

A Summary of Tools and Tricks for Software Risk Management

Michael J. Sepcot
Illinois Institute of Technology
sepcmic@iit.edu

INTRODUCTION

Risk Management in the software industry is one of the most critical processes in determining the outcome of a software project. A good program manager recognizes this fact and allocates the necessary time and resources to risk management processes. Risk management consists of two major parts: risk assessment and risk control. Each of these parts is, in turn, made up of sub-components that need to be accounted for in order for complete risk control and aversion.

In the software industry, a majority of the software projects are not being delivered on time, within a set budget, or up to quality standards within the industry. A major cause of software project failure is due to unforeseen events that creep into the project while the program developers and managers are focused on hammering out lines of code. Most of these risk items can be easily avoided by implementing proper risk management techniques in an explicit and systematic way – ad hoc methods will not be effective in countering risk.

This paper attempts to summarize the lessons learned in the implementation of risk management procedures across major governmental and commercial industries such as Daimler-Benz, Nokia, NASA's Space Infrared Telescope Facility project, and the Air Force Aeronautical Systems Center's Integrated Risk Management team [2][4][5]. This paper also draws on risk management principles and practices learned from the implementation of the Riskit [6] method in industry and custom applications designed to help program managers' account for risk in terms of implications on schedule, cost, and technical matters [1][2][3][4][5]. Only when risks are viewed from the perspective of project stakeholders and goals can risks be accurately accounted for.

This paper attempts to put forward, in a concise manner, all of the key areas in software risk management – from methodology to software tools – that will help program managers curb loss due to risk by identifying risk areas and implementing risk prevention techniques.

RISK MANAGEMENT

To understand risk management, one must first understand what a risk is. A risk in the software industry can be defined on two fronts: a negative event occurring in a project that adversely affects quality, schedule, or cost factors; or, an external event that creates an advantage for competitors. The latter of which cannot be controlled internally. Each risk is comprised of two main attributes: the probability that the risk event will occur; and the consequences of the risk event in terms of quality, schedule, and cost. A good program manager will use the risk management techniques described later to identify, address, and monitor risk events to reduce their possible impact on the program. Risk management is all about the process of controlling these risk events.

At the University of Maryland in 1997, Jyrki Kontio published the completed first version of the Riskit method. The Riskit method was developed to give program managers an explicit and systematic process for risk management while providing precise definitions of risks and risk effects, and tie risks to specific project goals and stakeholders. In doing so, all parties associated with a project can easily understand the risks that face the development of the program and assess the implications of risk events in their own terms (each stakeholder knows what risks have the potential to affect their goals).

The main characteristics of the Riskit method are defined in the following principles [6]:

- (1) *The Riskit method provides precise and unambiguous definitions for risks.*
- (2) *The Riskit method results in explicit definition of objectives, constraints and other drivers that influence the project.*
- (3) *The Riskit method is aimed at modeling and documenting risks qualitatively.*
- (4) *The Riskit method can use both ratio and ordinal scale risk ranking information to prioritize risks reliably.*
- (5) *The Riskit method uses the concept of utility loss to rank the loss associated with risk.*
- (6) *Different stakeholder perspectives are explicitly modeled in the Riskit method.*
- (7) *The Riskit method has an operational definition and training support.*

To put the principles described above into practice, the Department of Defense [5] has developed a Risk Management structure (Figure 1) that contains the four basic strategies needed for success. Those strategies include: Risk Planning, Risk Assessment, Risk Handling, and Risk Monitoring.



Figure 1: Risk Management Structure [5]

With Risk Planning, the program manager needs to develop plans of attack. The processes for identifying and tracking risks are developed along with the risk migration and assessment plans that are used to monitor and control the risk through the project life-cycle. All scheduling work for risk identification and risk monitoring should be performed during this stage of the project to ensure proper resources are set aside for risk management activities.

During Risk Assessment, the program manager sits down with higher management and team leaders to identify all possible risks. By using brainstorming sessions and paying close attention to the Top 10 Software Risk Items defined by Barry Boehm (Table 1) [1], program managers guide risk identification meeting to analyze the program, process, requirements, and technologies to identify and document all possible risk areas. Once the risk areas are identified, a process of risk analysis is then followed to isolate the cause of the risk areas and determine their impact on the project.

