, such as technical knowledge and experience, among others. The researchers may consider these characteristics as risk factors that should be managed.

In this aspect, tactical management presupposes the usage of better-structured information, which directed us to support the usage of a metric-based strategy for multiple project managers with a focus on risk factors. In this context, this work presents an application of the metric “Risk Points” and its variations in an environment of multiple software development projects. This experience report aims at evaluating the proposed metrics as a decision-support tool and monitoring risk during the project life-cycle.

4. THE PROPOSED FRAMEWORK FOR RISK ASSESSMENT

A risk assessment framework is an approach to prioritizing and sharing information about the taxonomy risks posed to a software development organization. The information presented in different ways that refer to shared resources. The risk assessment framework proposed provides assistance to organizations in identifying the risk areas in the project.

The aim of the paper is to present an effective and practical risk assessment framework on a much larger scale project in a big organization. The Proposed Framework assumes that risk is a probability of loss. In this sense, risk is related to the quality of software development process where cost and effort could be minimized.

In this section, the researchers present a risk assessment framework for risk based on characterizing and prioritizing the identified risks purposes in order to help in making decisions through the shared resources. This framework is shown in Fig. 1. It displays a risk assessment model, which configures the risk-based categories according to characterizing and prioritizing risks through shared resources.

The execution of the process of characterizing and categorizing risks provide feedback to continuously refine and improve the risk assessment model. As mentioned in the previous section, the risk-based characterization and prioritization through the shared recourses process integrates risk assessment into the software development and uses the same resources to support all software development phases.

The risk assessment framework and its elements, therefore, determine the overall risk-based categories and prioritize process and are the two main phases of the proposed risk assessment framework for software development for large scale: (1) The elicitation phase. (2) The deployment phase.

Phase 1: Studying existing risks and classifying them into categories. Categories help identifying additional risks and may vary from project to project, by risk taxonomy and organizing the best practice to derive the risk identification rule. The first one aims at raising all the existing possibilities and events in a project or organization. To identify risks, the following techniques can be used: brainstorming, work breakdown analysis, risk breakdown structure, checklists, among others.

Phase 2: Deploying the risk identification rule to assess risks in current projects to characterize and prioritize the identified risks in order to help in making decisions. Risk Analysis describes the most important aspects of risks, aiming at distinguishing the best mitigation strategies. It strengthens the view of risk identification. Identifying the risks help to determine the frequencies and consequences and establish a big picture and identify the losses that might be happening and evaluate the risk and decide on precautions, proposed risk-reducing measures, and assess alternative risk-reducing measures to record the significant findings and update according to the priority of them.

In this process, the risks are usually categorized and prioritized, according to some specific criteria. In order to focus on the risk management on the main risk factors that are related to the organization. It also determines risk drivers and performs estimations, commonly through analysis of probability – likelihood – and impact – consequences of an event in different project and shard resources.

