Risks Management in Software Engineering

Dishek Mankad
M.C.A. Department, B.R. Patel Institute of Computer Application Dahemi, Anand-Gujarat, India

Abstract

Risk management is an action that helps a software development team to understand what kinds of risks are there in software development. Risk is always concern with today's and yesterday's uncertainty. It is a potential problem. So, it might happen, it might not. It is better to identify its probability of occurrence.

Keywords

Risk management, Software Engineering, Development, Risk Identification.

1. Introduction

There are lots of risks involved while creating the high quality software on the time and within budget. However, in order for it to be advantageous to take these kinds of risks, they must be cover for by a perceived reward. The greater the risk, the greater the reward must be to make it worthwhile to take the great opportunity. In applications building the possibility of reward is great, but so is the prospective for tragedy. "If you don't actively attack the risks, they will actively attack you". In order to successfully manage a software project and reap our rewards, we must learn to identify, analyze, and control these risks. This Paper focuses on the basic concepts, processes, and techniques of software risk management. There are basic risks that are universal to almost all software projects. Although there is a simple basic component of risk management inherent in good project management, risk management differs from project management in two following ways:

Project Management	**Risk Management**
Designed to address the general risks.	Designed to focus on risks unique to each project
Looks at big graphical plans for details	Looks at potential problems and plans for contingencies
Programs what should occur and looks for ways to create it happen	Evaluates what could happen and looks for various ways to minimize the fault.
Plans for the success	Plans to manage and mitigate potential causes of failures

Boehm defines four major reasons for implementing software risk management [Boehm-89]:

1. Avoiding software project disasters, including run away budgets and schedules, defect-ridden software Products and operational failures.
2. Avoiding rework caused by erroneous, missing, or ambiguous requirements, design or code, which typically consumes 40-50% of the total cost of software development.
3. Avoiding overkill with detection and prevention techniques in areas of minimal or no risk.
4. Stimulating a win-win software solution where the customer receives the product they need and the vendor makes the profits they expect.

2. Define Software Risks

So, what are risks? Risks are concern with danger. It is potential problems. Suppose every time we cross the street, we run the risks of being hit by the vehicle. The risk does not start until we make the commitment, until we step in the street. It automatically ends when the problem occurs or the possibility of risk is eliminated. A software project may encounter various types of risks likes:

- **Project risks** identify potential budgetary, schedule, personnel (staffing and organization), resource, stakeholder, and requirements problems and their impact on a software project.
- **Technical risks** include problems with languages, project size, project functionality, platforms, methods, standards, or processes. These risks may result from excessive constraints, lack of experience, poorly defined parameters, or dependencies on organizations outside the direct control of the project team.
- **Business risks** include 1. Building and excellent and intelligent product or system that no one really wants (Market risks). 2. Building a product or system which is no longer fits into the overall business strategies for company (strategy risks). 3. Building a product that the sales force doesn't understand how to sell (sales risks).

4. Losing the support of senior management due to a change in focus or change in people (Management risks). 5. Losing the budgetary or personnel commitment (budget risks).

- **Known risks** are those that can be uncovered after careful evaluation of the project plan, the business and technical environment in which the project is being developed, and other reliable information sources.
- **Predictable risks** are extrapolated from past project experience. (Staff turnover, poor communication with the costumer, etc.)
- **Unpredictable risks** are the joker in the deck. They can do occur, but they are extremely difficult to identify in advance

3. Risk Management Process

This process starts with the identification of a list of potential risks. Each of these risks is then analyzed and prioritized. A risk management plan is created that identifies containment actions that will reduce the probability of the risk occurring and/or reduce the impact if the risk turns into a problem. The plan also includes contingency actions that will be taken if the risk turns into a problem and the associated triggers (indicators that the risk is turning into a problem). The containment part of the plan is then implemented and actions are taken. The tracking step involves monitoring the status of known risks as well as the results of risk reduction actions. If a trigger indicates the onset of a problem, the corresponding contingency plans are implemented. As new status and information are obtained, the risk management plans are updated accordingly. Tracking may also result in the addition of newly identified risks or in the closure of known risks.

