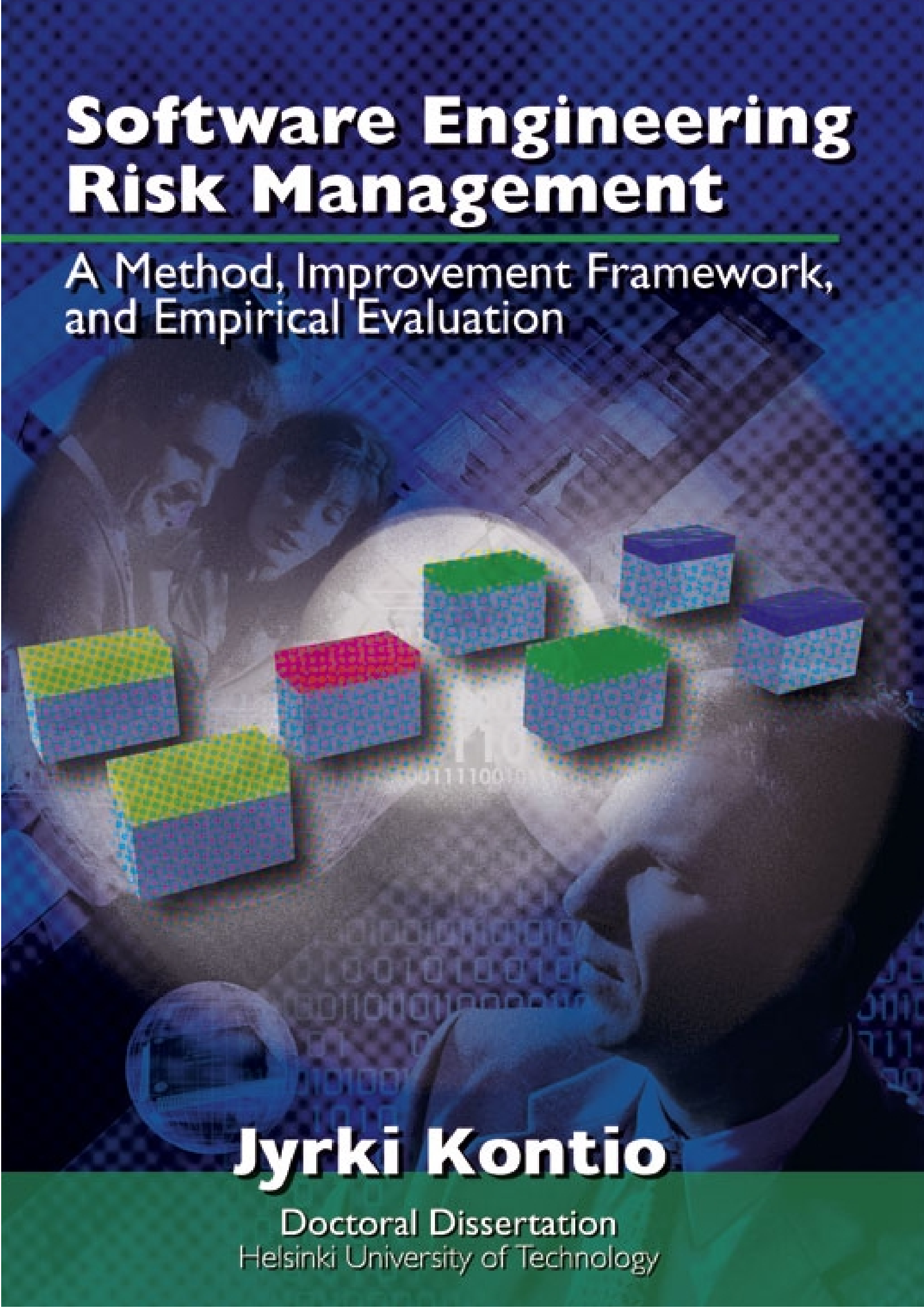
Software Engineering Risk Management

A Method, Improvement Framework, and Empirical Evaluation



Jyrki Kontio

Doctoral Dissertation

Helsinki University of Technology

HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering
Laboratory of Information Processing Science

Software Engineering Risk Management: A Method, Improvement Framework, and Empirical Evaluation

Jyrki Kontio

Nokia Research Center

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in the Auditorium T2, Computer Science Building, Konemiehentie 2 at the Helsinki University of Technology on the 28th of September, 2001, at 12 o'clock noon.

Kontio, Jyrki: ***Software Engineering Risk Management: A Method, Improvement Framework, and Empirical Evaluation***.

Keywords: risk management, project management, process improvement, software management, experience factory, quality improvement

Publisher:
Suomen Laatukeskus / The Center for Excellence
Purotie 1
00380 Helsinki
Finland

ISBN: 952-5136-22-1

Abstract

This dissertation presents a method for software risk management, its improvement framework, and results from its empirical evaluations. More specifically, our objectives were:

1. *Develop a comprehensive, theoretically sound, and practical method for software engineering risk management.*
2. *Develop a framework and supporting software tools for the continuous improvement of software engineering risk management and for improving knowledge about risks.*
3. *Evaluate the method in practice to provide information on its feasibility, effectiveness, advantages and disadvantages, and to improve it.*

Although risk management has been considered an important issue in software development and significant contributions to risk management have been made over the past decade, risk management is rarely actively and explicitly applied in practice. Furthermore, most risk management approaches in software engineering use simplistic approaches and fail to account for the biases common in risk perception.

We have developed a method, called *Riskit*, that complements existing risk management approaches by supporting qualitative and structured analysis of risks through a graphical modeling formalism. The method supports multiple stakeholder views to risks by considering their potential utility losses. The Riskit method is comprehensive, i.e., it supports all aspects of risk analysis and risk management planning in a software development project. We propose that our method has a sound theoretical foundation, avoids common biases in risk evaluations, and results in a more thorough understanding of the risks than traditional approaches.

Associated with the method, we have also developed a risk management improvement framework that supports continuous, systematic improvement of the risk management process. The improvement framework is based on the Quality Improvement Paradigm, and is supported by the eRiskit application. The eRiskit application supports the management of risks while simultaneously acting as a risk management repository that captures risk management data for improvement purposes. The eRiskit application also acted as a proof of concept for the correctness of the underlying concepts in the Riskit method.

We have validated the feasibility and effectiveness of the Riskit method in a series of empirical studies. The empirical studies were designed to provide characterization information and feedback on the method, as well as to act as initial validation of the method. The empirical evaluations showed that the method is feasible in industrial context and it seemed to improve participants' confidence in risk management results. In addition, our research indicates that industry needs sound, systematic, yet cost effective methods for risk management, a common and customized approach to improve communications within an organization, and support and enforcement of the common approach.

Preface

Risk management is an exciting topic that has practical implications in all aspects of life and business. My motivation in this research has been to deepen my understanding in area, embark on an intellectual learning journey, and to make an impact on software engineering.

One of the most gratifying things in this research has been the opportunity to work with many talented and experienced scholars and colleagues. Even though a dissertation is a demonstration of individual academic achievement, this work contains significant influences from the people that have shared their views and knowledge with me. I have used the plural pronoun "we" throughout this text not only because it is a common, scientific writing style – I have also used it because it symbolizes and acknowledges the fact that I could not have done this alone and that I am grateful for all the help and ideas I have received.

Professor Victor R. Basili of University of Maryland has been a great supporter of this work from the very beginning. His example alone has set me a very high academic standard to aim at and his advice has helped me to focus the research and to go the extra mile. Professionally, and as a friend, he has helped me to "do the right thing" over the past years.

There are several other people at University of Maryland's Experimental Software Engineering Group that have left a clear imprint on my this work. Dr. Carolyn Seaman reviewed some of my earlier papers, helped me in collecting some of the empirical data, and lead me to discover qualitative research approaches; Dr. Filippo Lanubile challenged and deepened my thinking about empirical study designs and validity threats; discussions with professor Adam Porter helped me phrase my research questions better, and Dr. Lionel Briand's early comments on my research encouraged me to pursue this research further. It was a pleasure to work with Gianluigi Caldiera as well – he helped me locate the first two case studies at NASA and Hughes. Professor Reidar Conradi from Norwegian University of Science and Technology provided valuable comments on the final version of the manuscript.

The empirical studies were a central element in this research. I am grateful to Helena Englund who assistef in making all the practical arrangements and analysis in first study at NASA. Thomas Gwynn of the Computer Sciences Corporation committed to this study and provided most of the empirical feedback in that study.

Cooperation with DaimlerChrysler's Research and Technology unit has been one of the most influential and fruitful empirical contributions to this work. Gerhard Getto made the initial contact and commitment and the interactions and discussions I had with him and Dr. Dieter Landes, as well as the additional discussions with Ton Vullinghs have been rewarding and helpful. Sven Seibold helped me finalize the design of the risk management database by providing several valuable comments. Kurt Schneider's feedback was central in clarifying and positioning the on the improvement framework better.

The Fraunhofer Institute for Experimental Software Engineering has an excellent and competent team working in software risk management. It was a pleasure and a challenging experience to work with them and apply the Riskit method in practice. Dr. Peter Kaiser established and coordinated our empirical study and Bernd Freimut was instrumental in analyzing the findings. I am also grateful to Werner Kobitzsch and Tenovis GmbH for using the method and allowing the publication of the main findings.

I have gained most of my professional experienced while working for Nokia in various responsibilities. Nokia is an exciting workplace with more than its share of top talent. Dr. JT Bergqvist and Dr. Pertti Lounamaa introduced me to Nokia and gave me many challenges that deepened my experience. I spent several years working with Pertti and his support and encouragement was crucial in finally deciding to pursue my doctorate studies. While working at Nokia Networks, Jussi Ilmarinen allowed me to take the time to complete some critical steps in this research. Working in Jussi's management team gave me hands-on experience in strategy definition and risk management in a very dynamic market and many of my colleagues in that team taught me practical insights of business management. However, it was Pete Pihko's personal interest and commitment that helped me conduct one of the critical empirical studies that resulted in concrete improvements in the method in this thesis.

I have also had the pleasure of working with people that improve Nokia's risk management practices, including Petri Toivanen, Mikko Routti, Michael Svedlin, Sari Rinne, and Timo Korvenpää. Their feedback and the possibility to discuss practical challenges of risk management have helped me understand the problems and potential solutions in risk management. Unto Kuivalainen provided valuable links to standards that cover risk. Working with the NoPETs team, especially with Lockhart Burck, Gilles Teissier, and Graham Honeywill, helped me understand how to implement process management in practice. During this time I also appreciated the possibility to work with late Frank McGovern – he set an example for integrity and persistence in getting things done.

Towards the completion of this work, I have received much support from Nokia Research Center, in particular from Kari Känsälä and Heikki Saikkonen. The last mile is the toughest one and their flexibility and patience allowed me to, finally, wrap this up.

The invisible part of research are the numerous hours a researcher spends seeking for articles and analyzing them. I could not have mastered the reference material I used – literally thousands of references – without the continuous support from the world-class professionals that run the Nokia Information Service. They sent me the article copies and books I asked for – and were most patient at times when I failed to return the material on time.

I have spent several years at the Helsinki University of Technology as a doctoral student and later as a part-time professor of software engineering. Professor Shosta Sulonen's friendship, personal support, encouragement, and guidance have been most valuable. His patience, together with his soft pressure "to wrap it up", has helped me finally complete this research. He duly deserves much of the credit for making this research happen.

Several colleagues at the Helsinki University of Technology have also shaped this research. My initial cooperation and discussions with Olli Pitkänen influenced the scoping and orientation of this work; and Marjo Kauppinen kindly reviewed some sections of this thesis, providing valuable comments. Johanna Lehtola has been most helpful in many practical arrangements that were involved in my work at the university.

The eRiskit application was designed and implemented by a group of talented and committed students: Joni Hahkala, Veli-Pekka Kröger, Esa Rosendahl, Ari Tervo, Matias Turkkila, and Sami Visti. Later, Ms. Hua Huang completed the graphical editing functionality in the application. A major part of this research would be missing without their contribution. I am especially grateful to Esa Rosendahl who continued to work with me at R & D-Ware Oy in developing the application further, playing a central role in many empirical studies, contributing to the method development, and packaging much of the other Riskit related material for better technology transfer.

I am grateful of the financial support that has made it possible to pursue this academic research. The Academy of Finland, the Finnish Cultural Foundation, Thanks to Scandinavia Organization, and the Research Foundation of Helsinki University of Technology have supported my family and me during this research. I am especially grateful to Esko Kervinen and professor Markku Kallio who coordinated the doctoral program by Union Bank of Finland that not only provided financial support but also arranged several excellent seminars that were helpful in the early phases of my degree. I would also like to express my gratitude to Tekes, who funded a study that clarified the industry risk management needs and commercial potential of the results presented here.

The final hurdles in a dissertation are the formal, scientific evaluations that the dissertation needs to go through to be accepted. I was honored to have some of the most acknowledge experts in their fields to examine this work. Professors Timo Saarinen (Helsinki School of Economics and Business Administration) and Kalle Lyytinen (Case Western Reserve University) reviewed the thesis and gave valuable additional suggestions for improvement. My opponent, professor H. Dieter Rombach, has been a role model and an example of a world-class researcher. Regardless of the outcome of the dissertation defense, I am grateful for him agreeing to spend his time and intellect to review this research and to provide guidance for further work in this area.

All the previously mentioned individuals and organizations have been fundamental in supporting this intellectual, academic endeavor to deepen my understanding and academic skills. However, I also need to acknowledge the support and encouragement that I have received from my friends and family. My mother, Kerttu Kotila, has helped our family by taking care of our children on the many occasions when both Katri and I had to concentrate on our work. Jouko Manninen provided invaluable advice and guidance on the statistical analysis methods used in one of the studies. Our children, Mirva, Elias, and Oliver, have been reasonably patient at times when I had to concentrate on this research instead of spending time with them. They have also been quite interested in understanding what I am doing, why I am doing it, and when I am going to be done with it. Quite honestly, their questions lead me to understand my motives better and put this work into proper perspective.

My deepest gratitude and love goes to my wife, Katri. Over the years, she has had to take the extra effort and responsibility to do many of the practical things in our family, and tolerate my mental absence due to this work. On many occasions, she has also helped me by not allowing me to work on this thesis, and do something else instead. I value her support, patience, and care during this research. She is my trusted guide and companion on a journey that is far more challenging and rewarding than a mere academic milestone.

Jyrki Kontio

Espoo, September 10, 2001

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Background and Motivation

The world is runs on software. Practically all fields of human activity are increasingly dependent on software to support, operate, or control machinery, information flows, records, or processes. An increasing number of products contain embedded software, people who design them use software to plan and engineer the product, the manufacturing process is planned and controlled by software, the logistics chain that delivers the product to the customer is dependent on software and databases, and, finally, the monetary transaction for the product takes place through the software and databases of financial institutions. Our ability to develop and run software strongly influences how, or actually, whether the society functions.

Not only has the role of software increased substantially over past decades, the software development industry has grown into a large segment of the economy, the world-wide value of information and communication sector has been estimated to be 320 billion Euros (Anon.1998b). In the U.S., the growth rate of software industry has been 2.5 times the growth rate of the economy as a whole (Nukari & Forsell 1999), and the information technology sector accounts for a quarter of the economic growth in the U.S. (Anon.1998a). Software is being developed in projects that employ thousands of people across companies, countries, and time zones. Furthermore, the importance of software is likely to continue to increase. The Internet alone is an environment that operates on software and accelerates the distribution and use of software, the so called new economy will further enhance the business opportunities based on software (Shapiro & Varian 1998), and intelligent appliances for individuals will make software even more present and common in the society.

Software projects have turned out to be difficult to manage within time and budget. The software industry has a long track record of overrun budgets, missed deadlines and lacking functionality, reports of such problems have been presented over several decades (McFarlan 1974; Rothfeder 1988; Charette 1989; Flowers 1996; Glass 1997). In fact, the term "software crisis" has been used to describe the state of the practice in the industry: projects continually fail to meet the increasing demands for better quality, higher productivity, and greater functionality. While there are undoubtedly many "runaway" projects, it is questionable whether the word crisis is, still, the appropriate term to describe the state of software engineering practice. One can argue that the reported disaster projects represent extreme cases and the large mass of successful or nearly successful projects simply do not pass the publishing threshold (Glass 1998). Nevertheless, the large volume of software development and at least the very big potential for disasters call for improvements in the way software development is planned and managed.

