Journal Of Harmonized Research (JOHR)

Journal Of Harmonized Research in Engineering
1(2), 2013, 104-109

ISSN 2347 – 7393

Review Article

MANAGING RISKS: REDUCING ERRORS

Nikita Chhillar, Nisha Yadav, Neha Jaiswal

Department of Computer Science and Engineering,
Dronacharya College of Engineering, Khentawas,
Farukhnagar, Gurgaon, India

Abstract

Project risk management is fundamentally the study of the art and science of recognizing, examining, and responding to risk all the way through the life of a project and in the finest interests of meeting intentions as well as the objectives of project. Usually, in software development, there are various types of software projects that each of them can be influenced by different type of risk. Therefore, it necessitates a distinctive appraisal process of the potential risks that may cause failure or loss of the project whenever they occur. There are actually number of risk assessment researches and investigations carried out in the direction of software projects. Risk management is project management for matures. It directs project managers to focus exclusively on what can be erroneous or incorrect, instead the more standard "if everything goes as planned" approach. Risk Management is a set of practices and sustains tools to identify, evaluate, and treat risks unambiguously. Risk management tries to reduce the possibility of a risk to arise and the impact which is generally loss caused by risks. Risk management is often disregarded in projects, but it can help in enhancing project success and victory by helping choose high-quality excellent projects, determining project scope and extent, and developing practical, sensible and realistic estimates.

Keywords: Risk, risk management, software project management, Software Risk Assessment

Introduction

Risks are indispensable feature for the development of software projects unproductive. Risk is portrayed as "the possibility of experiencing loss that describes the impact on the project which could have been a low quality of software solution with augmented costs, breakdown, or postponed completion". Additionally, all projects share some degree of risk, and the greater part of Information Technology projects have substantial risks. Risk can, though, be abridged, stewarded, supervised and handled in accordance with tight scheduled preparation

For Correspondence:
nikitachhillarATyahoo.com
Received on: October 2013
Accepted after revision: December 2013
Downloaded from: www.johronline.com

and assessment. We cannot forecast that in future there will be 100% accuracy in the software project which is being developed. If we fail to plan for risks early then it leads to catastrophic management or we can say firefighting. Risk is computed as the probability and consequence of not achieving a previously stated and defined objective of the project. Risks however change the life of a project as we become alert how we should work to get into a right direction for developing the project. Risk management is generally known as the act or practice of dealing with risks by minimizing them and enhancing the possibilities of a better project development.

Risk management is considered as proactive rather than reactive because it works before any error occurs. Risk management is not a separate job but it is an aspect of sound project management. Treating a risk means being aware of it in a better way, avoiding and reducing it which is called the risk mitigation, or planning for the risk to materialize. Software engineering risk management activities are carried out at various levels which are the project level, process level, and product level. It is assumed that the management of risks can lead to the successful completion of projects and getting a desired result out of that. Risk management has been also admired in non-software domains for several decades. Currently risk management in software is a developing phenomena – it is rarely understood, and practiced. Compared to the risk management literature available in insurance, and manufacturing the volume of risk management literature available in software is inadequate. Here we attempt to review the fundamentals of software risk analysis and management, the different well known risk management process models, and the recent research trends and styles in the area of software engineering risk management. During the planning of a project, it is significantly essential to know what the major risks are, and is possible plan for them. Some possible areas where planning is done are as stated below:

- Turnover of the Staff
- Change in the Management
    - Unavailability of the required Hardware
    - Change in the Requirements of the project
    - Delays in the Specification
    - Underestimation of the Size
    - Change in the existing Technology
- Competition of the Product with various products already present.

What is risk????