Table 1: Top 10 Software Risk Items [1]

Risk Item	Risk Management Technique
Personnel shortfalls	Staffing with top talent, job matching, team building, key personnel agreements, cross training.
Unrealistic schedules and budgets	Detailed multi-source cost and schedule estimation, design to cost, incremental development, software reuse, requirements scrubbing.
Developing the wrong functions and properties	Organization analysis, mission analysis, operations-concept formulation, user surveys and user participation, prototyping, early users' manuals, off-nominal performance analysis, quality-factor analysis.
Developing the wrong user interface	Prototyping, scenarios, task analysis, user participation.
Gold-plating	Requirements scrubbing, prototyping, cost-benefit analysis, designing to cost.
Continuing stream of requirements changes	High change threshold, information hiding, incremental development (deferring changes to later increments).
Shortfalls in externally furnished components	Benchmarking, inspections, reference checking, compatibility analysis.
Shortfalls in externally performed tasks	Reference checking, pre-award audits, award-fee contracts, competitive design or prototyping, team-building.
Real-time performance shortfalls	Simulation, benchmarking, modeling, prototyping, instrumentation, tuning.
Straining computer-science capabilities	Technical analysis, cost-benefit analysis, prototyping, reference checking.

After we have isolated the cause of the risk areas, we move into Risk Handling. Here we identify various strategies to reduce the impact of the risk areas on the project. There are four basic categories of Risk Handling strategies: Avoidance, Control, Prevention, and Assumption. With Risk Avoidance, we make changes to the project requirements to reduce the probability of the risk occurring while still meeting program objectives. In Risk Control (or Mitigation), we take active steps in determining the correct process to use to reduce the impact on the process. Risk Prevention (or Transfer) re-allocates design requirements to areas that can accomplish program objectives at a reduced risk (such as offshore companies). Finally, there is Risk Assumption, where we just accept the risk for what it is and hope for the best.

The final stage in the Department of Defense Risk Management structure is probably the most important: Risk Monitoring. In Risk Monitoring, we actively track and evaluate risks and their risk handling actions through the use of indicators and metrics. Both the indicators (qualitative) and metrics (quantitative) measures need to be analyzed to provide feedback on our risk handling activities.

IMPLEMENTING RISK MANAGEMENT

To successfully implement risk management processes in industry, adequate training needs to be given to all participants. Case studies performed at Daimler-Benz AG [3], Nokia Telecommunications [3], and Tenovis [4] show a direct relationship between time spent in risk management training and benefits derived from the implementation. The project management and subproject team leaders in Daimler-Benz were only given one hour of formal training. With the low level of training, Daimler-Benz failed to formally define the risk management mandate, they did not support stakeholder analysis in risk areas, and they perceived Riskit scenarios as too

complex. At Nokia, where informal training was facilitated to the project team in a one-hour session after a two-hour private session to the project manager, risk management was better carried out. Risk management also benefited Tenovis where a full-day workshop was provided.

In the above mentioned case studies, risk monitoring was viewed as one of the most important aspects of risk management. There are two key points that can be drawn from the case studies:

- (1) *Risk Monitoring needs to be performed on a regular basis*
- (2) *Risk Monitoring needs to evaluate the progress of risk handling actions*

Performing risk monitoring on a regular basis keeps the most critical risks under check. With ad-hoc meetings every now and then, it is difficult for management to have a realistic view on the status of risk items and the processes being carried out to reduce their impact. Barry Boehm [1] recommends following a *Project top-10 risk-item tracking* system to reduce management surprises when it comes to critical risk areas within the project. To build the Top-10 list, program managers should follow the *Riskit Pareto ranking technique* [6] where expected utility loss is estimated by multiplying the probability of the risk item occurring by the utility loss that would incur given that scenario. There is even a software solution available for risk prioritization in P/CS [5]. P/CS or Probability/Consequence Screening is a VB program designed to be implemented at any point during a project life-cycle. P/CS has the ability to link risk assessment to specific schedule and cost files as well as deliver a P/CS Matrix that graphically lists risk items in a matrix for easy identification of critical risk areas.

Other techniques available to risk monitoring include: Milestone Tracking, Risk Reassessment, and Corrective Action. In Milestone Tracking, program managers document the status of risk handling actions at predefined milestones throughout the project. During this time it is also helpful to go through a short risk identification process in order to document and understand all the risk scenarios that might have come up or been noticed since the initial project risk identification. Any time risks can be identified and documented before they occur gives program managers a chance to implement risk handling actions to reduce project delays. With Risk Reassessment, program managers need to re-evaluate the status of risk scenarios to develop new Top-10 lists and focus on the most critical risks at the given time. Finally, with Corrective Action, we address the second point described above: evaluating the progress of risk handling actions. Here, program managers review the risk handling techniques being applied to various risk scenarios and make changes where necessary to ensure the best techniques are being used to counter risks.