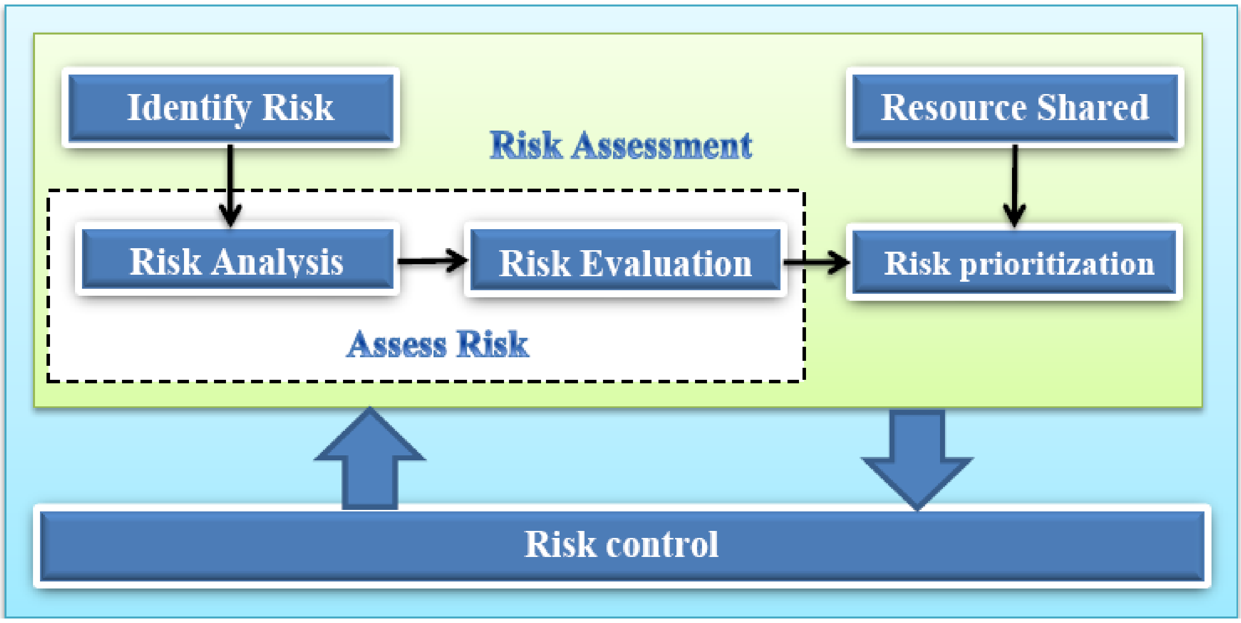

Fig 1: Proposed Framework for Risk Assessment

Fig. 1 shows the proposed theoretical research framework that combines the main two-phase constructs. Then, researchers discussed all aspects of risk assessment framework, which will add effective and integrated nature to risk assessment framework: -

4.1 Identification of Risk

Identify risk and threats, identify hazardous events, determine frequencies and consequences, establish picture and Evaluate risk (against risk acceptance criteria), propose risk-reducing measures, and assess alternative risk-reducing measures in Risk control that Make decision-related to risk-reducing measures, implement measures monitor effects, and communicate risk

Identification of risk involves a systematic process of studying situations and providing solutions using some practices such as group discussions and brainstorming sessions to generate a variety of ideas. While all the ideas or issues generated may or may not be relevant, it is important to document all problems (risks), possible impacts and solutions recognized.

Risks can be identified by providing records of previous activities or events and categories according to SEI Taxonomy [19].

4.2 Analyzing Risks

This step determines and addresses the impact of the risks that have been documented. The risks identified are rated according to the probability of occurrence. The potential of an identified risk can be estimated by categories and the effect it has on resources of the organization. When analyzing a risk, one decides on the relationship between the possibility of a risk occurring and the costs of the risk identified. Then we define the level of risk in order to set priorities and manage it, then it is explored according to the existing assets or making an investment in new resources.

Qualitative techniques for risk presentation, like Risk Points which the researchers referred to previously, can provide better definition to the nature of a certain risk, to display different risks once they have been analyzed using Risk Matrix tool.

4.3 Evaluate Risks

In this step, the degree of acceptable risk is determined; the identified risk may be acceptable or unacceptable. The evaluation takes into account the following:

An acceptable risk is a type of risk that does not have a major impact on project and organization. An acceptable risk has to be a less degree of priority. But an unacceptable risk is a main high degree of priority for the project and organization, the losses cannot be accepted. In such an event, it is important to address and treat the risk in a suitable method.

4.4 Prioritization of Risks

In this step, we need to order and organize risks by the degree of effect on the project and the resources of the organization. Accordingly, we can deal with risks in several ways; it can be avoided, reduced, shared or retained. The risk is avoided when suitable decisions are taken to remove all possible drawbacks in that way avoiding the situation from occurring. In most decision-making processes, measurements are made to achieve a balance between the cost and effect.

4.5 Risk Control

Every organization, regardless of its size, clearly attempts to reduce the risks involved. In order to reduce risk, organizations have to align their rules and structures in a constant method. In addition, there is a need to have shared resources (financial, human resources, technology etc.) efficiently on different levels of the organization. It is important to seek advice of the suitable and competent consultants when a risk is identified. Risk control is the most important method selected to deal with risks and reduce their impact.

5. CONCLUSION

Risks are present in software development projects because of the nature of software development that is based on knowledge and new technologies. As a result, the researchers have closely investigated risks and areas influenced by such risks in software development projects for large-scale in big organizations. Throughout this paper, the researchers propose a framework of risk assessment in multiple projects environment with shared recourses for different scales in organizations. The researchers have proposed two risk assessment phases: the first is the elicitation phase and the second is the deployment phase. This paper proposed categories, shared resources, and priorities-based framework to enhance risk assessment as an attempt to control risk impact as early as possible in the development process to increase project opportunities for success. In the future, the researchers plan to provide a set of guidelines on how to configure the risk assessment framework to achieve additional support to specialists. In addition, the researchers will perform empirical case studies to further evaluate and improve the proposed framework.

6. REFERENCES

[1] Nagy. R. Darwesh. Mokhtar B. Riad, Sherif A. Mazen, “A Proposed Model for Calculating Performance Indicators of Information Systems Project Management,” Egypt. Informatics J., vol. 6, no. 2, 2005.