Of the many engineering disciplines, software engineering is more prone to risks than many other areas. In software projects, processes and requirements evolve more, complexity of products is higher, and there are a higher number of potential risk factors than in many

other disciplines (Fairley 1989). This is partially due to the inherent nature of software development, in principle, all projects are more unique than those of other disciplines: identical software can be copied at no cost whereas similar physical components still need to be manufactured. This uniqueness makes it more difficult to plan, model and predict the progress of a software project.

Both Fairley (Fairley 1989) and Brooks (Brooks jr. 1987) have highlighted the complexity of the software engineering as one of its specific characteristics. While abstraction is often used to deal with the size of systems and to some degree with complexity, abstraction can only provide partial views to the complexity of a system, it cannot abolish complexity. Complexity is a typical characteristic of software and developing complex systems is difficult and dealing with this complexity makes us prone to mistakes.

Brooks (Brooks jr. 1987) also pointed out that software engineering is a man-made discipline that does not have any universal constants or "natural laws" that would provide a clear theoretical platform or anchor points for the discipline. Many of the standards and practices in software engineering have been established or agreed upon by de facto market domination or by negotiation process by key players in the industry. As a result, these standards and "laws" are not necessarily compatible with each other or constant.

Software is linked to many other human and organizational systems that evolve over time (Brooks jr. 1987). The interfaces, required functionality, and system architecture are all designed and programmed explicitly for a given environment. As these other systems evolve, software also must also evolve, even during a development project. The changing environment creates additional uncertain elements in a software project, again increasing the potential for risks to surface and occur.

The invisibility of software (Brooks jr. 1987) also contributes to the risks in software projects. Invisibility makes it harder for people to understand all the relationships between different components and aspects of software.

Software is also a young field and the cumulative experience of the community is far less than almost in any other engineering area. While software programs have been written in industrial scale since the late 1950's, the software engineering discipline began to emerge in the late 1960's and even today the software engineering community debates whether it has truly become an engineering discipline (Basili 1996; Pfleeger 1997; Tichy et al. 1995; Tichy 1998; Zelkowitz & Wallace 1998). The lack of empirical, historical measurement data also makes it difficult to understand and manage the software engineering process.

Software engineering is a field with very rapid technology development cycles. The hardware platforms, systems standards, and development tools are under a constant change (Selig 1993) and new technology is adopted with less rigorous evaluation than in other fields (Fenton et al. 1994; Tichy 1998; Zelkowitz & Wallace 1998). It is not uncommon that the technological platform is changed in a middle of a project. Each new technology will bring along new, possibly unknown risks to a project and it may invalidate the previously accumulated risk knowledge. Traditional engineering fields benefit from several engineering cycles that gradually perfect the designs and eliminate potential problems (Petroski 1985; Fortune & Peters 1995). This is more difficult in software development as designs are less often reused and subjected to such repeated reviews.

User expectations from software are also continuously increasing as people are using different applications and are exposed to new opportunities offered by software. These

growing expectations decrease the stability of a software project, again opening a door for potential risks.

Finally, the software industry has recently become truly international as far as the competition is concerned. Even a small software company must compete with alternative solutions that may be offered by international industry giants that have much larger development resources. The Internet has also opened competition much wider: many customers can easily access and consider competing alternatives from geographically distant locations. The increased competition encourages, if not force, software companies to take bigger leaps in their software development, leading to greater risk taking.

As we will discuss in chapter 2.3, the software engineering community has become aware of the importance of risk management and several risk management approaches have been proposed and are in use. However, some surveys indicate that the industrial practice of risk management is informal and infrequent: according to Ropponen's survey, 75% of the surveyed project managers did not use methods to identify, evaluate, or control risks (Ropponen 1993). The limited survey data from a workshop by Basili and Torii supports this: only 20% of respondents claimed to use risk management techniques "extensively" while 40% stated that they are not using "any risk management techniques or approaches" (Kontio 1995a). Clearly, the industry practice in applying risk management methods seems to be less than the significance of the problem would suggest.

We believe there are several factors that partially explain the situation. First, it seems that risk management is often performed in an implicit fashion under the general umbrella of project management (Chittister et al. 1992). They may not consider this type of risk management as something they would report in a survey. We believe that such an ad hoc, implicit approach may be sufficient in situations where people involved in risk management have been exposed to the domain for a long time, the complexity of the domain is low, and the past empirical experience can effectively build up a correct understanding of risks. As we discussed earlier, software engineering is complex and dynamic field where new and increasingly large projects are being managed by people who may not have sufficient amounts of relevant experience. Therefore, we believe, an explicit and systematic risk management is not only beneficial but also critical for software development organizations. This view is supported by many other contributors in this field (Boehm 1989; Charette 1990; Charette 1989) and the Boehm's spiral model even advocates that the risk management is the primary driver in project management (Boehm 1991).

Second, project managers and management teams are constantly under time pressure and this simply limits the time available for any activities that do not have an immediate and concrete benefit to the project. Unfortunately, as risk management solves future problems, other urgent issues often get priority. Clearly, by investing in preventing future problems there are potential gains that should help organizations avoid the fire-fighting mode.

Third, practitioners may not be aware of the available methods and techniques for risk management. Lacking easy to use tools, they simply may not perform risk management. Fortunately, several methods are easily available and making them available to project personnel is simply a matter of executives' decision to deploy such techniques in the organization.

Fourth, some organizations have a culture that discourages bringing up negative issues or risks. Such climate may effectively prevent explicit discussion of risks. If risks are ignored due to this, organization will refrain from taking preventive action and unnecessary risks may

occur, causing problems and financial losses to the organization. Correct and focused risks management activity will not conflict with such a "winning culture". Instead, it should give people more confidence in being able to reach their goals.

Finally, it is quite difficult to measure the benefits of risk management: successful risk management may prevent some problems from occurring and very few organizations have good enough measurement systems or data points to show the impact of risk management. Even then, a successful risk management practice actually might not even reduce the damage that occurs in a company if the improved practice simply allows the organization to take bigger risks that yield higher profits. This last point may be the biggest hurdle preventing effective risk management to take place in an organization – if the management is not convinced of the benefits of risk management, they will not support and enforce it in an organization.

Many risk management approaches have been proposed in the software engineering field since 1980's when Barry Boehm (Boehm 1989) and Robert Charette (Charette 1989) brought the discipline into the mainstream focus in the field. However, as we will discuss later in chapter “2.4 Limitations of Current Approaches”, most of the existing risk management approaches have severe theoretical and practical limitations that may have contributed to their slow acceptance by the industry.

The software industry is operating in an increasingly risky business environment and risk-taking is required for success (Charette 1999), yet the industry practice of risk management is largely either absent or based on limited and biased practices. The underlying motivation for this work is to improve the industry practice in software engineering risk management. We hope to make a contribution that will allow more effective control of risks, leading to improved success rate of even more challenging projects.

We have articulated our motivation and view of the industry needs as a set of suppositions that have been used as a basis of our work:

S.1 More effective risk management methods and techniques need to be used in software projects to improve projects' success rate.

S.2 Risk management methods in software development must be systematic and explicit so that results are transparent and communicable to various participants and stakeholders of the project.

S.3 Risk management methods used in software projects must have low overhead and be able to produce concrete results quickly in order to be used in practice.

S.4 Benefits of risk management must be demonstrated and measured more effectively so that the industry decision makers become more concretely aware of the importance of risk management.

In this work, we have developed a risk management method that avoids limitations common to many current approaches while being a practical and feasible approach in industrial projects. The risk management improvement framework presented in this work provides examples and guidelines to establish a continuously improving risk management system into an organization. We also report on the empirical studies that applied this method. These studies acted as initial validation of the method that we developed, as well as provided empirical feedback on the method itself and on risk management in general, allowing us to improve our understanding of risk management and to improve the method itself.

1.2 Key Concepts

We have defined most of the terms and concepts as they are used in the text. In addition, we have also provided a general terminology in Appendix C. In this section, we present some key terms that are necessary to communicate the scope and contributions of this work.

We define *risk* as a possibility of loss, the loss itself, or any characteristic, object, or action that is associated with that possibility. In other words, we are primarily concerned with the negative consequences of potential future events. There are two important points to make regarding this definition. First, it is a different definition from the one often used in finance and economics, where risk is essentially defined as a volatility of a financial instrument over time (Crouhy et al. 2001; Williams et al. 1998). Second, even though our definition emphasizes the negative consequences, our approach also includes and supports the modeling of positive consequences of future events, i.e., risk management can also be seen as a way of recognizing and managing opportunities.

Our definition of risk also extends the traditional view of risk by including explicit links to expectations – or goals – and stakeholders, as shown in Figure 1. Risk is characterized by a probability and loss associated with it. The losses are defined by what the expectations or goals are, and they, in turn, are defined by the stakeholders that may have different expectations. Thus, definition and quantification of a risk requires that the expectations and stakeholders are known. We have included a more detailed discussion on the definition of risk in chapter 3.2.

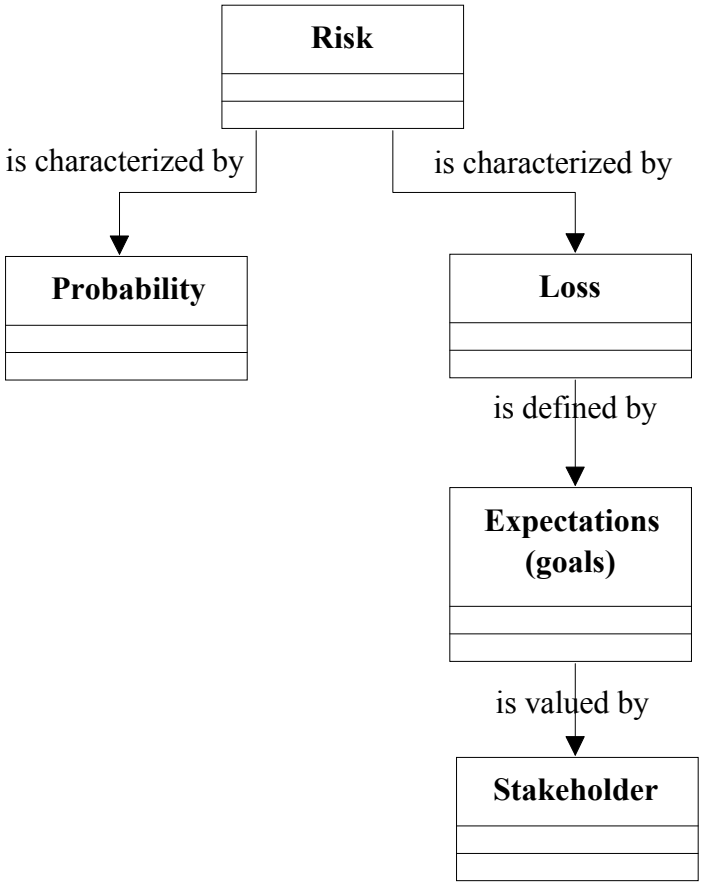


Figure 1: Definition of risk in the Riskit method

The term risk management has been used in two meanings in the software management literature. On one hand it has been used to refer to the *discipline* that that studies how to identify, address, and eliminate risks (Boehm 1989), on the other hand it has been referred to as the activity or process that attempts to identify what could go wrong in a project and taking steps to avoid such problems in advance (Charette 1989; Dorofee et al. 1996; Hall 1998). In this work, we use the term *risk management* to refer to activities that are taken to identify, analyze and control risks. We use the term *risk management process* to refer to a systematic and explicit risk management activity. The risk management process can be seen as having the inputs and outputs presented in Figure 2: the input of the process consists of project context information, goals, and plans for the project. Risk management methods and tools are used to support risk management, and the process delivers two outputs: understanding of the risks and risk controlling actions.

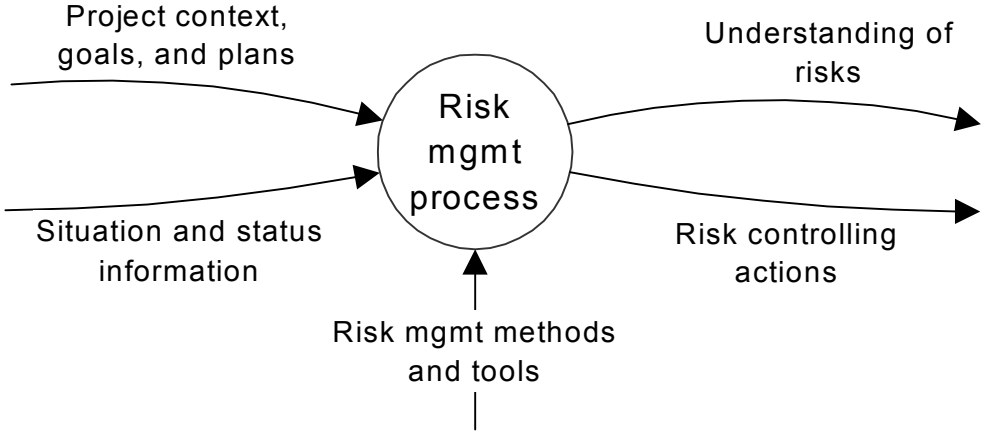


Figure 2: Risk management process inputs and outputs

One of the contributions of this work is the improvement framework for risk management. We define an *improvement framework* as a collection of concepts, high-level processes, and examples that can be used to build or instantiate a concrete improvement system in an organization.

The terms data, experience, and knowledge are all relevant to improvement models. *Experience* can be defined as (i) "active participation in events or activities, leading to the accumulation of knowledge or skill: a lesson taught by experience; a carpenter with experience in wall and roof repair" and (ii) "knowledge or skill so derived" (Anon.1992). Traditionally in the Experience Factory context the term experience means the second definition, i.e., "knowledge or skill so derived". *Knowledge* is defined as "familiarity, awareness, or understanding gained through experience or study" (Anon. 1992). In other words, knowledge is something that we try to synthesize from the experiences we have had (i.e., experience) or from other sources. We can construct good experiments in order to have more leverage to formulate knowledge.

This definition is slightly problematic as we lack the term for definition (i) of "experience", i.e., for raw experience that has not been abstracted and formalized. Yet, the "raw data" vs. "packaged knowledge" is essential in the process improvement. The situation is highlighted on the left-hand side of Figure 3 where the two different meanings of the word "experience" can be used. As the analysis and processing of experience (i) into

experience (ii) is a fundamental concept in process improvement, these should be differentiated.

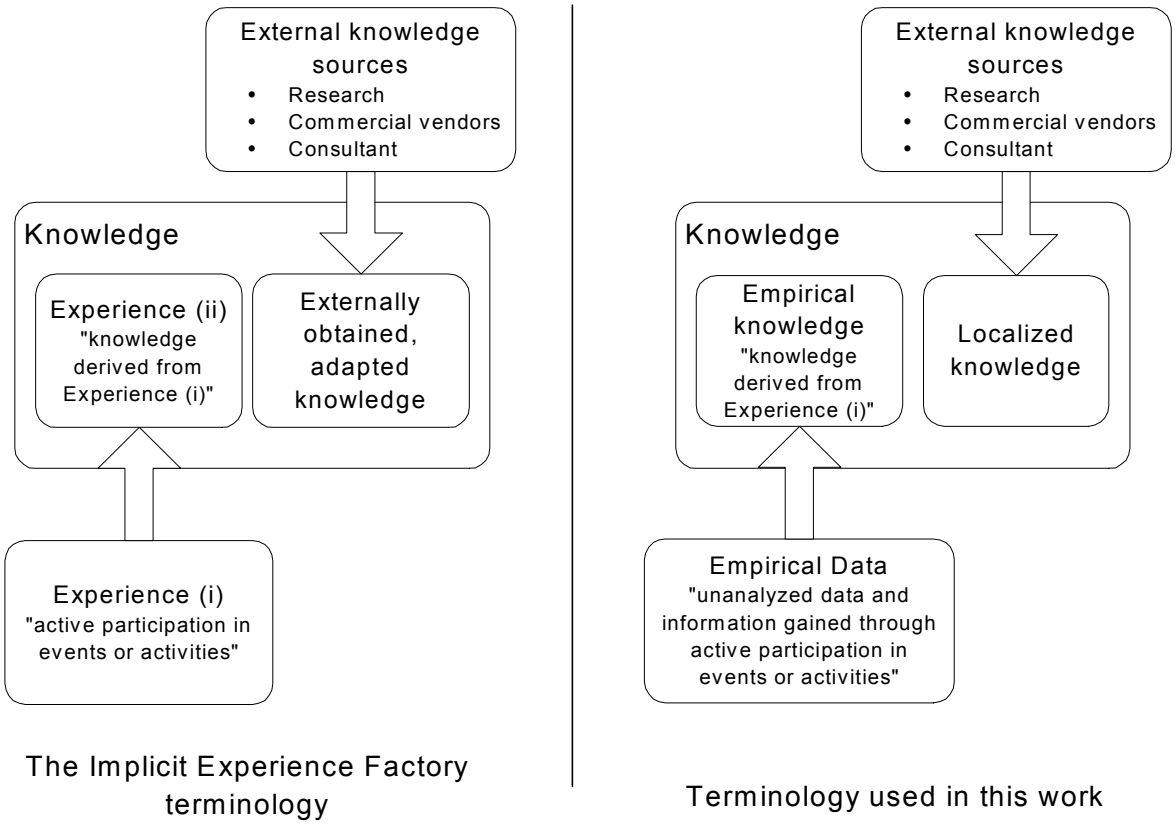


Figure 3: Definition of data, experience, and knowledge

In this work, the term raw experience is used to mean any un-analyzed information or data that has resulted from active participation in events or activities. This can include measurement data or project personnel experiences (definition (i)). We are using the term "knowledge" to refer to any information that has been formulated to be used in software development, regardless of its origin (empirical or external). If necessary, we use the terms "empirical knowledge" and "externally obtained knowledge" to refer to the origin of knowledge. In order to avoid potential confusion, the word "experience" is reserved to refer to union of empirical knowledge and raw data. These concepts are visually presented in the right-hand side of Figure 3.