While the Department of Defense Risk Management structure has four critical areas of focus, the Riskit methodology expands on this set and creates the risk management cycle [5] (Figure 2). The Riskit risk management cycle starts with the Risk Management Mandate Definition. This definition sites the scope and frequency of risk management to be used in the project along with identifying all of the key stakeholders that will be affected by the outcome of the project. With Goal Review, program managers can identify all of the areas where risk management needs to focus on, and identify all of the stakeholders for each of the project goals. Risk Identification focuses on identifying the potential threats to each of the project goals and stakeholders. Risk Analysis takes the identified potential threats, consolidates them, and analyzes each risk scenario

to determine the two main risk factors: the probability that the risk event will occur and the utility loss associated with the risk. Risk Control Planning takes the most critical risks identified in Risk Analysis and determines the best risk handling actions to take to lower the possibility of occurrence for each of the risks. Risk Control puts the planned handling actions into motion. And Risk Monitoring keeps constant watch on the risk situation by monitoring the status of the project and the status of risk monitoring metrics. These seven steps of the Riskit process are summarized in Table 2 below with the output of each step identified.

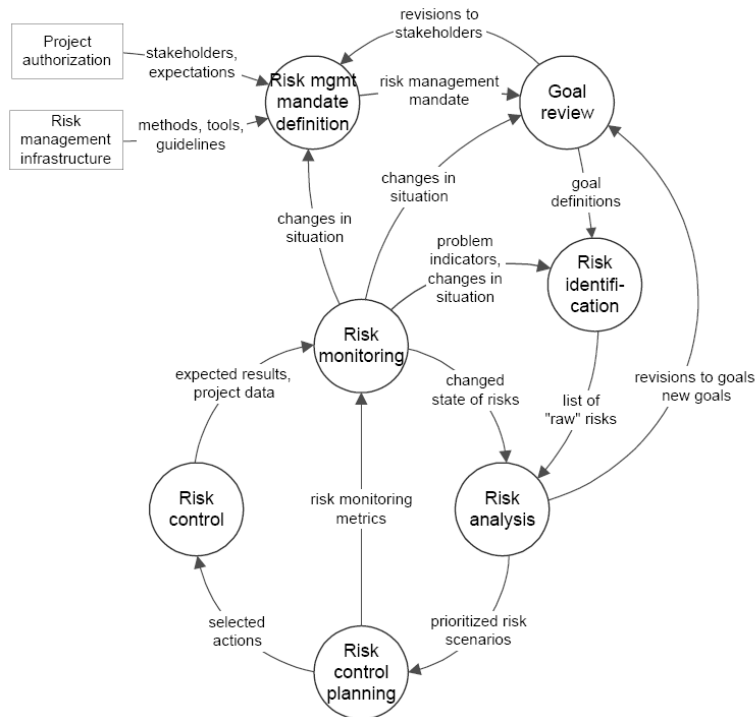


Figure 2: The Riskit Risk Management Cycle [6]

Table 2: Overview of outputs and exit criteria of the Riskit process [6]		
Riskit Step	Description	Output
Risk management mandate definition	Define the scope and frequency of risk management. Recognize all relevant stakeholders.	Risk management mandate: why, what, when, who, how, and for whom
Goal review	Review the stated goals for the project, refine them and define implicit goals and constraints explicitly. Analyze stakeholders' associations with the goals.	Explicit goal definitions
Risk identification	Classify and consolidate risks. Complete risk scenarios for main risk events. Estimate risk effects for all risk scenarios. Estimate probabilities and utility losses of risk scenarios.	A list of "raw" risks.
Risk control planning	Select the most important risks for risk control planning. Propose risk controlling actions for most important risks. Select the risk controlling actions to be implemented.	Completed Riskit analysis graphs for all analyzed risks. Ranked risk scenarios.
Risk control	Implement the risk controlling actions.	Reduced risks.
Risk monitoring	Monitor the risk situation.	Risk status information.

