Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 10, October 2014, pg.845 – 849

RESEARCH ARTICLE

RISK MANAGEMENT in SOFTWARE PROJECTS

Nidhi Sehrawat[1], Neha Munsi[2], Mahak Jain[3]
[1]Computer Science and Engineering & Maharshi Dayanand University, India
[2]Computer Science and Engineering & Maharshi Dayanand University, India
[3]Computer Science and Engineering & Maharshi Dayanand University, India
[1] nidhisherawat90@gmail.com; [2] nehamunsi@gmail.com; [3] mahakjain140993@gmail.com

Abstract

Software development is a highly complex and capricious activity which is associated with high risks. As more and more organizations are investing in substantial resources, risk management becomes crucial. We can define risk Management as a systematic process for identifying, analyzing and controlling risks in projects or organizations. The positive impact of accepting risk management strategies on projects has led many software development organizations to appreciate its important role in the quest of cost reduction; schedule overruns decrease and improved performance. The present paper tries to answer the questions like" what are the most frequently used risk management approaches? "," What is the process of risk management? "How to apply risk management in practice?" The paper discusses about the risk management approach: risk-list, risk-action list, risk-strategy model, and risk-strategy analysis. The paper estimates the project risk with the estimation of the impact of the Residual Performance Risk on the project Performance.

Keywords— ***risk list, risk action, risk strategy model, risk strategy analysis, residual performance***

I. INTRODUCTION

Risk management strategy has been used in various fields like the national security, space exploration, nuclear reactors, construction industry and financial investment. This research paper focus on the study of various approaches of risk management applied to software development projects. Although many projects are completed with success in the last decades, but the fact cannot be hidden that a large percentage of the projects are never completed, or fails to operate effectively and efficiently ,which renders to the study of these approaches even more imperative.

The majority of the software development organizations observe risk in a different and in a random way which contributes to the increase of project development instability and ineffectiveness. Kwak and Ibbs identified risk management as the least applied scientific field among the various knowledge areas of project management. In agreement with Kwak and Ibbs study, Adams and Pinto research states that risk management has not received sufficient attention and does not appear to be broadly believed within the software engineering community. Dedolph implies that the reason for the software risk management neglecting is primarily the organizational inertia and their native confrontation to change, due to the difficulty of risk management value assessment, the lack of resources and the need for structural changes and other problems. The intricacy of most software projects and the several risk types developed during the development stages, the desertion of risk management to human intuition and initiative can sometimes be proven effective, yet remains an insufficient substitute of the constant professional and stable approach of risk management. The paper discusses about what is risk management , its elements ,processs of implementing risk management, process of identification of risk , approaches frequently used to manage the jobs and other risk related issues.

II. RISK MANAGEMENT IN SOFTWARE DEVELOPMENT

Risk management is defined as the identification, assessment and prioritization of risks monitored by coordinated and economical application of resources to minimize, monitor, and control the probability and/or impact of unfortunate events in order to maximize the realization of opportunities. Thus risk management is a series of steps whose objectives is to identify, address, and eliminate software risk items before they become either fears to successful software operation or a major source of costly rework. The software industry is fraught with failed and delayed projects, most of which far exceed their original budget.

III.RISK IDENTIFICATION APPROACHES

According to Schoenthaler and practical experience, it is indubitable that the systematic use of risk management into the project development process will have a positive impact on project and negative impact on project risk level. In an attempt to promptly identify risk, managers of software projects have been using various methods. Four of them are going to be conferred below.

1)*Ad-hoc Approach:* The first one is the Ad-hoc Approach, which provides a valuation of risks when the initial symptoms appear on the project, as well as their mitigation with unofficial way.

2)*Informal Approach:* The second approach is called Informal Approach that includes a discussion with people, who are directly or indirectly involved with the project, relating to the several risk issues that appear (or will possibly appear) and the recording and documentation of the risks for future use.

3)*Periodic Approach:* The third is called as Periodic Approach, it involves the use of repetitive procedures for the identification and specification (quantitatively and qualitatively) of the risks. Finally,

4)*Formal approach:* The fourth one is the Formal Approach for the identification of the various risks . According to this , a detailed and in-depth assessment of each risk by independent individuals is performed.

IV. RISK IMPACT DEFINITION

The software risk indicates a particular aspect of a development task, process, or environment, which, if unnoticed, will increase the probability of project failure (Lyytinen, Mathiassen and Ropponen 1998). In core, there are two ways to define risk, one is in *quantitative* and the other is in *qualitative*. The quantitative definition of risk comes from Boehm (1989):

$$RE=Prob\ (UO)*Loss\ (UO)$$

where RE refers to risk exposure, Prob (UO) is the probability of an unsatisfactory outcome, and Loss(UO) is the loss to the parties affected if the outcome is unsatisfactory. The customer, developer, user and maintainer are various software development projects classes of participants. Each of them has specific satisfaction criteria.

