Project-Based Software Risk Management Approaches

Subhas C. Misra
Carleton University, Canada

Vinod Kumar
Carleton University, Canada

Uma Kumar
Carleton University, Canada

INTRODUCTION

The last few decades—especially the end of 20th century and the beginning of 21st century—have shown an increase in the interest in automation of different activities. Automation is dependent in its core on sound functional software. The complexity of software development has increased significantly over the years. Articles showing the failure of projects in the software industry are not surprising. Standish Group (1994) reports show that about 53% of projects get completed, but they do not meet the cost and schedule requirements, and about 31% are canceled before the completion of the projects. These failure reports are significantly alarming.

With the tremendous growth in the complexity of software development in the last 10 to 15 years, the management of risks in software engineering activities is becoming an important and nontrivial issue from three perspectives: project, process, and product. Therefore, researchers and practitioners are continually trying to find effective risk management approaches.

This article should help the academicians, researchers, and practitioners interested in the area of risk management in software engineering to gain an overall understanding of the area.

BACKGROUND

Meaning of Risk Management

Simply put, risk management is a way to manage risks. In other words, it concerns all activities that are performed to reduce the uncertainties associated with certain tasks or events. Risk management reduces the impacts of undesirable events on a project or the final product. Risk management in any project requires undertaking decision-making activities.

Origin of Risk Management

Risk management has its roots in probability theory and decision making under uncertainty. Three well-known theories in these areas—*expected utility theory* (Bernoulli, 1954; Hogarth, 1987), *theory of bounded rationality* (Simon, 1979), and *prospect theory* (Kahneman & Tversky, 1973; Kahneman, Slovic, & Tversky, 1982)—were of the greatest influence. These theories may be considered as disciplines by themselves. Therefore, to put our discussions on risk management in context, we briefly state hereafter only what each of these theories propose.

In brief, the expected utility theory discusses how people make choices from different alternatives, based on their expected utility. The theory of bounded rationality states that for real life events the outcomes and their associated probabilities are very limitedly understood by people to make the required decisions to maximize their expected utility. Therefore, people have a tendency to set up targets of aspiration in life by eliminating alternatives from the different options they have. This theory is useful for modeling the behavior of project management personnel in charge of risk management. Prospect theory, which has its origin in psychology, helps to model how the perceptions of human beings influence their choices from the given options. Thus, it helps for understanding and estimating the utility losses of different alternatives while analyzing risks in risk management.

Purpose of Risk Management

Risk management in software has different uses. It helps to save projects or products from failing due to different factors such as noncompletion of projects within the specified schedule and budget constraints and not meeting the customer expectations of the final product.

In the context of projects, risk management looks at projects from different perspectives to ensure that the threats

Copyright © 2009, IGI Global, distributing in print or electronic forms without written permission of IGI Global is prohibited.

to the projects are identified and analyzed, and appropriate strategies are undertaken to mitigate and control risks. The mitigation strategies may not necessarily mean the cancellation of tasks that involve risks. Many tasks are undertaken in the software industries even after knowing that undertaking them involves taking high risks. The high-risk tasks are sometimes important to provide the industries a leading edge over their competitors.

Software risk management takes a preventative approach leading to completion of projects or the development of products within predictable time, money, and according to the product specifications. In fact, risk-managed projects and products have the ability to reduce costs and time of completion and increase the overall quality of the project and product deliverables. Without these, organizations could risk loss of revenue and customer trust in an average case, or a complete bankruptcy of the participating organizations in the worst.

RISK MANAGEMENT IN SOFTWARE PROJECTS

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. However, with the increasing complexity of software development, industries have realized the importance of risk management, because it helps in reducing the uncertainties involved in developing software and decreasing the chances of project or product failures.

In the context of projects, before applying any risk management method, the team members should be clear about the following dimensions of risks in their projects (Smith & Pichler, 2005):

- The nature of *uncertainty* involved, and the likelihood with which the risk will occur.
- The *loss* that will be incurred if the risk occurs. Loss in software projects can take many forms including loss of revenue, loss of market share, and loss of customer goodwill.
- The *severity* of the loss.
- The *duration* of the risks.