1.3 Research Problem and Objectives

The overall goal of this work is to improve industrial practice of risk management by developing and providing improved methods, tools, and insights to support software engineering risk management. We have defined the research problem in this work into three main objectives, as listed below:

1. *Develop a comprehensive, theoretically sound, and practical method for software engineering risk management.*
2. *Develop a framework and supporting software tools for the continuous improvement of software engineering risk management and for improving knowledge about risks.*

3. *Evaluate the method in practice to provide information on its feasibility, effectiveness, advantages and disadvantages, and to improve it.*

These three research objectives are discussed in more detail in the following.

1.3.1 Develop a Risk Management Method

The starting point for developing the risk management method and tool was to synthesize findings and requirements from the literature as well as from our personal experience in risk management. We surveyed the literature and synthesized our previous experience in risk management to formulate specific requirements for the method we developed (Caplan 1994; Chittister et al. 1992; Diekman 1992; Edgar 1989; Garrabrants et al. 1990; Garrick & Gekler 1991; Hall 1994; Hefner 1994; Kahneman et al. 1982; Kontio 1994a; Kontio 1995a; Lyytinen et al. 1993; Meyers & Trbovich 1993; Rook & Cowderoy 1993; SEI 1993; SEI 1994; SEI 1995; Simister 1994; Williamson 1994). We used the criteria proposed by Garrabants et al (Garrabrants et al. 1990) as a basis for the development requirements we set for the method development and reviewed them from the perspective of our own experience in risk management (Kontio 1994a). Our own experience in risk management lead us to add two requirements to the list proposed by Garrabrants. First, risk management in an organization is a communication challenge: different perceptions and opinions need to be shared and consolidated during the identification and analysis and the results of risk analysis need to be communicated with several stakeholders. Second, cost effectiveness is a prerequisite for a risk management method to be used in practice: project managers have severe time constraints and risk management actions need to provide added value with limited effort. Therefore, we added need for communication and cost-effectiveness to the method requirements, modified the criteria slightly for our purposes, and synthesized the risk management method requirements as presented in Table 1.

R-1	Consistency: independent users should apply the method in similar way and get the same results in the same situation.
R-2	Usability: the method should be easy to learn and use.
R-3	Adaptability: the method can be applied to different situations and projects.
R-4	Feasibility: The method should be concrete and feasible in practice.
R-5	Completeness: the method should support all risk management activities and aspects in a software development project.
R-6	Validity: The modeling approach and the results of the process should represent the real phenomenon.
R-7	Credibility: the method should increase confidence in the validity of risk analysis results.
R-8	Communications: the method should support communication about risks.
R-9	Cost-effectiveness: the method should produce added value to projects within reasonable cost and effort.

Table 1: Requirements for the method development

The Riskit method was developed to satisfy these requirements. However, the empirical studies (Englund 1997; Freimut et al. 2001; Getto & Landes 1999a) (Getto & Landes 1999b; Kontio et al. 1996; Kontio & Basili 1997; Kontio et al. 1998)conducted during this development provided additional feedback and requirements to the method and the method

has been revised to include such feedback. The Riskit method is presented in chapter 3 and the results of empirical studies are presented in chapter 5. We are using the Promises framework (Kontio 1995b; Kontio 1998) to present both the Riskit method and the risk management improvement framework.

1.3.2 Develop an Improvement Framework

As with the development of the Riskit method itself, we followed a similar approach for the development of the improvement framework for the method. We surveyed the literature (Basili 1989; Basili 1993; Bhandari et al. 1993; Bollinger & McGowan 1991; Card 1991; Curtis & Paulk 1993; Dion 1992; Humphrey & Sweet 1987; Thomas & McGarry 1994), reflected on our own personal experience in process improvement, and synthesized the requirements for the risk management improvement framework.

Basili's Quality Improvement Paradigm (QIP) forms a foundation of the Riskit improvement paradigm (Basili et al. 1992b; Basili & Green 1994). In this work, we consider the QIP as a paradigm that has three main components: the Experience Factory Organization, the Experience Base, and the QIP cycle, discussed in more detail in chapter 4.2.

We selected the QIP as the underlying paradigm for Riskit process improvement as it emphasizes the continuous improvement though empirical data collection and studies and for providing a comprehensive framework for process improvement. In particular, the subjective nature of risk understanding seems to match well with the QIP philosophy of localizing knowledge.

We also considered alternative improvement paradigms for the Riskit process improvement process. Two major paradigms were considered: maturity models, in particular the CMM (Paulk et al. 1993a) and SPICE (Anon.1998c); and Hall's risk management maturity model (Hall 1994; Hall 1995). CMM and SPICE were rejected because they lacked sufficient details on risk management and for representing a static view on key aspects of process improvement (Kontio 1995b). Hall's maturity model was rejected as it also represented a static view on process improvement and for the lack of evidence for its validity. However, note that these different paradigms are not necessarily conflicting with the QIP. While they differ in many key aspects, they can be used to complement and extend risk management process improvement, as they do address several aspects of risk management.

The Experience Factory is based on learning from internal experience, i.e., it uses the principle of systematically collecting experience and basing improvements on the insights gained from these experiences. This is a fundamental difference from the assessment models described earlier, the Experience Factory assumes that there is not necessarily a universally applicable, correct model for software development and, therefore, it is necessary to accumulate experience to learn the characteristics of one’s own software development domain.

Based on our survey and analysis of our own experience, we formulated the requirements for the risk management framework as presented in Table 2.

The improvement framework for risk management was developed by instantiating and customizing the Experience Factory concept for risk management. This essentially required the definition the following aspects of the framework:

- Definition of the risk management improvement process, including identification of information flows.

- Definition of roles and responsibilities in the process.
- Definition of information types and items to be created, used, and captured during the process
- Definition and implementation of a software tool to support the execution of the risk management process, as well as the capturing of the information created during the process.
- Definition of a set of empirical designs and supporting documentation.

The risk management improvement framework is presented in chapter 3.7.

I-1	Continuous learning cycle: The framework should support continuous learning and feedback for improvement.
I-2	Complete improvement process: The framework should contain a complete process definition for process improvement.
I-3	Empirical learning: the framework should support learning from local experience.
I-4	Capturing of risk management data: the framework should support efficient and meaningful capture of risk management data and raw experience to support learning.
I-5	Clarification of roles and responsibilities: the framework should give clear guidance on the responsibilities of individuals enacting in the improvement process.
I-6	Definition of possible knowledge repositories: the framework should identify the typical and needed repositories of knowledge.
I-7	Traceability: the framework should support traceability of experience and knowledge so that interpretations and conclusions can be re-evaluated and revised if necessary.

Table 2: Requirements for the improvement framework

1.3.3 Empirical Evaluation

The third objective of this work was to evaluate the method and the improvement framework in practice to provide information on their feasibility, effectiveness, advantages, and disadvantages; and to improve them. This was done in a series of empirical studies that covered various aspects of the work. Table 3 lists these studies and explains their contribution to the work.

The empirical part of the work (chapter 5) explains and details the study designs used, discusses validity issues, reports the data, and analyzes the results.

1.4 Scope of Research

This work focuses on risk management in software development projects or programs. More specifically, we are studying and developing methods, tools and techniques for people involved in risk management in software projects. The development of the method and its improvement framework included the definition of process, information flows, roles and responsibilities, information types, and the templates used in the process.

The results of this work are intended to be applicable to any kind of software development, as the wide range of application domain covered by the empirical studies suggests. However, we believe that large organizations and complex projects are more likely to benefit from the

results of this research. Small organizations or trivial projects will be able to conduct their risk management more intuitively.

While the domain of this research is software development, we believe that the methods developed are quite applicable to any goal-oriented activity, such as product development in general, project based industries, manufacturing and building industry. However, to validate such claims is beyond the scope if this work. We also believe that the basic concepts of our results can also be used in business risk management and strategic planning but – again – our intent is not to study such application opportunities in this research.

We have surveyed several other fields in our literature survey part of our research. We have included contributions and insights from economics, psychology, organizational development, government, and engineering to construct our method and tailor it to software engineering domain. However, it has not been our intent to verify whether and how our findings could be applied in these other fields.

Study	Objectives	Main Contributions
Study 1: Focus Groups	• Identification of main issues in corporate risk management.	• Research motivation and presuppositions confirmed • Model of industry needs in risk management
Study 2: Exploratory Study at NASA (Kontio & Basili 1997)	• Evaluate feasibility of the method • Feedback for method development	• Confirmation of method feasibility • Initial efficiency and effectiveness indications • Revisions to Riskit Analysis Graphs • Method revisions
Study 3: Characterizing Study at Hughes	• Describe and understand the goal review and risk identification steps • Feedback for method development	• Guidelines for goal review • Definition of the Riskit risk identification approach
Study 4: Nokia and DaimlerChrysler Study	• Evaluate the Riskit method in industrial projects. • Feedback for method development • Understand issues in introducing risk management into software projects.	• Confirmation of the feasibility of the method • Indications of efficiency and effectiveness of the method
Study 5: Method Introduction Study with IESE and Tenovis	• Analyze the usefulness and adequacy of the Riskit method. • Analyze the cost-benefit of the Riskit method in industrial context.	• Confirmation of feasibility • Indications of efficiency and effectiveness of the method • Benefits and disadvantages characterized • Guidelines for introducing risk management into projects
Study 6: Risk information visualization study	• Compare different techniques for capturing risk information.	• Detailed characteristics and comparisons between modeling approaches
Implementation of the eRiskit software	• Demonstrate the correctness of the conceptual model of risk in Riskit through software implementation • Understand the potential of software support for Riskit	• Proof of concept for the Riskit model of risk • Improved understanding of the potential benefits and problems in software support for Riskit

Table 3: List of empirical studies and their objectives

When dealing with software risks, the risks caused by the *operation* of software are often also considered. As our primary focus has been to support software development organizations in *developing* their products, we have excluded the evaluation and management of risks that might occur during the operation of software. Again, it is plausible that many of the underlying principles and methods developed in this research are also applicable to risks in operating software, but in order to control the focus of this research we have not addressed such issues.

A critical issue in software engineering risk management work is the identification of generic, or typical, risks in specific subdomains, understanding their characteristics, and providing support for controlling these risks effectively. As several other researchers are addressing these issues (Carr et al. 1993; Jones 1994; Laitinen et al. 1993; Monarch et al. 1996; Ropponen & Lyytinen 2000), we have not attempted to include such topics into the scope of this work. However, we believe that both the Riskit method and the risk management improvement framework presented in this work provide a basis for researching common or domain specific risks more effectively.

1.5 Contributions

This chapter presents a summary of the scientific and engineering contributions made in this research. We briefly describe them and refer to the specific chapters that discuss the contribution in more detail later in this work.

The primary objective and contribution of this work was the development of a comprehensive method for software risk management. As we will discuss in chapter 3, the method has several characteristics that are not found in other methods. These characteristics include a comprehensive process definition, integration of stakeholders and goals into the risk management process, use of utility theory and prospect theory in risk prioritization, approaches to control common biases regarding risk analysis, the identification of a risk controlling action taxonomy, as well as the definition of various information types and templates for them to support the risk management process. These characteristics as a whole are novel and, as our empirical findings indicate (chapter 5), they seem to improve the credibility and impact of risk management. In particular, to our knowledge, Riskit is the first method to integrate stakeholders, goals, and risks into the risk management process so that their relationships can be understood and kept up-to-date in practice.

A specific contribution of this work is the Riskit Analysis Graph (see chapter 3.5), a graphical modeling notation that has a well-defined underlying semantic model, yet it is practical and easy to use in industrial context. The Riskit Analysis Graphs allow accurate and unambiguous modeling of risks without sacrificing clarity and understandability of the risk information. While some simpler, similar formalisms have been presented, we believe that the Riskit Analysis Graphs are novel in combining well-defined formalism with representational clarity.

The research also produced the eRiskit application, a web-based application that supports identification, analysis, and controlling of risks, as well as the tracking and archiving of risk status and situation. The eRiskit application is fully Riskit compliant, providing additional support for the use of the method as well as supporting the risk management improvement framework. Its technical characteristics are also somewhat novel as it is fully Internet based, allowing users to cooperate and share risk information through their web browsers. The eRiskit application also acted as a proof of concept for the Riskit method, demonstrating that

the key concepts of the Riskit method can be formalized into software and database application.

The risk management improvement framework is another main contribution of this research. Based on the adaptation of the Experience Factory model for risk management context, the improvement framework is the first comprehensive improvement framework for risk management. It complements the static risk management improvement models by providing an experience -based improvement paradigm.

The improvement framework and the eRiskit application together also form another contribution of this work: we have defined risk management data and knowledge repositories that are needed to perform risk management and to improve risk management practice (see chapters 3.7 and 4.5). Based upon a search of the literature and a study of available commercial products, the risk management meta-model is the first comprehensive model of risk management information and is more comprehensive and more powerful than the models contained in the commercial products we have evaluated.

The process model included in our improvement framework is also detailed process model of the activities, artifacts, and information flows in an Experience Factory. While refining the Riskit Experience Factory we have made several small contributions and solutions that extend and detail the Experience Factory concept. This process model can be used as a reference framework or example for implementation for organizations implementing the Experience Factory. Our explicit articulation of the EF and QIP paradigm is also a contribution that will help researchers in this paradigm understand it better.

The empirical part of our work has produced empirical evidence of the benefits and disadvantages of the Riskit method, as well as additional insights on risk management in general. The series of empirical studies performed in this research are among the most extensive empirical studies in software risk management reported in the literature. While these findings are discussed in the empirical study conclusions and in chapter 6.1 in more detail, they are summarized below:

- Riskit is a feasible and effective method for software engineering risk management.
- The Riskit Analysis Graphs were perceived as effective, easy to use, and accurate way of capturing and analyzing risk information.
- Stakeholders and goals strongly influence risk management decisions and Riskit method can maintain links to them throughout the risk management

We also made some general conclusions and recommendations about software engineering risk management. They are discussed in more detail in section 6.4, but the main aspects are summarized below:

- Industry needs systematic and sound, yet easy and cost effective methods for risk management. However, most of the methods currently in use are biased and not used consistently.
- Risk management needs to be supported and enforced to be effective.
- A common risk management approach will result in more consistent and effective risk management practice.

Our empirical findings of the current use of risk management methods in industry have contributed to the better understanding of academic and industrial challenges in risk management, augmenting the earlier surveys and studies in this area. The empirical study designs that were used can also be considered a contribution that will help other researchers

replicate the studies and thus contribute to the body of knowledge on software engineering risk management.

The Riskit process and risk management improvement framework were presented in this work using the Promises framework for process management. Thus, chapters 3 and 4 in this work demonstrate the use of the Promises framework and contribute to supporting its feasibility as a process modeling framework.

During the development of our improvement framework we also integrated key concepts and approaches from business process re-engineering field and software process improvement field by defining roles and responsibilities of risk management process owners in the Experience Factory. Such integration may help apply and leverage the contributions in these fields into practice.

The literature survey of this work provides a synthesis of applicable findings from other fields and an up-to-date summary of recent findings and advances in software risk management.