TABLE 1
THE DEFINITION OF "UNSATISFACTORY OUTCOME" BASED ON STAKEHOLDERS

Stakeholder	**Definition of unsatisfactory outcome**
Customers, developers	Budget overruns and failure to deliver on time
Users	Products with the wrong functionality, user-interface shortfalls, performances shortfalls, or reliability shortfalls
Maintainers	Poor-quality software

V. SOFTWARE RISK MANAGEMENT

Software risk management is one positive approach to deal with system failures. Risk management in software projects helps the practitioners focus on problematic aspects, emphasizing potential causes of failure, linking potential threats to possible actions, and facilitating a shared perception of the project among its participants . Risk management is used to identify, analyze, and tackle project portfolio risks, system development risks, requirement risks, and also in implementing risk.

A. Risk Management Approach

Risk management approaches are usually represented by using the concepts **of risk items, risk resolution techniques, and heuristics** (Lyytinen, Mathiassen and Ropponen 1998). Thus on the basis of this we identify four major types of approaches for software risk management which include **risk list, risk-action list, risk-strategy model, risk-strategy analysis**.

1) *Risk list:* Risk list approach offers a list of prioritized risk items. This list of risk items aids a project manager focus on possible source of risk. Information about appropriate resolution actions are not contained in this. Usually, this risk list is easy to use in assessing risks. Organizations can amend risk lists according to specific organizational context. Usually, risk items can be classified into two classes: system items and specific items. System items are those items occur in most software projects. Specific risk items are related to project characteristics and organizational context. Although this approach offers robust supports to help project managers assess risks, it does not help identify relevant resolution actions and does not provide a planned oversight of the risk profile and applicable strategies for action.
2) *Risk-action list:* It is an approach which offers a list of prioritized risk items with associated resolution actions. Prioritized risk items are contained in it, with one or more related resolution actions for each risk item. Risk action lists are also easy to use in assessing risks; they are easy to build; and they are easy to modify when needed. Conversely, to build the resolution actions, they require additional knowledge of the possible effects of different types of actions. Besides all, risk action list uses a heuristic to identify possible actions that will help resolve specific risks. This action list is easy to use and easily build. It is also useful in resolving specific risks. But, this approach focuses on isolated pairs of risk items and resolution actions and does not emphasize on a policy for addressing the risk profile as a whole.
3) *Risk-strategy model:* Risk-strategy model is a contingency model that relates collective risk items to aggregate resolution actions. This model combines comprehensive lists of risk items with resolution actions. This approach first extracts categories of risks to arrive at a risk profile and after that abstracts categories of actions to arrive at an general risk strategy. This approach uses a simple scale such as high or low to assess the risk profile along the risk categories. Through this way, it is possible to classify the project into one of a few possible situations. Then, for its situation, the model offers a risk strategy with several detailed resolution actions. This model is easy to use as it helps managers to appreciate risks, identify relevant actions, and build an overall understanding of the risk profile directly. But it has some shortcomings. The risk-strategy model is difficult to build and modify. The model can be built and the risk profile can be abstracted only when you completely understand the factors which influence risk profile.
4) *Risk-strategy analysis:* Risk-strategy analysis is a stepwise process.it links a detailed understanding of risks to an complete risk management strategy. It is very similar to risk-strategy model. As it offers detailed and aggregate risk items and resolution actions, but then it applies different heuristics. In this process, the involved actors such as customers, developers, managers, link risks to actions to develop an overall risk strategy. In contrast with risk-strategy model, the relationship between the aggregate risk items and aggregate resolution actions is looser. This approach is difficult to build but easier to modify than risk-strategy model because of the loosely coupled relationship between risk items and resolution actions.

B. Risk Dimensions

Barki *et al*. suggested that the software project risks consist of interrelated dimensions and their assessment should be made on many dimensions. Every dimension should be defined separately theoretically as well as empirically. The multidimensional assessment of risk can supply a clear specification for research and practical purposes.

McFarlan found three major dimensions of risk in the software development process: project size, technology experience and project structure. He also suggested that project administrators should develop a complete and combined software risk profile for every software project. Boehm proposed a software risk management framework that included the evaluation and control of risk and conducted a list with the top ten risks based on his personal professional experience. Wallace *et al*. proposed 27 software development risks that could be grouped into six dimensions such as Users, System Requirements , Project Complexity, Planning and Control , Team and Organizational Environment. Han and Huang research attempted to assess both the probability of occurrence and the influence for each risk by the respondents which determines the risk level of each dimension. In order to calculate the risk exposure (RE) for each risk item, the formula of Cooper *et al*. was adopted.

$$RE=P*C$$

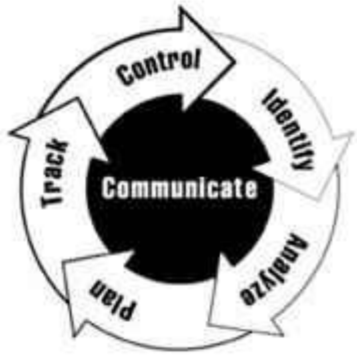
But this formula has several disadvantage such elements with high consequence, yet low possibility can return low risk exposure factors and as so they can be falsely considered as insignificant.