Different Approaches

Project Risk Management

Several software project risk management approaches have been proposed in the past, most of which assess risks during all the phases of software development, by integrating risk management practices along with the software development process. As a result, in these approaches the risk management approaches follow a disciplined process. These approaches are listed as follows:

- Boehm's risk management model (win-win) (Boehm & Ross, 1989; Boehm & Bose, 1994; Boehm et al. 1998),
- SEI's software risk management model (SRE Version 2.0) (Williams et al., 1999),
- Hall's risk management model (P^2I^2) (Hall, 1998)
- Karolak's risk management model (Just-In-Time Software) (Karolak, 1998), and
- Kontio's riskit methodology (Kontio, 2001).

A "horizontal" comparison of all of these approaches may not be fair because although each of them addresses risk management, they were developed under different circumstances for solving—may be related but different issues. For example, Hall's P^2I^2 was developed from a risk management capability modeling perspective. On the other hand, Boehm's win-win model (Boehm & Ross, 1989; Boehm & Bose, 1994; Boehm *et al*. 1998) was developed primarily as a novel software development process model ("spiral" development) taking a risk-based approach. However, we provide later on an overview of the characteristics of all these approaches.

Of all these approaches, Boehm's win-win (Boehm & Ross, 1989; Boehm & Bose, 1994; Boehm *et al*. 1998) is perhaps the most influential software engineering risk management process model, which became popular during the early 1990s. He developed the first software engineering risk management process model, which integrates seamlessly into the software development lifecycle.

SEI's software risk evaluation approach (called, SRE) (Williams et al., 1999) is also quite popular in practice. It has been applied in several software development projects of several government, and nongovernment organizations. SEI's SRE provides a systematic, detailed, and step-wise approach one could use in software development and acquisition projects. It is based on the idea of continuous risk management. Another characteristic of SRE is that it integrates team risk management principles into the core framework.

Hall (1998) proposed a framework from a different perspective. She proposed a comprehensive framework based on the notion of risk management capability maturity. Her approach is based on four critical success factors of risk management, namely, people, process, infrastructure, and implementation (P^2I^2). However, it is the "process" component of the framework which discusses the risk management processes.

Karolak (1996, 1998) looked at software engineering risk management from the just-in-time viewpoint, the idea of which was popular in the traditional manufacturing industries. Like Hall, he also provided a complete framework that one

3 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/project-based-software-risk-management/14039

Related Content

Benchmarking Serverless Computing: Performance and Usability
Mubashra Sadaqat, Mary Sánchez-Gordón and Ricardo Colomo-Palacios (2022). *Journal of Information Technology Research (pp. 1-17).*
www.irma-international.org/article/benchmarking-serverless-computing/299374

Wireless Networks for Vehicular Support
Pietro Manzoni, Carlos T. Calafate, Juan-Carlos Cano, Antonio Skarmeta and Vittoria Gianuzzi (2009). *Encyclopedia of Information Science and Technology, Second Edition (pp. 4135-4140).*
www.irma-international.org/chapter/wireless-networks-vehicular-support/14197

The Role of Social Media in the Diffusion of E-Government and E-Commerce
Ibrahim Osman Adam and Muftawu Dzang Alhassan (2021). *Information Resources Management Journal (pp. 63-79).*
www.irma-international.org/article/the-role-of-social-media-in-the-diffusion-of-e-government-and-e-commerce/275725

Trends in Information Technology Governance
Ryan R. Peterson (2005). *Encyclopedia of Information Science and Technology, First Edition (pp. 2865-2870).*
www.irma-international.org/chapter/trends-information-technology-governance/14709

Privacy Rights Management: Implementation Scenarios
Larry Korba, Ronggong Song and George Yee (2007). *Information Resources Management Journal (pp. 14-27).*
www.irma-international.org/article/privacy-rights-management/1304