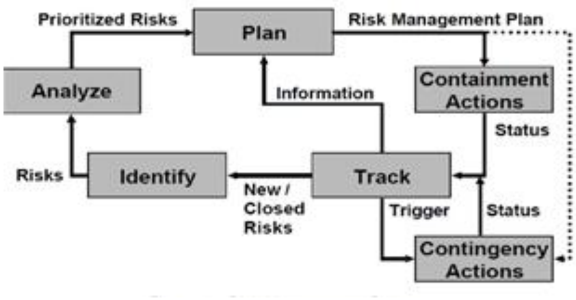


Figure1: Risk Management Process

Risk management process is on-going part of managing the software building. It si designed to be a continuous feedback loop where additional information and risk status are utilized to refine the project's risk list and risk management plans.

Risk Identification

Risk Identifications is a systematic attempt to specify threats to the project plan. By identifying known and predictable risks, the project manager takes a primary step towards avoiding them when possible and controlling them when necessary.

There are two types of risks for each of the categories that have been presented. Generic risks and product-specific risks. Generic risks are a potential risks to every software that is to be built. Project specific risks can be identified only by those with a clear understanding of the technology, the people, and the environment that is specific to the software that is specific to the software that is to be developed. One method for identifying risks is to create a risk item checklist. The checklist can be used for risk identifications and focuses on some subset of known and predictable risks in the following generic subcategories that is:

- **Product size** – risks associated with the overall size of the software to be developed or modified.
- **Business impact** – risks associated with constraints imposed by management or the marketplace.
- **Stakeholder** – risks associated with the sophistication of the stakeholders and the developer's communicate with stake holders in a timely manner.
- **Process definition** – risks associated with the degree to which the software process has been defined and is followed by the development of organization.
- **Development environment** – risks associated with the availability and quality of the tools to be used to build the project.
- **Technology to be built –** risks associated with the complexity of the system to be built latest of the new technology that is packaged by the system.

Staff size and experience – risks associated with the overall technical and project experience of the software engineering that will do the particular work.

The risk item checklist can be organized in different way. Questions are relevant to each of the topic can be answered for each software project the answer of the each questions are allows you to estimate the impact of the risk

Finally the numbers of risk components and drivers are listed along with their probability of occurrence. The risk components are defined in the following manner:

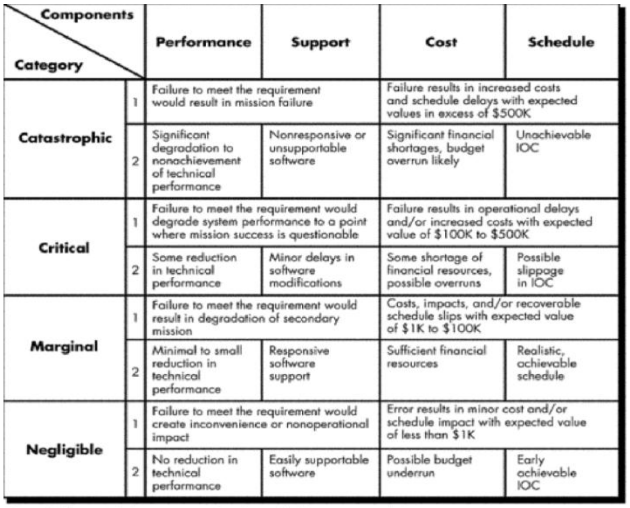
- **Performance risk** – that the product will meet its requirements and be fit for its intended use.
- **Cost risk** – that the project budget will be maintained.
- **Support risk –** which the resultant software will be easy to correct, adapt and enhance.
- **Scheduled risk** – which the project scheduled, will be maintained and that the project will be delivered on time.

The impact of each risk driver on the risk component is divided into one of four impact categories – negligible, marginal, critical, or catastrophic. For that we have to develop the risk calculation table or we can say risk table. The impact category is chosen based on the characterization that best fits the description in the table.