The lexicon meaning of risk is found to be “the possibility of a hazard, loss, injury or danger”. Risk is considered as a measure of the prospect and consequence of not accomplishing a definite project target for which it was build. A risk is an impending dilemma which means that it might occur and it might not. A *risk* is an event, which is uncertain and has a negative impact on some endeavor. Conceptual and theoretical definitions of risk says that it is concerned about future happenings and involves transformation in mind, opinion, actions, places, etc as well as it includes option and the ambiguity that choice necessitates. The term risk is used universally in different contexts. In the “software” world, risk is an important issue often referring to the sources of threat to software development, acquisition, acquirement, procurement, or maintenance. Before proposing any risk management framework one needs to specify the dimensions of risks and understand them carefully because it is a challenge to unanimously have the same opinion on the definition of risk. There are numerous formal definitions of risk presented like “A possible future event that, if it occurs, will lead to an undesirable outcome” another definition is that “Risk is a combination of an abnormal event or failure, and the consequences of that event or failure to a system’s operators, users, or environment.

The most general and familiar definition of risk in software projects is in terms of disclosure to particular factors that are responsible for a threat to achieve the anticipated results and conclusions of a project. On the basis of this definition, risk in software projects is typically defined as the probability-weighted impact of an event on a project. In classical decision theory, risk was examined as reflecting variation in the probability distribution of promising results, negative or positive, associated with a specific resolution. The general concept which is being used these days in software projects is that to reduce the possibility of an unfavorable project outcome, all probable risk aspects should be acknowledged at the initiation of the project. The risk exposure for each factor is then estimated after which the exposures are given priorities to recognize the risks that correspond to the maximum threat to the project. Then the occurrence of the high risk factors is minimized and the degree of impact if they are realized, through control measures. Risk factors are scrutinized progressively to perceive and detect, as early as possible, when they materialize or if the threat changes. A progressive status of identified risk factors is sustained and sporadically updated. Recognition of a risk is regularly identified through the onset of an already defined risk trigger or the reaching of a predetermined risk threshold, at which time predefined contingency plans are set in motion to diminish the impact. This common conception of risk has some limitations.

- First, even before software engineering adopted the definition, management research found that this approach does not match actual managerial behavior.
- A second limitation of this definition is that it is very difficult in practice to estimate the probability of impact of many risk factors, especially in software projects.

The Importance of project risk management

Risk management in software projects has various applications like it helps in saving projects from failure due to different factors for instance non-completion of projects within the specific schedule, not achieving the expectations customer etc. Risk management views the projects from various perceptions to make sure that the dangers to the projects are analyzed and appropriate strategies are undertaken to mitigate the risks. The principle of risk management is to be familiar with all possible risks to a project and consequences, as well as the resolution steps depending on the nature and character of the risks. The idea is to reduce the unpredictable issues taking place during the course of the project by appropriately planning for eventualities. Proper planning minimizes uncertainties, which might lead to astonishing results, or a complete termination of the projects. Software engineering risk management takes a preventive approach leading to completion of projects within scheduled time and money. The risk-managed projects have the capability to decrease project costs, and time of completion, and increase in taken as a whole quality of the project deliverables. Devoid of these, projects could risk loss of revenue as well as customer belief or an absolute bankruptcy of the organization taking part. Conceptually, from the organizational point of view, risk takes place when organizations pursue chances in the face of indecisive, restricted by capability and expenditure. The major challenge is to find a location on each of these dimensions that, when joined together, represents a risk profile that is suitable to the proposal and adequate to internal and external stakeholders. Accordingly, risk and risk management are strategic and governance concerns that usually engross a compromise a risk-averse strategy can boundary distinguishing achievement however, a risk-embracing strategy can amplify project losses. Explicitly supervising this balance is often under-played or ignored in the detection of preferred ambitions. At the project level, software projects have long been acquainted with as high-risk ventures prone to failure arguments are also there that there are basically two classes of software project risk common risks familiar with all projects, and project-specific risks. Several of these risks are easy to spot and control but others are less observable and are more difficult to predict their nature. As a result, risk management in software projects is considered really significant to help in avoiding disasters as well as focus and balance effort and stimulate win–win situations. At the same time as not all risks have their source in software practices, they all have the potential to impact the result of the software process via the project mechanism through which the software artifact is frequently delivered. Risk and risk management are also important because software projects can be automobiles of delivering IT-enabled managerial modifications, so achieving business purposes can be critically reliant upon their achievement.