[2] M. Esterman and K. Ishii, “The Development of Project Risk Metrics for Robust Concurrent Product Development (CPD) across the Supply Chain,” Concurr. Eng., vol. 13, no. 2, pp. 85–94, Jun. 2005.

[3] C. Sauer, A. Gemino, and B. H. Reich, “The impact of size and volatility on IT project performance,” Commun. ACM, vol. 50, no. 11, pp. 79–84, Nov. 2007.

[4] G. P. Kulk, R. J. Peters, and C. Verhoef, “Quantifying IT estimation risks,” Sci. Comput. Program., vol. 74, no. 11–12, pp. 900–933, Nov. 2009.

[5] L. Turnbaugh, “Risk Management on Large Capital Projects,” J. Prof. Issues Eng. Educ. Pract., vol. 131, no. 4, pp. 275–280, Oct. 2005.

[6] D. Gupta and M. Sadiq, “Software Risk Assessment and Estimation Model,” in 2008 International Conference on Computer Science and Information Technology, 2008, pp. 963–967.

[7] D. Izquierdo-Cortazar, G. Robles, F. Ortega, J. M. Gonzalez-Barahona GSyC, and L. Universidad Rey Juan Carlos Madrid, “Using Software Archaeology To Measure Knowledge Loss in Software Projects Due To Developer Turnover *.”

[8] A. Hosseingholizadeh, “A source-based risk analysis approach for software test optimization,” in 2010 2nd International Conference on Computer Engineering and Technology, 2010, pp. V2-601-V2-604.

[9] G. Singaravel, V. Palanisamy, and A. Krishnan, “Overview analysis of reusability metrics in software development for risk reduction,” in 2010 International Conference on Innovative Computing Technologies (ICICT), 2010, pp. 1–5.

[10] J. Pajares and A. López-Paredes, “An extension of the EVM analysis for project monitoring: The Cost Control Index and the Schedule Control Index,” Int. J. Proj. Manag., vol. 29, no. 5, pp. 615–621, Jul. 2011.

[11] Chun-Hui Wu, “Exploring impacts of software development process maturity on project risk,” in 2008 IEEE International Conference on Industrial Engineering and Engineering Management, 2008, pp. 1033–1037.

[12] E. Souza, C. Gusmao, K. Alves, J. Venancio, and R. Melo, “Measurement and control for risk-based test cases and activities,” in 2009 10th Latin American Test Workshop, 2009, pp. 1–6.

[13] K. Sultan, A. En-Nouaary, and A. Hamou-Lhadj, “Catalog of Metrics for Assessing Security Risks of Software throughout the Software Development Life Cycle,” in 2008 International Conference on Information Security and Assurance (ISA 2008), 2008, pp. 461–465.

[14] J. Wang, W. Lin, and Y.-H. Huang, “A performance-oriented risk management framework for innovative R&amp;D projects,” Technovation, vol. 30, no. 11–12, pp. 601–611, Nov. 2010.

[15] S. Fiedler, “Managing resistance in an organizational transformation: A case study from a mobile operator company,” Int. J. Proj. Manag., vol. 28, no. 4, pp. 370–383, May 2010.

[16] A. Sharif and S. Basri, “A Study on Risk Assessment for Small and Medium Software Development Projects,” Int. J. New Comput. Archit. their Appl., vol. 1, no. 2, pp. 325–335, 2011.

[17] S. Basri, A. M. Sharif, and A. S. Hashim, “SME software risk assessment : a conceptual framework,” J. Sci. Res. Dev., vol. 2, no. 13, pp. 42–51, 2015.

[18] J. M. Jr., M. Wanderley, C. Gusmão, and H. Moura, “Application of Metrics for Risk Management in Environment of Multiple Software Development Projects,” in Proceedings of the 18th International Conference on Enterprise Information Systems, 2016, vol. 2, pp. 504–511.

[19] J. Menezes, C. Gusmão, and H. Moura, “Defining Indicators for Risk Assessment in Software Development Projects,” Clei Electron. J., vol. 16, no. 10, pp. 1–24, 2013.

[20] B. Shehzad, K. M. Awan, M. I.-U. Lali, and W. Aslam, “Identification of Patterns in Failure of Software Projects,” J. Inf. Sci. Eng., vol. 33, no. November, pp. 1465–1479, 2017.

[21] S. Vahidnia, Ö. Tanrıöver, and I. N. Askerzade, “An Early Phase Software Project Risk Assessment Support Method for Emergent Software Organizations,” Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 5, pp. 105–118, 2017.

[22] M.-Y. Hsieh, Y.-C. Hsu, and C.-T. Lin, “Risk assessment in new software development projects at the front end: a fuzzy logic approach,” J. Ambient Intell. Humaniz. Comput., vol. 9, no. 2, pp. 295–305, 2018.