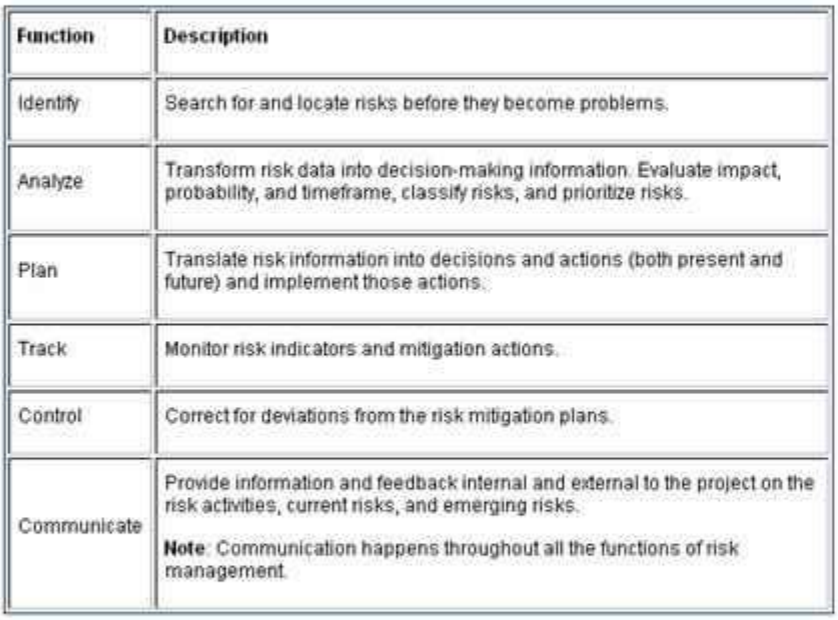
$$RE = P + C – (P*C)$$

The equation works only if the possibility of a risk to occur and the severity of its impact are in a scale from 0 to 1. The formula hopefully gives more stabilized, objective and realistic data about the significance, and hence the impact, of each risk dimension on the project.

C. Basic process of risk management

The model of risk management proposed by Software Engineering Institute (SEI) is very similar to the six steps proposed by Boehm (1991). Figure shows the model from SEI. The model demonstrates a set of functions that are identified as continuous activities throughout the life cycle of a project.





Function	Description
Identify	Search for and locate risks before they become problems.
Analyze	Transform risk data into decision-making information. Evaluate impact, probability, and timeframe, classify risks, and prioritize risks.
Plan	Translate risk information into decisions and actions (both present and future) and implement those actions.
Track	Monitor risk indicators and mitigation actions.
Control	Correct for deviations from the risk mitigation plans.
Communicate	Provide information and feedback internal and external to the project on the risk activities, current risks, and emerging risks. **Note**: Communication happens throughout all the functions of risk management.

Fig. 1 risk management process

Dorofee, et al. (1996) defined continuous risk management as a software engineering practice with **process**, **methods,** and **tools** for managing risks in a project. It allows decision makers to assess constantly what can go wrong in a project, in order to determine which risks are most important, thus implementing strategies to deal with these risks.

VI. CONCLUSION

A report in 2004 indicated that 53% of software projects failed to deliver on schedule, within budget, and with the required functions. But the studies shows that these disasters would have been avoided or strongly reduced if there had been an explicit early concern with risk management .the paper deals with risk management, risk identification approaches , risk impact definition. The paper also discusses about four types of risk management approaches. Each approach has its own advantages and disadvantages. One is approach is suitable for small projects or projects with high structure outcomes while other approach works better for projects with less people involved. After the study, I found that risk-strategy analysis approach for a complex project among those four approaches. It can be combined or used in different stages of software development as all are interlinked with each other. Software risk management is a continuous process rather than sequential activity. It emphasizes that there is a communication between different stakeholders. However, because risk management is a continuous process, each stage of development needs to consider risk management.

ACKNOWLEDGEMENT

This research paper is made possible through the help and support from everyone, including: parents, teachers, family, friends, and in essence, all sentient beings. I dedicate my acknowledgment of gratitude towards my co-Authors Neha Munsi and Mahak Jain for supporting me and guiding me.

REFERENCES

[1] Hughes &Cotterell, Software Project Management, 4 ed,TMH edition 2006

[2] Barry W. Boehm, “Software risk: management principles and practices”, reprinted from vol-8, No. 1, Jan 1991.

[3] Aihua Yan, “Risk Management in Software Development: A Continuous Process”, IS 6840 Term Paper, Fall 2008; Submitted to Dr. Vicki Sauter, November 21, 2008

[4] Paul L. Bannerman, Risk and risk management in software projects: A reassessment, The Journal of Systems and Software 81 (2008) 2118–2133).

[5] Risk management notes [Online]. Available: http://agile.csc.ncsu.edu/SEMaterials/RiskManagement.pdf