What Is Risk Management

Software project risk management is typically identified as a set of principles and standards intended at identifying, scrutinizing and managing risk factors to improve and enhance the opportunities of achieving a booming and successful project result and evade the chances of failure of the project. The software development projects in the early years of the last century conducted risk management using different ad hoc approaches, without following any systematic methodologies. Any discrepancy in approach is generally in the standards and practices engaged within this abstract realization of risk management. Most commonly, one or more of four inter-related approaches to risk management are found in the literature and practice. Merely understood, risk management is a method to manage and handle risks. It is concerned with all activities that are executed to reduce the vagueness and uncertainties correlated with particular tasks. With the perspective of projects, risk management diminishes the impacts of adverse events on a project. Risk management in every project necessitates undertaking decisive activities. Risk management is also considered as the act or practice of dealing with risk. It is proactive rather than reactive and not a separate activity but rather on aspect of sound project management. Risk management is a step by step process whose objectives are to recognize, address, and remove software risk items before they become either dangers to successful software operation or a major expensive rework with the same project. *Risk management* is the job of using risk analysis to invent management policy and strategy to minimize or restructure risk. The SEI definition calls attention to the continuous feature of risk management, hence it is given the name Continuous Risk Management (CRM). A project with risk management intends to early recognition of risks and then actively changes the course of actions to mitigate and reduce the risk. Risk management in software has been in existence for many decades. On the other hand, it is only in the last decade, and a half or so that it has gained extensive significance in the software society. Though, with the increasing complexity of software development, industries have realized the importance of risk management, because it helps in minimizing the confusions involved in developing software, and decreasing the possibilities of project failures. Prior to application of any risk management process, the project team should be quite clear about the following dimensions of risks in their projects:

- The character of uncertainty involved, and the probability with which the risk will take place.
- The loss that will be acquired if the risk occurs which has many forms including loss of revenue and loss of customer goodwill.
- The severity of the loss going to take place.
- The extent of the risks.

Risk Management Paradigm/ Model

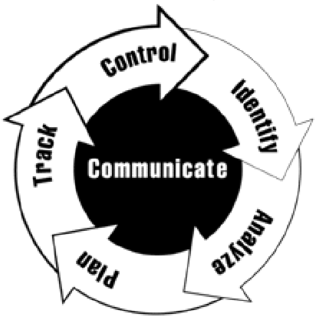
The risk management model shown below illustrates the various activities involved in the process of management of risk associated with project on which we are working. The paradigm is symbolized by a circle to accentuate that risk management is a continuous and nonstop process, whereas the arrows show the consistent flow of information among the activities. Communication is positioned in the heart or we can say middle of the model because it is both the channel through which every single information flows and, often, the largest barrier in risk management. Basically, the paradigm is a skeleton or framework for risk management. From this model, a project may structure a risk management practice most excellent appropriate for its management structure. Each risk management paradigm activity is described as follows.

Identify

Prior to risk management process taken into action, they are necessary to be identified. The identification examines and investigates risks before they become severe problems. There are different techniques for surfacing risks by the application of a systematic process that encourages project personnel to raise concerns and issues.

Analyze

Analyzing is the basically the translation of risk data into risk information which means the decision to be made. Analysis provides the foundation for the project manager to work on the appropriate and most significant risks.



Plan

Planning is the process which turns risk information into decisions and actions. Planning engages actions to be taken to tackle with individual risks, giving priority to risk actions, and generating an integrated risk management plan. The plan for a specific risk can take many forms like it can alleviate the impact of the risk by developing an emergency plan should the risk arise and evade a risk by changing the design of the project. The solution to risk action planning is to consider the future outcomes of a verdict made today.

Track

Tracking is composed of monitoring the condition of risks and the actions taken to modernize them. Suitable risk metrics are acknowledged and scrutinized to facilitate the evaluation of the status of as well as of the plans made.