Finally, we have made some contributions in the research methods. First, we have made several small enhancements and adaptations to the GQM method (discussed in chapter 1.6). Second, the detailed documentation of the empirical study designs and arrangements act as examples for further similar studies or replicated studies in this field. Third, our adaptation of the research life cycle can also be seen as a contribution that will help other scholars plan and structure their research using scientific principles. This is also supported by the extensive set of references included here so that other researchers can refer and use such methods in their research.

1.6 Research Methods

This chapter summarizes the current practice in software engineering research methods, discusses various research methods and approaches, discusses specific challenges in the research into risk management, and presents the research methods used in this research. Note that the details of each research approach and methods have been presented in the chapters where the empirical studies have been presented.

Software engineering has traditionally been a constructive engineering discipline, i.e., a research field where the building and development of models and hypotheses has been the focus of researchers and less attention has been placed on the empirical evaluation or testing of such constructs (Potts 1993). There are several reasons why the constructive research model has prevailed, including the academic history and origin of computer scientists from mathematics departments and the high cost of empirical work in software engineering (Glass 1995b). The academic standards in software engineering seem to allow research with little empirical evidence – according to Tichy et al., half of the sampled research papers in software engineering had no empirical validation and only 20% of the sample papers had more than a fifth of the paper devoted to reporting empirical findings (Tichy et al. 1995), according to Zelkowitz, a third of the sampled software engineering papers had no experimental validation at all (Zelkowitz & Wallace 1998). Clearly, practitioners and researchers will benefit if more research that is empirical is carried out and their results shared.

The research approaches in software engineering can be categorized into four main categories (Adrion 1993; Glass 1995b; Zelkowitz & Wallace 1998):

- The scientific method, i.e., based on observing a phenomenon, a theory is developed, hypothesis formulated, test the hypothesis using measurements and data, and replicate the validation.
- The engineering method, i.e., a solution is developed; it is tested, based on the test results the solution is improved until no further improvements are required or feasible.
- The empirical method, i.e., hypothesis is formulated, it is tested in an empirical study, and data is analyzed to assess the validity of the hypothesis. Compared to the scientific method, the empirical method does not include theory or solution building.
- Analytical method, i.e., a theory is formulated, results derived from the theory, and results are compared with empirical observations.

The scientific method represents the ideal and complete approach to research, the engineering and empirical methods focus on different stages of theory building and validation, and the analytical method is applicable in some theoretical fields, even though it is a considerably weaker research method in fields that are empirical. However, it has been widely argued that software engineering is essentially a discipline that has a strong need for empirical research and the field cannot progress without better empirical orientation (Curtis 1980; Basili et al. 1986; Fenton et al. 1994; Adrion 1993; Kitchenham et al. 1995; Votta et al. 1995; Basili 1996), and similar concerns have been expressed in information systems research (McFarlan 1984; Järvenpää 1988; Lee 1989; Benbasat & Nault 1990; Keen 1991; Galliers 1992; March & Smith 1995; Benbasat 1999). While several authors have also expressed concerns whether information system development area can actually apply the scientific method in its research, even these authors acknowledge the need to perform empirical work and use suitable methods (e.g., (Fitzgerald 1991; Galliers 1985)). Therefore, our goal in this work was to conduct our research using the scientific method and performing empirical studies to develop and validate our models.

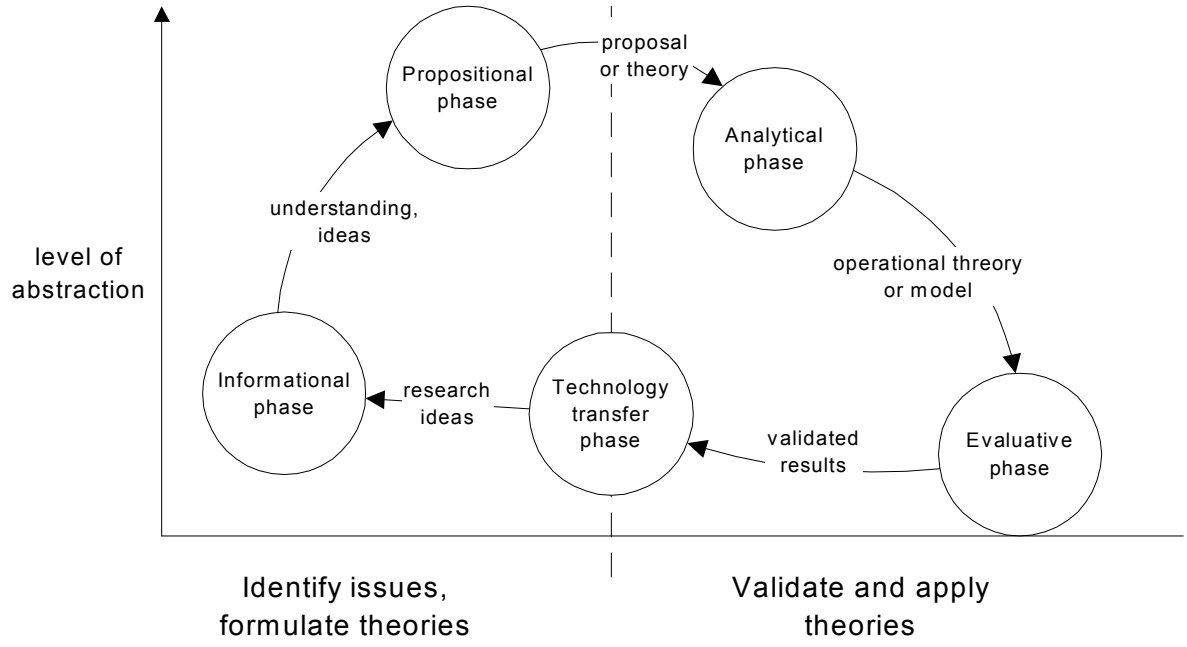


Figure 4: Research phases used in this research

We elaborated the research phases proposed by Glass (Glass 1995a) by presenting the research phases as iterative process, adding the phase of technology transfer, and modeling the process in two dimensions of level of abstraction and theory and issue formulation vs. validation. The model is shown in Figure 4 and Figure 5 shows the approximate timings of the various phases in this research. Note that the actual research work cannot be perfectly matched to such separate phases, as the practical research does not necessarily follow such sequential models (McGrath & Martin 1982). For instance, in our research, the latter phases also often included informational phase activities, such as broadening our literature search or using surveys and interviews to obtain more accurate information about the state-of-art and industry needs.

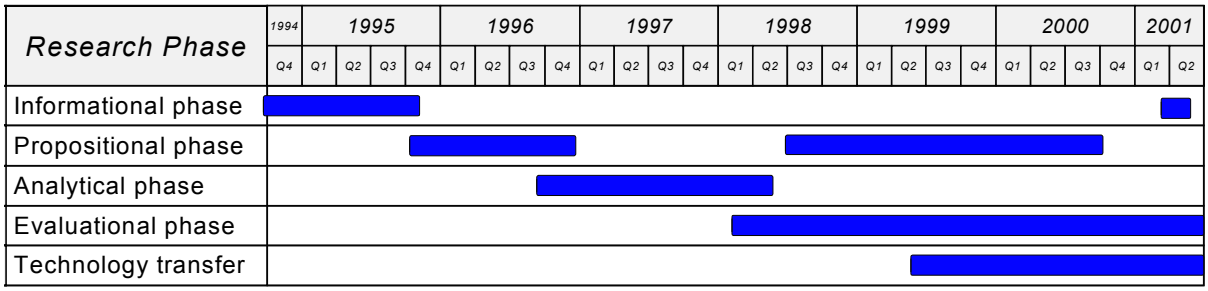


Figure 5: Research phases and timing in this research

The *informational phase* involves observing the current state-of-art and practice to identify problems and potential solutions. This can be based on literature surveys but empirical studies can also be used to systematically identify relevant research problems (Potts 1993). We conducted this phase by surveying the existing literature in software engineering, industrial project management, economics, psychology, government and safety literature, and social sciences. We used professional library services to identify potential publications and references and obtained the relevant articles and books to review them. The scope of literature search covered several hundred references. The method of literature survey was a four-step process. First, we studied potentially relevant references to understand their contributions and contexts. Second, we identified common and accepted practices or principles in various fields. Third, we synthesized the best practices in various fields and wrote a report (Kontio 1994a) and a unpublished working paper (Kontio 1995c) that documented the state-of-art and state-of practice, as well as identified some potential research themes. Fourth, we had several industrial software managers review the report and used the feedback and insight thus reviewed to identify our research objectives. An important part of these steps was also, naturally, my personal experience in software project management and my earlier involvement in the study of software life cycles, which helped the analysis of information.

In the *propositional phase,* the construct is formulated. The term construct in this context includes theories, models, and hypotheses that potentially contribute to the body of knowledge in the field. Based on the approach proposed by Nunamaker et al. (Nunamaker jr. et al. 1991), the construct formulation includes the formulation of conceptual framework and theories, and the development of implementation architecture and more detailed design. In our research, the propositional phase consisted of a more detailed formulation of the research objectives and research scope (Kontio 1996a), development of the main characteristics of the method to be developed, and the definition of the initial version of the method (Kontio

1996b). Again, the proposed construct was reviewed with several colleagues, mainly within the academic domain, but including also several industrial participants.

The timeline in Figure 5 also shows another instance of the propositional phase in this research. The propositional activity in 1998 and 1999 refers to the development of the eRiskit application, which was done in cooperation with a student group at Helsinki University of Technology. Given the database specification, Riskit method definition, and an initial requirement specification, the student group refined the requirements, created the technical design for the system, and implemented the prototype version of the eRiskit application. This prototype was developed further by a company specializing in risk management solutions.

The *analytical phase* involves the operationalization of the construct, as well as its analytical evaluation and improvement. As the analytical phase aims at making the proposed theories operational, this phase may include exploratory use or trials in laboratory or real-life context. The analytical phase often also includes further development or revision of the construct as the operationalization or trials provide further insight and feedback how to improve it. In our research, we carried out the analytical phase by applying the method in a small, industrial project at NASA. This first trial of the method forced us to develop an operational definition of the method and the feedback received served to improve and finalize the method further. This phase concluded in the release of the version 1.0 of the Riskit method (Kontio 1997).

The *evaluative phase* aims at testing and evaluating the construct. Usually this involves empirical studies, measurements, and analysis of evaluation results. This phase also includes the development and revision of the construct as evaluation feedback is taken into account. In our research, the evaluative phase consisted of six empirical studies. Evaluations can take place in industry (in vivo) or in a laboratory setting (in vitro) (Jeffery & Votta 1999). There is an inherent trade-off between in vivo and in vitro studies: in vivo studies have less control but they contain "world realism", bringing an element or credibility to them (Mason 1989). The in vitro studies are the opposite: high level of control but performed in an artificial environment. Each researcher will need to find the right balance between these aspects.

In this research, the evaluative phase also included an extensive survey of empirical research methods and literature in other fields. We surveyed the empirical research methods used in software engineering, MIS, and social sciences to identify suitable empirical research approaches and methods to be used. Most of our empirical studies can be characterized as case studies, but we have also conducted surveys and field experiments. In the following, we will discuss the main empirical research approaches used. Note that each empirical study report in chapter 5 will discuss the specific research approach and issues when the study and its results are presented.

The case study designs and approaches were primarily based on experiences in the social studies field (Haytin 1988; Judd et al. 1991; Patton 1990; Stake 1995; Taylor & Bogdan 1984; Yin 1994) and in the MIS and management science fields (Cheon et al. 1993; Galliers 1991; Eisenhardt 1989; Ghauri et al. 1995; Järvenpää et al. 1985; Nissen et al. 1991), as well as reviewing the empirical and case study reports and guidelines in software engineering and management (Basili et al. 1986; Wohlin et al. 1999; Basili 1996; Curtis 1980; Juristo & Moreno 2001; Pfleeger 1997; Swanson & Beath 1988). We used a postivist approach in our case study designs (Cavaye 1996), i.e., we wrote an empirical study plan for each case study, documenting the targeted research questions, arrangements for the study, instrumentation

used, timing of the study and instrumentation, and the questionnaires, data or artifacts to be captured or collected during or after the study.

Our research paradigm is primarily a qualitative one, although one of our studies is a quantitative experiment (study 6, reported in chapter 5.6). We relied on qualitative research approach because of four main reasons, based on criteria proposed by Creswell (Creswell 1994). First, the Riskit method is a comprehensive construct and it is difficult to evaluate it in a quantitative experiment. Second, our constructs were used in a limited number of cases, making quantitative evaluation of them less powerful. Third, our constructs are new and we felt that qualitative approach provides more insightful information about their characteristics. Fourth, we did not find empirical studies and relevant theoretical frameworks for software engineering risk management that quantitative approach could have been based on.

The research goals for each empirical study were defined by using the Goal/Question/Metric method (Basili & Weiss 1984; Basili & Rombach 1987; Rombach 1991; Basili 1992; Basili et al. 1994a; van Solingen & Berghout 1999; Basili 1992; Basili et al. 1994a; Rombach 1991). The GQM-method is an approach that help articulate and focus research questions and refining them into a set of operational questions. The GQM method contains two main principles, a template for identifying main attributes of analysis goals and hierarchical decomposition of metrics from these analysis goals. In this research, we made some adaptations to the GQM-method. First, we documented the rationale of research goals as an additional element in the GQM goal statement. Second, we used the GQM approach not only to identify and formulate questions but also to identify artifacts or other information items to be captured during the study. Third, in one of our studies we used a set of tracking tables to allocate questions into questionnaire sets and to maintain full traceability from metrics to goals. Finally, we introduced the concept of question triangulation into GQM in order to improve the reliability of data obtained. The question triangulation principle means that for each question, if possible, we intentionally defined several metrics that attempted to clarify or answer the same question.

The GQM goals are usually expressed in GQM statements, we have adapted the GQM statement and are using it as shown in Table 4. *Object of study* identifies the entity being studied, such as “inspection process” or “requirement specification”. These objects can be abstract entities or concrete physical attributes. The *purpose* attribute indicates the type of analysis to be performed. Basili lists several possible purposes, such as characterization, assessment, understanding, evaluation, prediction, control, motivation and improvement (Basili & Rombach 1988; Basili 1992; Basili et al. 1994a; Basili 1996). *Attributes* refer to an attribute of interest in the object of study. Depending on the situation, they can refer to a broad range of things, such as “cycle time”, “cost”, “effectiveness”, “validity”, or “user satisfaction”. *Point of view* defines whose perspective is used in the evaluation of the results, e.g., “project management” or “users”, and *context* is defined when it is necessary to highlight the environmental characteristics that may influence the generalizability of the results. The *rationale* has been included to summarize the reason why the goal is relevant.

Despite the positivist, goal-oriented design of our empirical studies, we also relied on an opportunistic approach during our studies. We relied on the so-called responsive and naturalistic research approach (Guba & Lincoln 1981) to allow the identification and capture of issues that rose during the studies, even though they were not necessarily included in the a priori empirical study objectives or design. To support this, we included additional instrumentation and study practices to capture such items, such as conducting summary briefings after each session and asking participants to use and fill in personal logbooks. This

naturalistic approach provided us with additional insights and information on how risk management was done in our studies.

Analyze	The object of study: what entity is being studied
in order to	The purpose of measurement: what is the learning objective. Often stated in one of the following generic study goals: ***Characterize***: to record or measure features or attributes in the entity being studied. • *Describe:* Collect relevant information to document information about an object or phenomenon. • *Monitor*: Use measurement data to track progress or status. • *Understand:* Recognize and propose patterns, hypotheses, theories, or models. ***Evaluate***: • *Assess*: Evaluate against a well-defined standard or baseline. • *Compare*: Evaluate two or more alternatives against each other. • *Validate*: Evaluate feasibility. • *Appraise*: Evaluate effectiveness or usefulness.
With respect to	The attributes of interest: what attributes of the entity are of interest.
From perspective of	The utilization perspective: who is interested in using the information.
in the context of	Context of measurement: what is the overall context of measurement.
because	The rationale for the study measurement.

Table 4: GQM template used in this work

We controlled the threats to the validity of our research based on the guidelines presented in various sources (Campbell et al. 1982; Wohlin et al. 1999; Straub 1989; Judd et al. 1991). The specific validity control actions taken are presented in chapter 5 with each of the empirical studies. In general, as our empirical approach is mainly based on case studies and the number of case studies is relatively low, the generalizeability of the results is understandably low. However, in our analysis of the validity threats, we did not observe or recognize any major threats to external validity and we therefore believe that the empirical study results are representative. On the other hand, the internal validities of the empirical studies were addressed and controlled within the constraints and resources available and, therefore, we believe that the internal validity threads do not invalidate the results presented in this work.