Risk Projection

Risk identification is also known as risk estimation, attempts to rate each risk in two ways – (1) Probability that the risk is real. (2) The consequences of the problem associated with the risk, should it occur. You always work along with other managers and technical staff to perform four (4) risk projection steps:

- Establish a scale that reflects the perceived probability of a risk.
- Delineate the consequences of the risk.
- Estimate the impact of the risk on the projection and the product.
- Assess the overall accuracy of the risk estimation so that there will be no misunderstanding.



Components / Category		Performance	Support	Cost	Schedule
Catastrophic	1	Failure to meet the requirement would result in mission failure		Failure results in increased costs and schedule delays with expected values in excess of $500K	
	2	Significant degradation to nonachievement of technical performance	Nonresponsive or unsupportable software	Significant financial shortages, budget overrun likely	Unachievable IOC
Critical	1	Failure to meet the requirement would degrade system performance to a point where mission success is questionable		Failure results in operational delays and/or increased costs with expected value of $100K to $500K	
	2	Some reduction in technical performance	Minor delays in software modifications	Some shortage of financial resources, possible overruns	Possible slippage in IOC
Marginal	1	Failure to meet the requirement would result in degradation of secondary mission		Costs, impacts, and/or recoverable schedule slips with expected value of $1K to $100K	
	2	Minimal to small reduction in technical performance	Responsive software support	Sufficient financial resources	Realistic, achievable schedule
Negligible	1	Failure to meet the requirement would create inconvenience or nonoperational impact		Error results in minor cost and/or schedule impact with expected value of less than $1K	
	2	No reduction in technical performance	Easily supportable software	Possible budget underrun	Early achievable IOC

Note: (1) The potential consequence of undetected software errors or faults.
(2) The potential consequence if the desired outcome is not achieved.

The intention of these steps is to consider risks in a manner that leads to give priority. No software team has the resources to address every possible risk. By priority risks, you can allocate resources where they will have the most impact.

4. Conclusion

My research work is on Risk Management in software engineering. Here I am research work is focusing on risks are in the software building. By reading this paper everyone can easily understand that what is risk in software and how to manage the risk.

Acknowledgment

The success of my project is never limited to the individual undertaking the project. It is the cooperative effort of the people around an individual that spell success. For all efforts, behind this successful project, I am highly intended to the following personalities without whom this project would ever be completed. We find no words to express our gratitude towards those who were constantly involved with us throughout our work.

References

[1] Boehm, Barry. Software risk management. Springer Berlin Heidelberg, 1989.
[2] Fairley, Richard. "Risk management for software projects." IEEE software 11.3 (1994): 57-67.
[3] Gilb, Tom, and Susannah Finzi. Principles of software engineering management. Vol. 4. Reading, MA: Addison-Wesley, 1988.
[4] [Ould, Martyn A. Strategies for software engineering: the management of risk and quality. John Wiley & Sons, Inc., 1990.
[5] Carr, Marvin J., et al. Taxonomy-based risk identification. No. CMU/SEI-93-TR-06. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 1993.

Dishek J. Mankdad born in Junagadh – Gujarat. I was born on 19/09/1986. I am doing my graduation in Commerce in all the way from junagadh in Junagadh Kalevani Mandal – Junagadh in my graduation I've achieve second class. In my graduation my special subject is Computer. So, my first preference is to always stay connected with the computer after that I start preparation for M.C.A. Entrance and finally I got admission in M.C.A. program in N.S.V.K.M.S. M.C.A. College – Vishnagar(Gujarat). In M.C.A I've got first class in that. After completion of my M.C.A. I was working as a programmer in company named YudiZ Solution for 8 months. In the company my training period is considered as an employee. After that I was working in Veerayatan Institute of Computer Application as a H.O.D of computer Applications Department. And currently I am working with B.R.Patel Institute of Computer Application (M.C.A. Program) as an Assistant Professor.