Control

Risk control is responsible for correcting deviations from previously planned risk actions. When risk metrics and triggering events have been selected then there is nothing distinctive about risk control. Risk control joins into project management and relies completely on project management processes to control risk action plans, corrections in variations from defined plans, responds to the triggering events, and enhances risk management processes.

Communicate

Risk communication is positioned at the center of the paradigm to highlight both its commonness and its complexity. Without effective communication, no risk management process can be feasible. Communication facilitates interaction i.e. the exchange of information among the constituents of the model. In order to be analyzed and managed accurately, risks must be communicated with the appropriate organizational levels. This comprises levels within the development project and organization, within the customer organization, and across that threshold between the developer, the customer, and, where different, the user. Because communication is common, our approach is to address it as a whole to every risk management activity and not as something performed as a supplement to other activities.

Intelligent Risk Management Tools

Intelligence generally means the ability to learn, which means to acquire knowledge and information and then use that knowledge to achieve a modification in the behavior. Two most important faults of repository-based risk management tools is their deficiency of deductive power and that they are very common. This to a great extent minimizes their effectiveness in software projects where each project is unique and specific. Knowledge based tools, alternatively, while exhibiting deductive capability are tied to particular technology or development techniques. This diminishes their independence from the technology and systems which exists in their own environment. With this in practice, software development is a mixture or hybrid of notorious and frequently not-so well-acknowledged processes. Accordingly there arises a requirement of risk management tools which are actually intelligent and independent of the techniques and systems technology which are being used. This kind of tools contains the capability to learn and transform behavior depending on what precisely emerges during the life cycle of project. Additionally, use of such a tool in future projects makes certain continuity in the use of experience in risk management from previous projects. Apart from the knowledge-based risk management tools which are found to be based on artificial intelligence, there exist some tools based on mathematical modeling. Similarly we can develop more and more efficient tools keeping our requirements in our mind.

Conclusions

As we know that to complete any particular project there is a need of starting it and with the beginning of a project arises different types of risk for which there is a need of minimizing them using the process of risk management. In this paper, we endeavored to present a general idea of the essential and elementary perceptions on risk management in software engineering and significant research conducted in the area. The demand for risk management tools which are intelligent has also been exhibited. This kind of tools would have the competence to be used with any development methodology, whether traditional, agile, or even a grouping of them.

References

- Hazel T. A., 2004, Risk Management and Tacit Knowledge in IT Projects: Making the Implicit Explicit,
- Thesis, Queensland University of Technology, 2008
- Higuera R.P., & Haimes Y. Y., Software Risk Management, CMU/SEI-96-TR-012, ADA 315789, 1996
- Kimer, T.G., &Concalves, L.E., 2006, Software Engineering Techniques: Design for Quality , IFIP International
- Federation for Information Processing,Volume 227, pp. 149-154.
- Kontio, J , Jokinen J, & Rosendahl, E Visualizing and formalizing risk information: An Experiment,
- Proc. 10th International Symposium on Software Metrics, Finland, 2004
- Boehm, B.W., 1991. Software risk management principles and practices. IEEE Software 8 (1), 32–41.
- Boehm, B.W., DeMarco, T., 1997. Software risk management. IEEE Software 14 (3), 17–19.
- Roberts, B.B., 2001. Lessons-learned: round 2. In: 2001 INCOSE Proceedings of a Symposium on Risk Management,, 3.
- Schmidt, R., Lyytinen, K., Keil, M., Cule, P., 2001. Identifying software project risks: an international Delphi study. Journal of
- Management Information Systems 17 (4), 5–36.
- Yacoub, S.M., Ammar, H.H., 2002. A methodology for architecture level reliability risk analysis. IEEE Transactions on Software
- Engineering 28 (6), 529–547.
- Higuera, R.P., Gluch, D.P., Dorofee, A.J., Murphy, R.L., Walker, J.A.,Williams, R.C., 1994. An introduction to team risk management.
- In: Special Report, CMU/SEI-94-SR-1. Software Engineering Institute.
- Jiang, J.J., Klein, G., 1999. Risks to different aspects of system success. Information and Management 36 (5), 263–272.