We used several instruments in the studies to capture study data, as shown in Table 5. As the table shows, most studies used several instruments to collect the data. This was done to improve the validity of data, i.e., through triangulation of data sources the probabilities associated to instrumentation errors and data source biases were reduced (Jick 1979; Judd et al. 1991).

Most studies involved the use of questionnaires and semi-structured interviews. The questions themselves were identified using the GQM-method and we relied on published guidelines (Babbie 1973; Schuman & Presser 1981; Hakel 1982; Sudman & Bradburn 1982; Converse & Presser 1996; Ghauri et al. 1995) in the phrasing and ordering of the questions to avoid biases and loading of questions. In conducting the interviews we started each session by stating the objectives of the interview, ensuring the confidentiality of the interview, and asking the interviewees to provide their views as openly and candidly as possible.

	Data collection instruments						
Study	Semi-structured interviews	Questionnaires	Postmortem analysis of artifacts and data	Logbooks	Introspection	Video/audio recordings	Affinity grouping
Study 1: Focus Groups	X					X	X
Study 2: Exploratory Case Study at NASA	X	X	X		X		
Study 3: Characterizing Case Study at Hughes			X				
Study 4: Nokia and DaimlerChrysler Study	X	X	X	X	X	X[1]	
Study 5: Method Introduction Study with IESE and Tenovis	X	X	X		X		
Study 6: Risk information visualization study with students		X	X			X	

[1] The video recording in study 4 was only done in one of the sessions and to control facilitator involvement; it was not used to analyze the process flow.

Table 5: Data collection instruments used in the empirical studies

The semi-structured interview sessions were conducted using the predefined questions as an outline, but additional comments or threads in discussions were also recorded. Interviewers used an approach which refrained from actively commenting and pursuing issues with interviewees (Fowler & Mangione 1990; Guba & Lincoln 1981; Sudman & Bradburn 1982), but they asked clarification or confirmation questions when needed. The recording of interview answers depended on the available resources. Usually, a scribe was used to record the answers, but we also used video recordings, email, and the interviewer as the note taker. In the latter case, special care was taken to pause the interview session so that answers were recorded on notes sufficiently well. In all situations, the transcripts of the sessions were written within a few days of the interview.

Studies two to six used the Riskit method and documented the results of the analysis in predefined templates, forms and databases. Templates for these were designed prior to each study and all items were archived. These artifacts allowed the study of how many risks were identified, how they evolved during the process, and what kind of risks and controlling actions were considered and implemented. These artifacts were analyzed in postmortem analysis sessions.

We also used logbooks in the study number 4 (Nokia and DaimlerChrysler study). Participating project managers carried a special logbook with them to record any observations or issues that occurred during the studies. The logbook instructions were given as presented in Table 6.

We also relied on introspection in the studies where empirical study organizers were facilitating the sessions. In such cases the empirical studies can also be considered action research (Guba & Lincoln 1981; Mills 1999; Stringer 1999) but note that the empirical studies in all cases included and explicit objectives and arrangements that were carried out.

Also, while the author of this work was a facilitator in some of the studies (studies 2, 3, and the Nokia study in empirical study 4), the other studies were facilitated by people who were not developers of the Riskit method.

This logbook is a personal log for notes, observations, and issues that are observed or identified during risk management session in the risk management study. All information entered into this log is kept confidential. Any relevant observation or issue should be entered into this booklet. For each entry the following information is recommended to be included: • Date and time of the observation • Context and setting (e.g., a specific meeting, a discussion, etc.) • Participants • Description It is recommended that this logbook is available at all discussions, meetings, and individual work where risk management related work takes place. At the end of each such activity, it is recommended that the logbook owner takes a moment to fill in observations and issues from the session.

Table 6: Instructions for the use of the logbook

We used video and audio recordings in the focus group study (study 1) to record and capture the points occurred during the discussion. Participants were informed on the video recording in advance and were able to raise any concerns about it individually to study organizers. The recording was not started until everybody present in the meeting had agreed to the recording.

The focus group arrangements are discussed in more detail in chapter 5.1. In short, the focus group method was selected in order to obtain a more in-depth view of the current practices and near-term development needs in the industry. We surveyed and used the documented best practices in planning and conducting the sessions (Edmunds 1991; Stewart & Shamdasani 1990; Templeton 1994; Feig 1989; Templeton 1994), and used the affinity grouping technique (Brassard & Ritter 1994) to capture industry needs and near-term development areas.

The *technology transfer phase*, strictly speaking, does not belong to the traditional view of the research cycle. However, we are including it in this presentation because evaluations in the software engineering field should be done in an industrial context, and effective feedback and further development of constructs also benefits from continuous feedback and improvement. This is also in line the philosophy of the Experience Factory concept, i.e., industrial use of constructs should include continuous measurement, analysis, and packaging of knowledge. Thus, the technology transfer phase includes the packaging of the construct into a deployable, industrially usable form, making it accessible to industry or users, and establishing mechanisms to collect and analyze empirical feedback from its use. Credible evidence of the benefits of the technology, as well as a thorough understanding of the user needs are, among other things, critical success factors in technology transfer (Debou et al. 1993; Eldred & McGrath 1997b; Eldred & McGrath 1997a). As researchers, it is in our interest to make the technology transfer as effective as possible as it will bring additional feedback on our constructs' benefits and disadvantages, allow the accumulation and sharing of experiences on a much broader scale, and improve the industrial practice of risk management.

Of the three main constructs developed in this research, only the Riskit method clearly reached the technology transfer phase. The improvement framework and the eRiskit

application were not developed to a stage of industrial, packaged technology transfer, even though they have been used in industrial context, based on customization. The packaging of Riskit consisted of the following components and activities:

- Industrial quality training packages were developed and made available to several industrial organizations and users. The number of people who have received basic training in the Riskit method is approximately 300, including industrial tutorials and tutorials in conferences (Kontio 1999).
- A web site containing several papers and publications has been established to provide general and free access to Riskit related information (R & D-Ware Oy 2001).
- A consulting service[1] to support the customization and deployment of the Riskit method has been established and it has provided consulting to four major customers in Europe.

It is important to point out the dependies to R & D-Ware Oy (http://ww.rdware.com/) in context of this research and technology transfer. The company was founded as the technology transfer vehicle for results of this research with a motivation to disseminate risk management knowledge to a broader audience. However, as I am also a majority owner of R & D-Ware Oy, there is a potential conflict of interests and bias present. Even though I have concisously attempted to prioritize my research interest higher in any potential conflict situations, I urge readers be aware of this potential conflict of interest and bias.

As a part of our technology trasfer actions, a patent application on the key concepts of the Riskit method was filed in May 1999. However, subsequent analysis of the novelty of the innovation revealed that our earlier publications (Kontio & Basili 1996; Kontio 1997) had disclosed parts of the innovation in the patent application and the remaining patentable subject matter would have been too limited to justify the cost of pursuing the patenting process to its completion. The analysis of the innovation indicated, though, that without our earlier publications the likelihood of obtaining a patent would have been quite high.

In summary, the large scope of the research reported here has presented challenges for the research methods in use, especially for the empirical research methods. As our empirical studies have been conducted over five years, on two continents, and in three countries, involving eight organizations, the practical constraints present in each case have limited the control we have been able to place on the studies. Nevertheless, overall we believe that the rigor of applying sound empirical research methods satisfies or exceeds the current scientific standard in software engineering.

1.7 Dissertation Structure

This first chapter presented the motivation, objectives, and scope of the research, as well as explained the research methods used. This chapter also included a summary of the research contributions we claim as result of this research. As the content of the main chapters are presented in the paragraphs below, the relationships and contributions are highlighted in Figure 6.

[1] R & D-Ware Oy, http://www.rdware.com

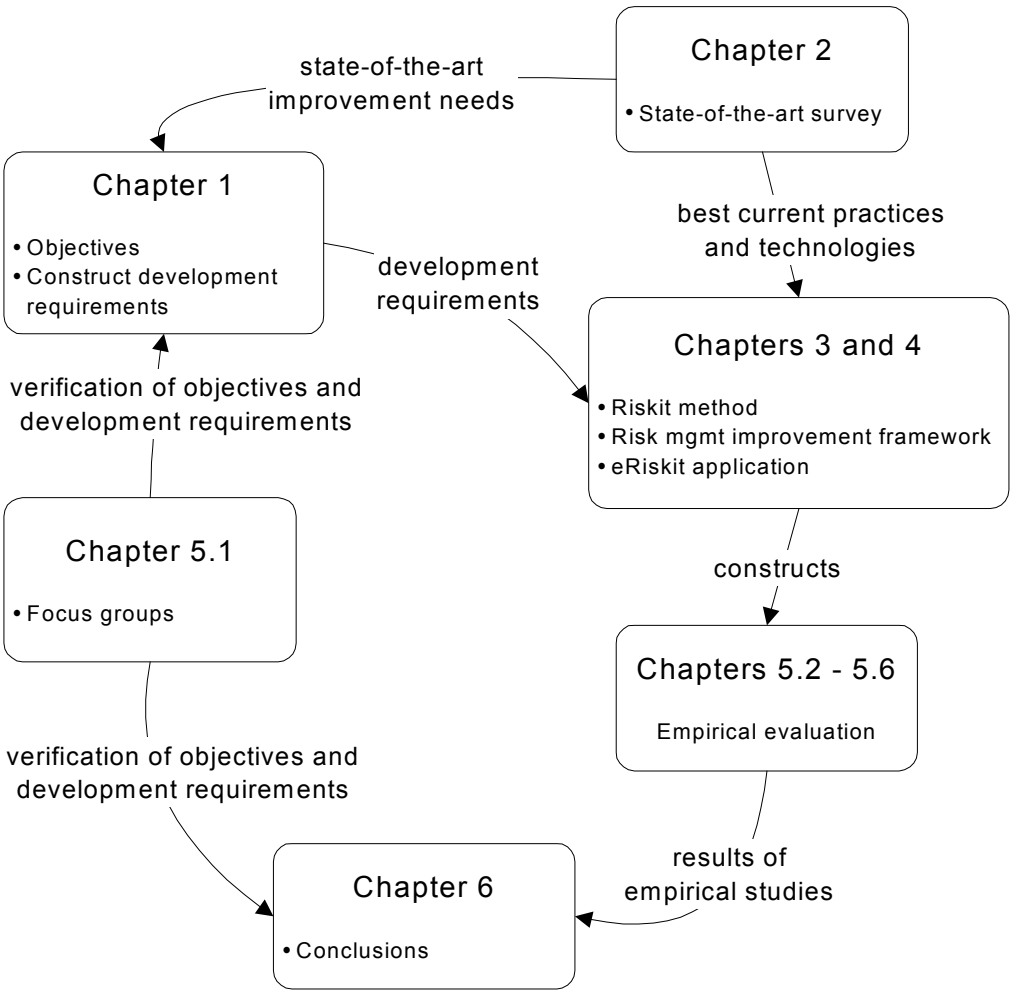


Figure 6: Relationships between main chapters in this thesis

Chapter 2 presents a summary of the literature survey conducted and characterizes the state-of-practice and state-of-art in risk management in software engineering. The main research directions and implications from psychology, management science, government and safety and project based business research are brought forth, and the chapter concludes in the definition and discussion of key terms used in this work.

Chapters 3 and 4 describe the main constructs that have been created in the propositional and analytical phases of this research. Chapter 3 presents the Riskit method using the Promises framework (Kontio 1995b; Kontio 1998) for presenting process models. Chapter 3 also contains the description of the key concepts, architecture and the main functionality of the eRiskit application. The chapter also includes a discussion of the experiences in populating the database with actual risk management data.

Chapter 4 presents the risk management improvement framework we have defined, using the Promises framework, as it was used in chapter 3.

Chapter 5 contains the description and results of the empirical studies conducted in this research. Each study, its objectives, arrangements, analysis methods, and its results are presented and discussed.

Chapter 6 contains conclusions of the research.

The chapter structure can also be visualized using the research life cycle introduced in chapter 1.6, as shown in Figure 7.

This work contains two main appendices. Appendix A contains a more detailed description of the risk management improvement framework. Appendix B describes the eRiskit application.

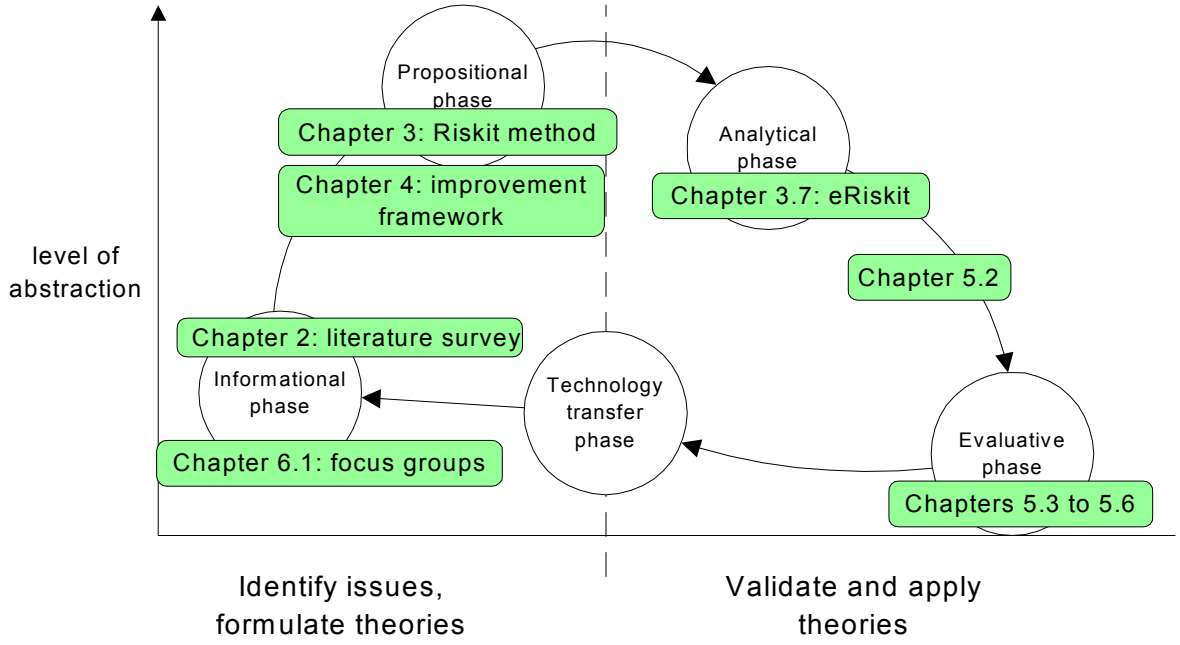


Figure 7: The research life cycle and the thesis main chapters

2. Review of Research on Risk Management

In this chapter, we provide an overview on risk management from historical perspective and from the various disciplines that address and study risk management and perception of risk. The chapter concludes in a summary of the review.

2.1 History and Origins of Risk

It is obvious that people have been dealing with risks on an intuitive level as long as the mankind has existed: hunting and farming have involved risks since the beginning of times. People have also been playing games of chance for possible several thousands of years (Covello & Mumpower 1998), even though the concepts of chance and probability have not been known, or at least not well understood (David 1978). Conscious and more analytical approach risk is still a recent phenomenon in our history.

The term risk originates through Italian from a Latin word *resceare*, meaning "to cut off. It is believed that the word was originally used by sailors to refer to danger (Anon.1913). However, the notion of risk and probabilities were not commonly used, until the fundamental concepts of risk were gradually discovered from 17th century onwards. Sambursky (Sambursky 1956) and Bernstein (Bernstein 1996) offer a theory to explain this: in ancient Greece and later in the Western culture the uncertain events were considered to be "acts of God" or results of faith. Estimating, calculating or managing such events was considered to be beyond what mortals should or could do. Instead, man was to take what faith brought along.

The modern risk management practice still relies on several fundamental concepts that were introduced since the 17th century. First, the theory of probability was created by Blaise Pascal and Pierre de Fermat, prompted to do so by a nobleman who wanted to find a solution to an old gambling puzzle (Anon. 1913; Struik 1987). However, several earlier authors, including Gerolamo Cardano and Galileo Galilei seem to have put forward some basic concepts on probability even earlier (David 1978). The theory of probability contains the axioms and theorems that establish the mathematical basis for calculating probabilities. The axioms of the probability theory are listed below (Fisz 1967):

Axiom I: *To every random event A there corresponds a certain number P(A), called the probability of A, which satisfies the inequality* $0 \leq P(A) \leq 1$

Axiom II: *The probability of sure event equals one, i.e.,* $P(E) = 1$

Axiom III: *The probability of the alternative of a finite or denumerable number of pairwise exclusive events equals the sum of the probabilities of these events.*

While the probability theory provided the mathematical basis for performing calculations on probabilities, until the 20th century it was only used in calculating games or in theoretical mathematical discussions. The practical application of the probability theory was rare.