There is an array of software tools available to program managers to aid risk management procedures. The Aeronautical Systems Center's P/CS, mentioned above, is just one of the tools at a program managers disposal. For those program managers using Microsoft Project® - ProjectGear Inc. has a software tool called Risk+©. Risk+ seamlessly integrates with Microsoft Project to provide program managers with the ability to quantify cost and schedule uncertainties and display risk critical paths through the use of probabilistic branching and conditional IF-THEN-ELSE statements. Decisioneering has an application entitled Crystal Ball® that is a Microsoft Excel® Add-In tool that uses randomly generated values for uncertainty variables to simulate real-life systems in their Monte Carlo simulation for spreadsheets. Crystal Ball provides program managers with graphical risk assessment feedback based on their spreadsheet data. NASA's Space Infrared Telescope Facility (SIRTF) project [2] created a web-based risk management system. SIRTF uses a risk management tracking database to capture risk knowledge, produce reports, and provide an easily accessible, expandable, and adaptable system to aid in NASA NPG 7120.5A requirements compliance. With the aid of tools like these, program managers can reduce the amount of time spent completing risk management activities and focus more on managing the overall project.

KEY FOCUS AREAS

Reviewing the lessons learned from the NASA [2], Baimler-Benz and Nokia [4], and Tenovis [3] case studies, there are six key focus areas for software risk management:

- (1) A common risk management framework, such as Riskit, is needed to make risk management efficient. Riskit provides practical and understandable features that lead to increased confidence in risk handling actions.*
- (2) Stakeholders and goals play a critical role in risk management and categorizing the risks helps provide insight to the program manager when making risk handling decisions.*
- (3) Risk management processes must be supported and enforced, and have the commitment of the program manager from the beginning of the project.*
- (4) Training for risk management should be done in a half-day session and be completed before the project starts to ensure efficiency and understanding of the risk management process.*
- (5) Explicit and systematic risk management, such as Riskit, is considered useful and produces different results than one would expect following intuitive risk management procedures.*
- (6) There are many software tools available to help program managers' deal with risk management, but these tools only provide additional information for the program manager to consider when making risk management decisions – they are not omniscient.*

CONCLUSION

In this paper, we have looked at what risk management is, the Riskit methodology for implementing risk management, and reviewed three risk management implementation case studies and the lessons learned in each of them. From the risk management summary, we have learned what a risk is and how to approach reducing the probability of risks occurring and reducing the impact on the project caused by these risks. From the review on implementing risk

management, we have seen various software tools and techniques illustrated that provide program managers with an additional aid when it comes to managing risks in a software project. We have identified that, with the proper training of program managers and team leaders in risk management methodologies and practices, we have the ability to 'jump' the necessary hurdles of loss, adverse project effects, and competitor advantages that come to us from risks through the use of risk handling actions. And, we have also seen that monitoring these risk handling actions is one of the most important aspects of the risk management process in order to ensure critical risks are being handled in an efficient way that reduces their likely hood of causing damage to the software project. Finally, in identifying the key focus areas drawn out of the case studies, we have acknowledged the critical areas in risk management that need to be implemented to ensure proper analysis of risks and risk handling actions are carried out effectively.

REFERENCES

- [1] Barry W. Boehm, "Software Risk Management: Principles and Practices" *Software, IEEE* January 1991, Volume 8, Issue 1, Page(s): 32 - 41
- [2] Keevin Fisher, George Greanias, Jim Rose, and Robin Dumas, "Risk Management Tools for Complex Project Organizations" *Aerospace Conference Proceedings, 2002, IEEE* Volume 2, Page(s): 2 - 721 – 2 - 727
- [3] Bernd Freimut, Susanne Hartkopf, Peter Kaiser, Jyrki Kontio, and Werner Kobitzsch, "An Industrial Case Study of Implementing Software Risk Management" *ACM SIGSOFT Software Engineering Notes, Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering*, September 2001, Volume 26, Issue 5
- [4] Jyrki Kontio, Gerhard Getto, and Dieter Landes, "Experiences in improving risk management processes using the concepts of the Riskit method" *ACM SIGSOFT Software Engineering Notes, Proceedings of the 6th ACM SIGSOFT international symposium on Foundations of software engineering*, November 1998, Volume 23, Issue 6
- [5] Jeffery G. Robinette and Janet S. Marshall, "An Integrated Approach To Risk Management and Risk Assessment" *Insight*, April 2001, Volume 4, Issue 1, Page 23
- [6] Jyrki Kontio, "The Riskit Method for Software Risk Management, version 1.00" CS-TR-3782, 1997, Computer Science Technical Reports, University of Maryland