Second, Jacob Bernoulli established the principle of statistical sampling in 1713 (Bernstein 1996; Struik 1987). This idea essentially allowed the use of samples to infer general

conclusions – or probabilities – to a larger population. Third, an English minister Thomas Bayes defined the theorem carrying his name (Bernstein 1996), supporting the calculation of probabilities and conditional probabilities in many practical applications.

Fourth, the phenomenon of normal distribution was discovered by de Moivre in 1725. The normal curve helps explain the variance and distribution of many natural phenomena. Fifth, Francis Galton's introduction of the regression to the mean helped understand how the data behaves in various distributions and how it can be interpreted (Bernstein 1996).

Based on these foundations, the theories related to risk and decision making evolved, especially during the 20th century, as discussed in the following chapter.

2.2 Decision-making under Uncertainty

The fields of economics and management science have studied human decision-making problems extensively over the past several decades. Several main theories have been proposed to explain human rationale and decision-making and we will briefly discuss each and conclude in recommending how these theories could help in software engineering risk management.

The central ideas of the utility theory were introduced by Daniel Bernoulli already in 1738 (Bernoulli 1738; Bernoulli 1954). However, the utility theory and Bernoulli's contributions to it remained largely unknown until Keynes published his work (Keynes 1921) on probability and economics (Bernstein 1996). Keynes also emphasized the need to rely on subjective probability, instead of only historical data on past event frequencies. Von Neuman and Morgenstern introduced how human preferences and subjective probabilities can be modeled and analyzed mathematically (Von Neumann & Morgenstern 1944). These contributions established the utility theory as the main theory to model human decision-making under uncertainty in economics and management science.

The expected utility theory, also known as decision theory, states that people make decisions based on the expected utility that different alternatives give to them. The utility is determined by each individual's utility function and it has been established that the utility function is, in most cases and for most individuals, non-linear (Rescher 1983; French 1986; Mas-Colell et al. 1995). In practice, this means that the expected values of events cannot be used as criteria for prioritizing alternatives. Instead, it is the alternatives' utility that determines these preferences.

The expected utility theory is based on a set of axioms, or assumptions, that describe how people should behave in order to use the expected utility theory (Hogarth 1987). First, people should have consistent beliefs, i.e., people's preference judgments should conform to the basic axioms of the probability theory. Second, people should have consistent preferences, i.e., given a set of outcomes, people should be able to consistently indicate their preferences over them. The axiom of consistent preferences has three important implications: (i) it implies *transitivity* of relationships between outcomes, (ii) if *dominant* outcomes can be established, outcomes that are inferior to dominant ones can be ruled out, and (iii) the preferences between outcomes should be *invariant* w.r.t. how the outcomes are presented. The third axiom of the expected utility theory is the principle of maximizing the expected utility of various alternatives, i.e., people should seek to choose outcomes whose expected utility gain is the highest. Finally, the fourth axiom of the utility theory is called the "sure-thing principle", which states that given outcomes and their relative preference, uncertainty not affecting these outcomes should not influence the stated preferences.

The expected utility theory has been widely used in economics and management science since the basic axioms are applicable in most situations and the theory allows mathematical calculations and models to be built on these axioms. The theory can also act as a working normative model in decision making situations, supporting decision makers in seeking their actual preferences (Mas-Colell et al. 1995). However, there are several problems and limitations that are associated with the expected utility theory (Hogarth 1987; Mas-Colell et al. 1995; Rowe 1977). First, there are several known paradoxes that the expected utility theory cannot explain, including the Bernoulli's original St. Petersburg paradox (Bernstein 1996), the Allais' paradox (Allais 1953) and the Machina's paradox (Machina 1987), all of which have shown that a large number of people who may not behave according to expected utility theory. It has also been argued that people have difficulties in keeping the preference judgments and probability judgments independent, finding out the exact form of a utility function is difficult or impossible in practice, and compound utility functions may cause difficulties in assessing preferences (Rowe 1977). However, these inconsistencies are not serious as, among other arguments, they involve extreme probabilities or gains and the use of the theory in a normative manner can change people's preferences to be in line with the expected utility theory (Mas-Colell et al. 1995).

The utility theory can be considered as a widely accepted approach to prioritize uncertain outcomes. We, therefore, recommend that it be used in software engineering risk management as the underlying theory to rank risks. It is interesting to note that while the software engineering risk management field has included many references to utility theory (Boehm 1981; Charette 1989; Hall 1998), we have not found any method that would have actually used and applied it in practice. In fact, the mainstream and textbook approach in the field recommends the use of expected loss, not expected utility loss, as the method for prioritizing risk (Boehm 1989; Charette 1989; Dorofee et al. 1996; Hall 1998; Pressman 2000).

Another major theory that influences risk management practice is the *theory of bounded rationality* by Herbert Simon (Simon 1979). Simon observed that the people and organizations have limited knowledge of outcomes and their probabilities and lack the required mental computational power to actually make decision that would maximize their expected utility. Nevertheless, they still are able to function and make decisions. Simon's theory of bounded rationality explains this by arguing that since the maximization of the expected utility is an impossible task, people choose to seek satisfactory alternatives by setting aspiration levels and using them to rule out alternatives in the search space (Hogarth 1987). The theory of bounded rationality has also been proposed and applied to information technology domain in the literature: Lyytinen et al. (Lyytinen et al. 1993) have developed a conceptual framework that uses the theory of bounded rationality to integrate management environment and project environment through risk management activities (Lyytinen et al. 1996; Lyytinen et al. 1998). Their work demonstrated how the theory of bounded rationality can be applied in project risk management and how risk management can be seen as a way of heuristically searching the solution space more effectively, i.e., risk management can be seen as a way to help project managers identify working solutions better.

The theory of bounded rationality can be used to explain and model project managers' behavior in risk management. While they seek to maximize their utility functions, the practical limitations described by the theory of bounded rationality make it impossible to find optimal solutions. Instead, satisficing solutions are sought and evaluated. The satisficing solutions can be compared against each other using the utility theory.

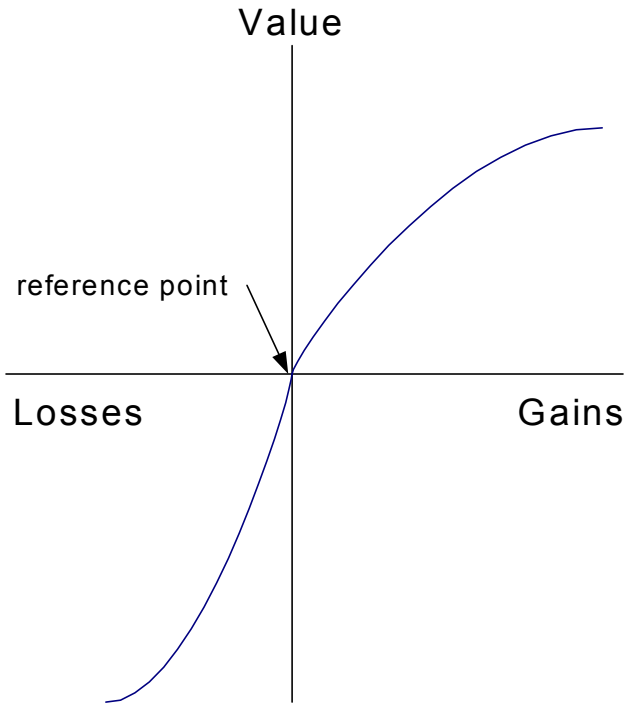


Figure 8: A typical prospect theory value function

Perhaps the most influential impact from the field of psychology to risk management is the prospect theory by Kahneman and Tversky (Kahneman & Tversky 1973; Kahneman et al. 1982). The prospect theory explains how people's perceptions influence the preferences between choices. Essentially, Kahneman and Tversky discovered several biases that people have in the way they evaluate decision alternatives. Understanding these biases is essential in dealing with subjective probabilities and assessing utility losses of alternatives, both of which are vital elements in risk management.

The prospect theory assumes that people have a reference point, a status quo point that is the basis of their value judgment. The shape of the value function, according to prospect theory, is like the one presented in Figure 8: losses cause steeper value losses than corresponding gains increase value. Therefore, the correct positioning of the reference point is important for the correct phrasing of the choices (Hogarth 1987). Another aspect of the prospect theory is that people seem to be more sensitive to differences near the reference point than to those farther away from the reference point.

Kahneman and Tversky also identified several biases that have been commonly seen in practice (Kahneman et al. 1982). They have categorized these biases into three main groups. The first category, biases linked to *representativeness*, refers to bias of associating similarity with probability, leading to the following five common biases.

People are *insensitive to prior information about probabilities* and, instead, associate much value to representativeness, i.e., judgments are based on how representative a given choice is to a stereotypical situation. If the representativeness is high, high probabilities are associated with the choice, regardless of the information available on the actual probabilities. For example, if people are aware of a project disaster that happened in a project using a specific technology, say Java, they may associate any new Java-based project as a risky one, regardless of accessible information that might indicate that Java was not the cause of the problem in the failed project.

People also seem to be *insensitive to sample size* and its implications to probabilities or value distributions. In particular, people generally fail to note that small samples are more likely to deviate from the mean than large ones. In practice, this may bias people's interpretations about historical data that is used to obtain reference points for probability estimates.

People have *misconceptions about probabilities*, assuming that even small samples should behave like the whole process with a large number of runs. A typical example of this is the gambler's fallacy: a long run of unfavorable outcomes should make the favorable outcome more likely. In addition, it has been observed that people can have overconfident attitudes and are not good at estimating probabilities near zero or near one (Covello 1984). In software projects it is likely that a long stretch of failed projects causes people to seek the root cause of the problem and if one is found, it may be corrected and more accurate probability estimates may be obtained. However, the probability estimates could be biased when participants perceive that the problems or failures were caused by outside factors or "bad luck", gamblers fallacy may influence the estimates.

People are *insensitive to predictability of information* provided. Again, people mainly seem to make estimates based on representativeness instead of considering how predictive or reliable the information available is. For instance, past project reports may contain positive but irrelevant information and people are inclined to interpret it positively, even though it actually may not contribute to such conclusions.

People seem to have *illusions of validity*, i.e., they rely on representativeness instead of critically evaluating whether the information available is reliable or up-to-date. For instance, people might use information provided by project manager or sponsor in evaluating project success probabilities, not realizing that such individuals may, perhaps unintentionally, present the information in too favorable light.

People have *misconceptions about regression*, i.e., they fail to account for the tendency of outlier values to regress to the mean. This bias may, for instance, cause people to interpret improvements in historical data as signs of systematic improvement, instead of realizing that improvements may have happened due to regression to the mean.

The second bias category refers to *availability of memory recall*. People tend to rely on information and experience that is most easily available when an assessment or decision needs to be done. This leads to the following four specific biases.

People have *biases due to retrievability of instances*, i.e., associate high weight to cases or situations that are easy to remember, erroneously relying on the recalled cases to represent the unbiased data or situation. For instance, highly publized or recent projects may be used as reference points, even when they might not be as relevant as projects that are more difficult to recall.

People seem to have *biases due to the search strategy* they use for recall. For example, when asked whether there are more words that start with the letter "r" of those where "r" is the third letter, people find it easier to recall words that start with "r" and therefore associate higher frequency to it. In software engineering context, this may cause people to look for examples that have some outstanding characteristics that direct the memory recall. For instance, use of a given technology or a high profile vendor might associate these examples too strongly into the analysis.

People also have a *bias related to imaginability*, i.e., if outcomes or situations are difficult to imagine or construct, people tend to emphasize the easy-to-imagine cases and fail to consider the hard-to-imagine cases in similar manner. This bias might influence the estimation of projects that involve new domains or technology, i.e., people find it difficult to think about risks and scenarios that could go wrong.

The phenomenon of *illusory correlation* occurs when people assume or discover correlation between observations or data sources if one of the sources can be considered reliable, even when the data does not justify such correlation to exist (Chapman & Chapman 1969; Tversky & Kahneman 1974). For instance, people may assume wrong or simplistic theories or causalities about the reasons for past failures.

The third bias category is called *adjustment and anchoring*, which means that people start from some reference point and adjust their estimate gradually. Typically, however, the adjustments are insufficient. The initial reference point influences the final estimate and it can be given or implied by the phrasing of the question, by incorrect calculation, or by false recall. The three adjustment and anchoring related biases are described below.

A typical bias is created by *insufficient adjustment*: if the initial anchoring point is too far off the mark, people often fail to make sufficient adjustments. For instance, people may take the past risk frequency data as the initial reference point and fail to adjust it sufficiently with the situation specific information.

Biases in the *evaluation of conjunctive and disjunctive events* means that people tend to overestimate the probability of conjunctive events and underestimate the probability of disjunctive events (Cohen et al. 1972; Tversky & Kahneman 1974). For instance, a project's success may require that a series of uncertain events should happen (or risks not happen). People may falsely overestimate the probability of success for these conjunct events, leading to false optimism.

When making assessments about *subjective probability distribution* of events, the anchoring effect seems to reduce the variance of such distribution, i.e., people estimate the probability distributions to be too narrow. In other words, people tend to fail to account to more extreme cases in risks.

The biases and heuristics identified by Tversky and Kahneman have been observed in several experiments (Kahneman et al. 1982) and they seem to influence judgments and decisions regarding risk management.

The practical implication of the prospect theory to software engineering risk management is that risk management methods should account for these biases and contain procedures that avoid or compensate them. This can be done by providing training to practitioners so that they are aware of these biases, developing methods that structure risk evaluation situations in a way that avoids or minimizes bias, or using checks in the evaluation process to reduce potential bias.

While the prospect theory assumes that people know the probabilities that are involved in making judgments – an assumption that is seldom true – the *ambiguity theory* postulates that probabilities are actually estimated by a process that is similar to the anchoring and adjustment approach, described earlier (Einhorn & Hogarth 1985). The estimate of probability depends on the amount of ambiguity present in a situation, e.g., due to lack of data or experience, and on a person's attitude toward ambiguity, e.g., whether the person is inclined to optimism or pessimism.

The *theory of cognitive dissonance* (Festinger 1957) is also a relevant paradigm for goal-oriented risk analysis situations (Harmon-Jones & Mills 1999). The theory of cognitive dissonance argues that when people are faced with information that is in conflict with their beliefs or knowledge, they experience unpleasant state of tension, called cognitive dissonance. People try to reduce the cognitive dissonance and usually use one of two approaches for it. First, they may increase the number of consistent cognitions, i.e., they seek to enforce their current view through selective data, rationalizing, selective recall of past experience. Alternatively, they may change their current attitudes and accept the new information into their belief set.

From the perspective of software risk management, the cognitive dissonance may result in the rejection of information, i.e., risks, that are in conflict with the agreed and committed project objectives. A successful project team needs to develop a high commitment to a project and its objectives in order to succeed (DeMarco & Lister 1987; Pressman 2000; Sommerville 1995), any information threatening the goals is likely to create high cognitive dissonance and thus projects may have a tendency to reject information about risks. To counter this, project teams should be made aware of the cognitive dissonance phenomenon and provided with practical tools to control risks.

Shefrin and Thaler have proposed the concept of *mental compartments* (Shefrin & Thaler 1988). According to their theory, people do not try to maximize the expected utility, instead, people view decision within a smaller context and allocate different decision criteria in each mental compartment. There are several economic phenomena that could be explained by mental compartment theory (Shiller 1999) and it is a plausible theory to characterize human behavior. The practical significance of mental compartments in software risk management is to try to make the potential compartments explicit so that they can be understood and dealt with explicitly amongst the stakeholders.

The *regret theory* has been proposed to better explain some of the paradoxes associated with the utility theory. According to regret theory, people do not try to maximize their expected utility; they try to minimize the regret, i.e., prospect of loss may create strong avoidance behavior that influences the decision. It has been argued that the regret theory is potentially more robust than the expected utility theory and therefore it describes human behavior better (Sudgen 1986).

B.F. Skinner observed the phenomenon of "*magical thinking*" in experiments with pigeons: pigeons were automatically fed every 15 seconds, yet the birds developed manners and procedures as if they could influence the supply of food (Skinner 1948). Quite interestingly, forms of magical thinking have been observed in people and organizations as well, e.g., people may falsely believe that decisions or actions they have taken actually influence the outcomes in the past, leading them to repeat and rely on them in the future as well (Langer 1975; Quattrone & Tversky 1984; Shafir & Tversky 1992). From the risk management perspective, the magical thinking may result in ineffective risk controlling actions. This should be compensated by analyzing and planning risk controlling actions in an unbiased fashion.

Tversky and Shafir have also observed a behavior called *disjunction effect*: people tend to push off their decisions until further information is available, even when the pending information does not affect the decision (Tversky & Shafir 1992). The disjunction effect is in contradiction with the "sure-thing" principle discussed earlier and thus undermines one of the axioms of the expected utility theory. The disjunction effect may cause problems in making decisions in the risk management process – it creates a bias to delay implementing risk

controlling action and this may reduce the scope or effectiveness of such actions. To control this effect, a systematic and unbiased approach should be used in analyzing and deciding risk controlling actions.

The prospect theory can be considered a descriptive theory about human behavior, i.e., it explains why people have certain preferences in decision-making situations. However, the utility theory can be considered more of a normative theory, i.e., one that explains how people *should* behave. The intent of software risk management is to guide decision makers to make the decisions that are right for the organization and key stakeholders. Therefore, we recommend that the utility theory be used as the normative model in risk related decision making and the prospect theory is used as a descriptive framework to control and understand the biases present in human decision-making. The other theories presented and discussed in this chapter can be used in developing risk management methods that take into account the biases associated to these theories.

2.3 Risk Management in Software Engineering

Within the realm of systems and software, risk management was initially addressed by the management information systems (MIS) community. Nolan (Nolan 1973; Nolan 1979) and McFarlan (McFarlan 1974) proposed models for managing the information technology and project portfolio in the 1970's, Alter and Ginzberg proposed that analysis of risk factors can help developers succeed (Alter & Ginzberg 1978), Davis proposed a model for selecting a development approach based on the uncertainties in requirements (Davis 1982) and Saarinen et al. have developed more elaborate models to support project portfolio management (Saarinen 1993). Despite these efforts, risks in software development were not addressed in any detail until late 1980's, when Boehm (Boehm 1988; Boehm 1989) proposed and synthesized more detailed approaches for risk management. His work was complemented by Charette (Charette 1989) and software engineering risk management is now an established area within software engineering community. The Software Engineering Institute and annual software risk management conferences acted as the main forum to share experiences and results between practitioners and researchers (SEI 1993; SEI 1994; SEI 1995; SEI 1997).

More recent advances in software risk management have produced well-documented approaches for risk management (Karolak 1996; Michaels 1996; Pandelios et al. 1996; Hefner 1994), several categories of risks have been proposed (Chittister & Haimes 1993; Carr et al. 1993; Laitinen et al. 1993; Boehm 1989), quantitative approaches for risk management have been proposed and used (Bowers 1994; Fairley 1994; Berny & Townsend 1993), and there are several software tools available for risk management. Furthermore, most commonly used software engineering standards require some form of risk management to take place, although they do not provide detailed requirements on risk management, as discussed in chapter 2.5. (DoD 1988; ESA 1991; IEEE 1992; IEEE 1987; ISO 1994; ISO 1991b; Singh 1991; IEEE 1992; Paulk et al. 1993a; Koch 1993)

Despite the recent advances in risk management and the obvious industry interest in it, it seems that only a minority of organizations are using specific risk management approaches actively. The U.S. defense sector has been addressing risk management systematically since early 1980s (Anon.1983; Anon.1989; Anon.1988; Edgar 1989). The government had set up an internal training program that included education on risk estimation and analysis techniques. The U.S. Army, in fact, had set up a systematic risk assessment requirement already in 1981, program managers were required to estimate the cost of risk using a defined method called TRACE (Edgar 1989). TRACE required systematic steps to be followed in

assessing project risks and proposed a set of techniques for these steps. As a result, projects were allocated a specific fund that could be used if risks realized. It seems that these funds were not used for risk prevention, however. The notion of risk budget may be a useful idea for commercial projects as well, but it should be primarily used for risk prevention. However, their primary focus has been in the planning and budgeting stage of defense programs, not in day-to-day management of software projects and, consequently, some programs do not manage risks explicitly at all (Kirkpatrick et al. 1994).

Industrial reports on software risk management are relatively rare with some notable exceptions (Boehm 1991; Chittister et al. 1992; Eslinger et al. 1993; Fairley 1994; Gemmer & Koch 1994; Hefner 1994; Laitinen et al. 1993; Meyers & Trbovich 1993; Morin 1993; Williamson 1994; Conrow & Shishido 1997). None of these reports has been able to provide concrete, quantifiable data about the benefits of risk management methods, although they do provide indications that some benefits exist.

Barry Boehm's work has been the main foundation for most of the risk management work in software engineering (Boehm 1981; Boehm 1987; Boehm 1988; Boehm 1989; Boehm 1991; Boehm 1992). His main contributions have been in establishing the risk management as an important field of study in software management, introduction of some key measures for risk, and synthesizing a set of techniques into a single framework for risk management. Boehm's spiral life cycle model was the first life cycle model to incorporate risk management explicitly in it (Boehm 1988) and many recent papers on software life cycles have incorporated similar notions of risk in them. In his risk management tutorial (Boehm 1989) Boehm presented more detailed account of his risk management approach.

Boehm's risk management approach relies on the quantification of risk. He used the term *risk exposure* as a measure of risk:

$$\textit{Risk Exposure} = \textit{Probability(Outcome)} * \textit{Loss(Outcome)} \qquad (1)$$

Boehm also used the term *risk reduction leverage* as a measure of effectiveness of risk reducing action:

$$\text{Risk Reduction Leverage} = \frac{\text{Risk Exposure}_{before} - \text{Risk Exposure}_{after}}{\text{Risk Reduction Cost}} \qquad (2)$$

The risk exposure is essentially the *expected value* of the risk (event). Expected value is a well-established way of calculating uncertain events. The use of expected value as a measure of risk has several important benefits (Rescher 1983): it has a solid theoretical foundation, it can be used with different measurement units and scales, and it allows aggregation and disaggregation of results. However, the use of expected value in risk prioritization can be considered biased as it does not take into account the aspects highlighted by the utility theory, as discussed in chapter 2.2.

While the theoretical foundations of the expected value calculations are sound, the practical value of expected value concept is limited by the difficulties in estimating the probability of different risks. In principle, there are four possible sources for obtaining an estimate for risk probability: use of historical data (statistics), theoretical analysis, and subjective estimates. In software engineering historical data is seldom available and when it is, it rarely can provide reliable basis for estimating probabilities: the situation and context may have changed and, by definition, many risks are low probability events that may have few occurrences in the past, i.e., there may be too few data points to determine probabilities. Theoretical analysis means the use of some theories or models to determine the probability,

e.g., a symmetrically shaped coin can be expected to have 50% probability of landing on either side. In software engineering, unfortunately, there are few general theories that lend themselves to be used in such analyses but, for instance, cost models can be used to support such analysis. The use of subjective estimations of probability is a common method in software engineering, sometimes supported partially by historical data or by theoretical analysis. Finally, computer models and simulation tools can be used to study how system may behave (Abdel-Hamid & Madnick 1991; Berny & Townsend 1993; Bröckers 1995) and use this information to estimate probabilities of some events. The use of such models requires that accurate models can be developed and that accurate parameters or performance data can be used as a basis for the system.

Risk Management	**Risk Assessment**	Risk identification	Checklists Decision-driver analysis Assumption analysis Decomposition Brainstorming
		Risk analysis	Decision analysis Network analysis Cost models Quality factor analysis Performance analysis
		Risk prioritization	Risk exposure Risk reduction leverage Compound reduction
	Risk Control	Risk management planning	Buying information Risk avoidance Risk transfer Risk reduction Risk element planning Risk plan integration
		Risk resolution	Prototypes Simulations Benchmarks Analyses Staffing
		Risk monitoring	Milestone tracking Top 10 tracking Risk reassessment Corrective action

Table 7: Boehm's risk management model

In practice, probability estimates have high margins of error and this easily leads to large errors in expected value calculations. Consequently, the expected value of risk is not a very reliable measure of risk in most instances. Despite these limitations, expected value concept has been widely used as a measure of risk.

Another major contribution of Boehm was the consolidation of some main techniques for risk management into a single framework. He divided risk management into two main aspects, *risk assessment,* and *risk control.* These were further divided into steps that were supported by a set of techniques. Table 7 presents Boehm's risk management model. The right-most column presents the techniques that can be used to support each step. More information on these techniques is available in the references (Boehm 1989; Boehm 1991).

SEI's Software Risk Evaluation method has been developed to support systematic risk evaluation (Sisti & Joseph 1994). The method has been also extended to support teams in risk management (Pandelios 1996) and SEI has started collecting their assessment results into a database for further analysis of identified risks (Monarch et al. 1996). SEI's method is structured around a set of continuous tasks that guide the risk management process:

- *Identify*: The method relies on SEI's risk taxonomy to identify potential risk areas (see Table 8).
- *Analyze*: Transforming data from the identified risks into decision-making information. The SRE approach recommends using two alternative, table-based approaches for ranking risks.
- *Plan*: Plan risk mitigation, i.e., define and rank actions to mitigate risks, prioritize actions, and integrating them into an executable risk management plan.
- *Track*: Monitoring the status of risks and their mitigation actions along with the use of metrics and triggering events.
- *Control*: Correcting the deviations from planned risk mitigation actions by using existing program or project management control functions.

A. Product Engineering	B. Development Environment	C. Program Constraints
1. Requirements	**1. Development Process**	**1. Resources**
a. Stability	a. Formality	a. Schedule
b. Completeness	b. Suitability	b. Staff
c. Clarity	c. Process Control	c. Budget
d. Validity	d. Familiarity	d. Facilities
e. Feasibility	e. Product Control	**2. Contract**
f. Precedent	**2. Development System**	a. Type of contract
g. Scale	a. Capacity	b. Restrictions
2. Design	b. Suitability	c. Dependencies
a. Functionality	c. Usability	**3. Program Interfaces**
b. Difficulty	d. Familiarity	a. Customer
c. Interfaces	e. Reliability	b. Associate Contractors
d. Performance	f. System Support	c. Subcontractors
e. Testability	g. Deliverability	d. Prime Contractor
f. Hardware Constraints	**3. Management Process**	e. Corporate Management
g. Non-Developmental Software	a. Planning	f. Vendors
3. Code and Unit Test	b. Project Organization	g. Politics
a. Feasibility	c. Management Experience	
b. Testing	d. Program Interfaces	
c. Coding/Implementation	**4. Management Methods**	
4. Integration and Test	a. Monitoring	
a. Environment	b. Personnel Management	
b. Product Integration	c. Quality Assurance	
c. System Integration	d. Configuration Management	
5. Engineering Specialties	**5. Work Environment**	
a. Maintainability	a. Quality Attitude	
b. Reliability	b. Cooperation	
c. Safety	c. Communication	
d. Security	d. Morale	
e. Human Factors		
f. Specifications		

Table 8: SEI's risk taxonomy

- *Communicate*: Exchanging risk management information among the functions and at all levels of the organization.

The above steps are visually represented in Figure 9.

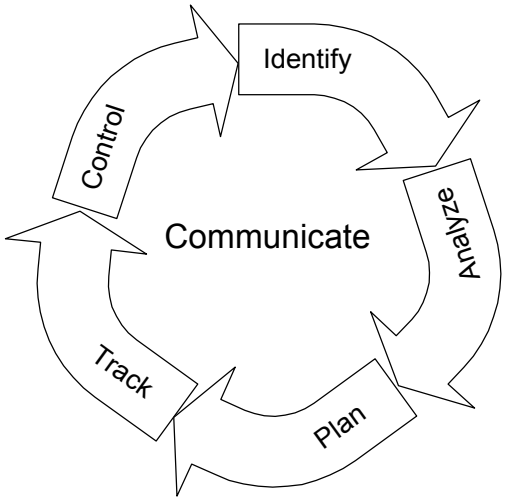


Figure 9: SEI's risk management cycle

A central element in SEI's approach is the risk taxonomy and associated questionnaire. The taxonomy is presented in Table 8. The taxonomy is covered with a questionnaire that, through its 194 questions, covers all areas listed in Table 8. Example questions are presented below (Sisti & Joseph 1994):

[8] Are there any requirements that may not specify what the customer really wants?
(Yes) (8.a) How are you resolving this?
...
[17] Does any of the design depend on unrealistic or optimistic assumptions?
...
[23] Has a performance analysis been done?
(Yes) (23.a) What is your level of confidence in the performance analysis?
(Yes) (23.b) Do you have a model to track performance through design and implementation?

The SEI method documents risks using risk statements (Gluch 1994; Dorofee et al. 1996). Risk statements document risks in condition – consequence pairs. The condition attribute contains a sentence describing the situation and the consequence attribute describes the outcome of the current condition if a risk occurs. The risk statements can be used visually, as shown in Figure 10, or on textual basis, as presented below:

- (Given the condition that)
  we must use Java and we have little experience in it;
- (then there is a concern that)
  the implementation phase may last longer than planned.

The SEI's method is well defined and requires broad participation from the organization that is using it. A typical risk management cycle lasts two to four months (Sisti & Joseph 1994). Given its higher costs it seems that the SEI method is suited for assessing program level in the beginning of a large program. It provides training and understanding on risk management issues while identifying risk areas. The SEI Continuous Risk Management Guidebook (Dorofee et al. 1996) is one of the most comprehensive collection of practical techniques that can be used in various steps during risk analysis. However, while being a practical and easy to use, the guidebook contains hardly any theoretical introduction to risk management and it does not discuss many key limitations and biases associated with the techniques it presents.

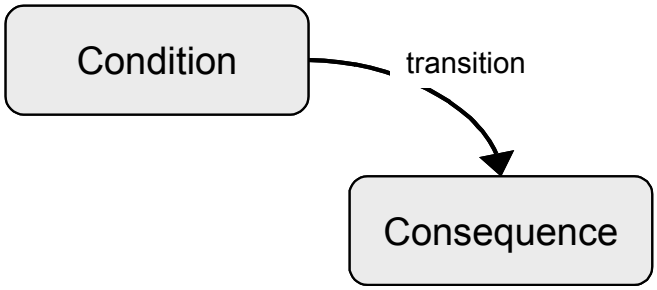


Figure 10: Visual format for the SEI risk statements

Hall has recently proposed a five-level capability maturity model for risk management, the RM-CMM (Hall 1995). The model contains a set of factors that are used to assess the maturity of a risk management system. The factors used in Hall's model are listed later in this thesis in Table 38 (page 100). Although Hall presented some survey data to support the model, it has not been formally validated. Consequently, it perhaps should not be seen as a normative maturity model but a framework for identifying risk management issues.

Hall has also developed a comprehensive risk management approach that includes risk management process definition, description of the risk management infrastructure, and guidelines for implementing risk management in practice (Hall 1998). Hall's approach also incorporates goal setting, project planning, execution, measurement, improvement, and discovery of new information into a conceptual framework for project execution and improvement. This "six-discipline model" can be considered an improvement paradigm that consolidates some aspects of risk management into it.

An ESPRIT project MERMAID and its Finnish participant, VTT, have developed a risk management method called RiskMethod (Känsälä 1993; Känsälä 1997). The method is also supported by a MS-Windows-based tool called RiskTool[2] (Kalliomäki & Känsälä 1993). Both RiskMethod and RiskTool focus on cost and schedule risks and the tool was integrated with three different cost estimation models. Madachy (Madachy 1997) has also presented an approach that uses heuristic rules in an expert system to feed the parameters of COCOMO (Boehm et al. 1995). Kitchenham and Linkman (Kitchenham & Linkman 1997) have discussed the issues and problems associated in accounting for the estimation errors in the use of cost models in risk management: they provide guidelines for dealing with errors due to assumptions used or the applicability o the model to the domain in question.

[2] Unfortunately, the development of the RiskMethod and RiskTool was stopped in the early 1990's.

The third major contribution of the MERMAID project and VTT is the list of 157 risk drivers from different, public information sources. This list also includes definitions for each risk driver. The definition includes name, definition, type of measurement scale, and possible values in the scale. The risk driver list has been divided into nine separate areas: customer, contract, supplier, project, project group, key personnel, initial situation, phase products, and software quality. The risk driver list is an important input for an organization that starts to develop its own risk driver list.

Karolak has developed a risk management approach that is based on identifying a set of high-level risk categories, called risk elements by Karolak, associating risk factors to them, and, again, associating specific risk metrics, or questions, to these factors (Karolak 1996). Questions are answered by project representatives, answers are converted to numerical values, and the network of answers and their weights are used to calculate risk factor values, using probability tree calculations. Karolak's model gives quantified estimates of project's risks along the risk factor categories. In addition, the questions in the model can be used as checklists to identify specific risks. We are not aware of empirical validation reports on Karalok's model, though, and thus the evaluation of its validity is an open issue. There are several other proposed models that use the same principle of trying to use situational factors (Foo & Muruganantham 2000; Roy & Woodings 2000; Deutsch 1991; Madachy 1997; Groth 1992), software component characteristics (Briand et al. 1993b; Briand et al. 1993a; Madachy 1997; Madachy 1997), or software architecture characteristics (Weyuker 1999) to predict project risks and using to predict projects risks.

The influence of the motivation, attitudes, and organizational context to risk management has been recognized by several authors. Gemmer studied practices within Rockwell and observed that current existing cultural rules acted as disincentives for proactive risk management (Gemmer 1997) and called for effective communication to change the climate to be more accepting to risk management. Hall has also recognized the importance of providing motivation and skills for risk management, as well as involving people form all levels in risk management (Hall 1998).

The traditional engineering fields have also addressed risk management over the past decades (Michaels 1996; Petroski 1985; Ricci et al. 1981; Waller & Covello 1984; Wang & Roush 2000). In particular, the Failure Modes and Effects Analysis (FMEA) approach is commonly used during the design of hardware products to identify potential design flaws by reviewing specifications and designs, and to estimate their impacts (Stamatis 1995). FMEA has been extended to include the evaluation of criticality and risk (FMECA), i.e., the method results in identification of most severe failure modes. Even though software is often more complex and its components not as clearly decomposable, the FMEA can be used to analyze risks associate to software operation. However, the method is not equally well suited to evaluating software development risks as software development project and organization cannot necessarily be decomposed in a way that it would help analyze main risks.

In summary, the software risk management has been an active research topic during the past 15 years, resulting in a comprehensive portfolio of approaches. However, as we will discuss in the next chapter, many of the proposed approaches have problems associated with them and these limitations are rarely discussed.

2.4 Limitations of Current Approaches

As the previous discussion shows, there is no shortage of proposed approaches for risk management. It is perhaps surprising that many existing approaches have various limitations that are often not acknowledged or addressed by authors or by practitioners. The reason for this may be that risk management in software projects always contains a dilemma: one expects reliable results with little effort and investment. After all, a project's goal is to deliver products, not to spend all of its time on pondering hypothetical problems. Nevertheless, failure to account for the limitations in the risk management approach used may result in serious bias in risk management results.

In this chapter, we highlight some of the main limitations that affect the applicability of many risk management approaches. Sometimes these limitations may have little practical relevance and they do not necessarily indicate that a risk management method does not work in practice. However, most of them have a high potential for creating bias in risk analysis so we recommend that any risk management program should take a conservative position and "prove" that the limitations are not serious in their situation.

Many risk management approaches address a limited number of goals, such as schedule, cost and product quality (Gemmer & Koch 1994; McCaugherty 1996; Sisti & Joseph 1994). There are many cases where projects actually have important other goals that eventually affect projects success, such as impact on reputation, ability to reuse projects results, compliance with constraints set to the project, need to maintain compatibility with other systems, and process conformance requirements. In fact, project's goals can rarely be truthfully expressed in two or three goals. If a risk management approach limits its risk identification and loss evaluation approaches to too few goals, some risks may be ignored or ranked lower than they should.

Few risk management approaches explicitly recognize the different expectations different project participants, stakeholders, have on the project and its goals. Sometimes the customer and other stakeholders are involved in the risk evaluation process (Sisti & Joseph 1994), but the involvement of such parties does not necessarily guarantee that their interests are supported in risk analysis phase. Even if other stakeholders are involved in the analysis, a joint, consensus-based ranking of goals may not be the most effective mechanism to deal with these stakeholder perspectives: it may increase communication overhead and sometimes politics prevent open discussion of critical issues when different stakeholders are present.

Many organizations have attempted to streamline their risk identification processes by developing risk checklists or taxonomies (Bezirkan & Mulazzani 1994; Boehm 1989; Carr et al. 1993; Jones 1994; Rook & Cowderoy 1993; Speaker 1993). These can be helpful tools in making sure that all previously identified categories of risk are covered. However, such taxonomies may also increase the tendency of participants to focus on the issues covered by the checklist and limit their ability to use their independent judgment to identify risks outside the checklist. The results of such taxonomy or checklist based risk assessments are sensitive to the appropriateness of the taxonomy used for the project and situation. If the taxonomy does not cover the "right" risks in a situation, the results are likely to be wrong.

Another potential problem with taxonomies is that they inherently contain trade-offs between coverage, detail, and user fatigue. A taxonomy that has broad coverage and is detailed may result in user fatigue: they tend to become less alert towards the end of the taxonomy list, possibly failing to recognize some risks.

Quantification and ranking of risks is a widely recognized challenge in risk management (Boehm 1991; Charette 1989; Friedman 1993). Normally risk ranking is based on estimating probabilities and losses. Probability estimates can be based on three main approaches: historical frequency data, subjective estimates, and estimation tables. All have potential major limitations that are rarely discussed by their proponents.

Historical data is rarely used for estimating probabilities, presumably because organizations rarely collect risk occurrence frequency data. This is actually not necessarily a drawback, as historical data's significance in predicting individual events is particularly questionable when situations change (French 1986). As each software projects is unique and technological changes are frequent, relevance of historical data is limited. However, historical frequency data can be used as a sanity check and reference point when the other two estimation methods are used.

Risk estimation tables have been used by many organizations to avoid subjective bias in probability estimates and to reduce the cost of risk analysis (Anon. 1988; Boehm 1991; Charette 1989; Karolak 1996; McCaugherty 1996; Sisti & Joseph 1994; Caplan 1994). These tables typically identify a set of factors – such as maturity of technology, complexity, requirements stability, and experience – and assign some probability score or value based on the "scores" on each factor. This approach has the following potential limitations and assumptions:

- Tables may produce a list of probabilities whose total may exceed one without any mechanism to account for joint probabilities.
- The same set of factors is used to evaluate all risks. Some risks may be influenced by other factors and the predefined set of factors may be a poor predictor of probability for such factors.
- The factors use same weights for all situations and risk items. The allocation of weights to factors is a subjective process and few, if any, table-based probability estimation approaches give details how the weights have been derived. Even if the weights are assumed to be representative in the general case, they may not be applicable in all situations.
- Scaling of scoring values for each factor is critical. Marginal increase of a value for a factor and its impact on probability should remain constant for all factors within a factor's value range. This is especially true if factor values are used in mathematical calculations. In other words, one should be able to assume that factor value scales are distance or ratio scale metrics
- Consolidation of factors should be based on the allowable mathematical operations, given the type of metrics used for factors. If factor values are not represented in absolute, ratio or distance scale metrics, they cannot be added or multiplied.

The above limitations are rarely addressed by table-based risk probability estimation approaches. The use of such tables is likely to lead to a consistent and low-cost, but unreliable and inaccurate risk estimation process.

Subjective probability estimates reflect a person's belief in the likelihood of a risk occurring (French 1986). Despite the subjectivity of such a definition of probability, there is a growing amount of research in dealing with such estimates (Kahneman et al. 1982; Tversky & Kahneman 1974; Kahneman & Tversky 1973). Subjective probability estimates, especially when done by individuals with access to past history data and good understanding of the domain, are perhaps the most reliable mechanism to estimate probabilities of future events.

Human experts may intuitively be able to process domain information to yield best available probability estimates. In order to compensate for individual bias, such estimates should be collected from several individuals and results discussed to consolidate differences.

Several different approaches have been proposed for the estimation of losses. The most obvious method is subjective estimates; perhaps the most widely used approach. In addition, Boehm proposed the use of cost models, network analysis, and quality factor analysis (Boehm 1989).

When a risk affects more than one valuable characteristic (a goal) in a project, the ranking of losses easily becomes non-trivial. Table-based estimation approaches have been used for loss estimation by many of the same approaches that use table-based probability estimation (Anon. 1988; Boehm 1991; Charette 1989; Karolak 1996; McCaugherty 1996; Sisti & Joseph 1994) and the same, often serious, limitations apply to such estimates. Although the decision analysis field has studied multiple criteria decision making problems extensively (Saaty 1982; French 1989), methods from that field have not been applied in software engineering risks management, despite their obvious potential.

Assuming that risk probability and losses have been estimated using distance scale metrics or better, the expected value formula can be used to quantify risks:

expected loss(event) = Probability(event) * Loss(event)

If probability and loss estimates are based on ordinal scale metrics, the above formula should not be used. Instead, risks can then be ranked by ranking tables where risks can be categorized in ordinal scale groups based on the loss and probability ranks. However, most approaches based on such tables (Sisti & Joseph 1994; Speaker 1993) use them inefficiently and fail to identify rankings within classes, resulting in unnecessary lack of precision, or do not include the rationale for the table priorities used (Greer et al. 1999; Newland et al. 1997). What is worse, it is also quite common to populate such risk tables by multiplying ordinal scale values of loss and probability, i.e., using a mathematically unsound operation and resulting in ranking risk rankings that cannot be justified (Conrow 2000).

We already discussed earlier the benefits associated with the use of utility theory in risk evaluation and the lack of use of this theory in software risk management. As most risk management methods are using the expected value of loss as the metric to prioritize risks, it is likely that many such evaluations have produced risk priorities that do not correspond to decision makers true preferences.

Finally, few risk management approaches give an accurate definition of risk. They mainly refer to risk as a “possibility of loss”. In practice this definition leaves open several alternative interpretations of risk, such as the actual loss that would result if the risk occurs (Anon. 1992), a factor or element that is associated with a threat (Anon. 1992), probability of a risk occurring, or a person that contributes to the possibility of loss (Anon.1995a). Clearly, there is some value in having such a broad and encompassing concept to facilitate initial discussion about risk. However, such ambiguity may eventually hinder analytical and detailed discussion about risk.

In summary, the software engineering risk management practice is using several methods that have potentially serious limitations of biases and the literature in the field rarely addresses these problems. With few exceptions (Conrow 2000), literature does not contain critical discussions of these limitations. We are afraid that as a result, practitioners are largely unaware of these limitations and continue to use such methods in critical software projects. It

that most organizations do not perform any systematic risk management in their projects, and of those that do, most use biased or incorrect methods to analyze their risks. Even if this statement is perhaps an exaggeration, the industry practices clearly have room for improvement.

2.5 Risk and Standards

The importance of risk management has also been recognized by some software engineering and quality standards. While these standards may not be the driving force in making risk management more common in industry, they do represent a gradually improving level of more systematic risk management in industry.

The IEEE Standard 1074 for Developing Software Life Cycle Processes (IEEE 1992) considers risk analysis a mandatory activity, requiring that "risk management is performed throughout the project's life cycle" and that "technical, economic, operational support, and schedule risks are identified and analyzed". However, the standard only gives recommendations on how risk management is carried out, it may include "modeling, simulation, prototyping, independent reviews and audits". The IEEE 1074 standard is a widely accepted and used standard in the U.S.

The IEEE standard 1058.1-1987 for project management plans describes the requirements for project management activity (IEEE 1987). One of the requirements is risk management, which is also an explicit section in the recommended project plan. The standard requires that risks are "identified and assessed" and "the mechanisms for tracking the various risk factors and contingency plans" are prescribed.

The U.S. DoD standard 2167A (DoD 1988), that describes the required software development processes of DoD contractors, states that contractors must "document and implement plans for risk management" and that the contractor shall "identify, analyze, prioritize, and monitor areas of the software development project that involve potential technical, cost, or schedule risk". The standard does not provide any further requirements.

The ISO 9000-3 guideline (ISO 1991b) for applying ISO 9001 standard (ISO 1987) to software does not address risk explicitly. However, the ISO 9000-3 does require that personnel have "freedom and authority to initiate action to prevent the occurrence of product nonconformity". Furthermore, it also requires that during contract review "possible contingencies or risks are identified". In summary, ISO 9000-3 only presents minimal and very general requirements for risk management and it cannot be said that ISO 9000-3 would support risk management. However, risk management would clearly contribute to the overall objectives of ISO 9000-3.

The Capability Maturity Model (CMM) of the Software Engineering Institute (SEI) requires risk management on level 2 (Paulk et al. 1993a). The process area of *project planning* has an activity (Activity 13) that requires that "the software risks associated with the cost, resource, schedule, and technical aspects of the project are identified, assessed, and documented". The same activity also requires that "risks are analyzed and prioritized based on their potential impact to the project" and that "contingencies for risks are identified". The key process area of *software project tracking and oversight* also requires that risks are tracked during the project and that "high-risk areas are reviewed with the project manager on a regular basis". Some other key process areas, such as *software quality assurance* and *software quality management* implicitly require some risk management activities to take place.

The IEC/ISO standard 15504, also known as the SPICE model (Dorling 1993), defines a framework for the assessment of software processes (ISO 1998c). Its reference model (ISO 1998b) defines requirements for the risk management process. It states that organizations should define the scope and strategies for risk management, identify and analyze risks, define metrics for risks, and take action to reduce risks.

The software engineering standard of the European Space Agency (ESA 1991) considers risk management to be a part of the quality assurance activity. The standard *requires* that projects identify risks and analyze their impact and these findings are documented in the project plan. Furthermore, the standard *strongly recommends* that the quality assurance function monitors how risk management is carried out in projects.

The ISO draft standard Information Technology Software Life-Cycle Process (ISO 1991a) also addresses risk, although only briefly: projects should manage technical, cost and schedule risks.

There are also several other standards that address risks from different perspectives. The standard 7799 defines the requirements for an information security management system (ISO 1999). It is supported by a software tool that helps implement a standard compliant process (ISO 2000). The ISO 13335 standard provides guidelines and techniques for managing the information technology asset security (ISO 1996; ISO 1998a). The IEC standard 60300 provides general guidelines on performing risk analysis of "technological systems", e.g., plants, business operations, or projects (Anon.1995b). The PD standard 6668 (Anon.2000a) provides guidelines and requirements for effective corporate governance.

The British standard 6079 defines a process for identifying, assessing, and controlling risks in projects (Anon.2000b). The standard defines a risk management process, gives guidelines on each step, provides example risk ranking tables, and contains a general checklist for project and business risks. The standard recognizes stakeholders and their impact to risk evaluation.

In summary, the software related standards contain only limited requirements and guidelines for risk management. The requirements are so general that quite simplistic risk management practices satisfy them. However, some of the other standards, as discussed above, provide more detailed requirements and support for risk management but, to our knowledge, they are not widely used in the software industry. It also seems that the more recent standards have more detailed requirements on risk management.

2.6 Conclusions on Literature Overview

Risk management is a relatively young but very multi-disciplinary field: as a formal and explicit activity it has been practiced and researched in many fields since the middle of 20th Century. The software engineering risk management will benefit from taking advantage of the contributions in other fields to develop techniques to support risk management. This is particularly important as the current state-of-practice in software development is based on very primitive – and often faulty – techniques. Software development is too important to be controlled by biased or superficial techniques.

The review of relevant literature highlighted several issues and contributions that are used later in this work. First, the requirement for systematic risk management is common in many disciplines, explicit and formal risk management practices need to be in place to ensure sufficient frequency and quality of risk management. Therefore, the methods and procedures

presented here are defined with sufficient detail and rigor that they can be applied systematically and consistently.

Second, the expected value of loss cannot be used as the prioritization criteria for risks. The utility theory has established that a more correct and realistic way of ranking risks is to use the expected utility loss. However, the prospect theory extends the concepts of utility theory and provides more insight into how risk estimation situations should be phrased to decision makers and how different biases can be controlled. Therefore, we have used the utility theory and key aspects of prospect theory as the underlying paradigms in the development of the Riskit method. More specifically, the following principles are used in Riskit:

- Utility loss is used as the criteria to evaluate losses associated to risks; prospect theory is used to augment this view and to support the control of biases in decision-making.
- Project goals are used as the reference point in assessing the loss of stakeholders.
- Empirical and historical data is used to support subjective probability estimates to avoid human estimation biases.

Third, the role of stakeholders and their perspective on losses should be made more explicit in risk management, as they are the ones that can determine the significance of potential losses of risks. We have developed the Riskit method so that it supports explicit links to stakeholders and their expectations.

Finally, risk management practice and understanding must be continually improved, both from the perspective of software industry, as well as from the perspective of each organization. The software industry is not using state-of-art knowledge and methods in risk management and we need to improve practitioners' awareness of more correct and more effective techniques. Each software development organization should also establish a risk management improvement framework that supports and forces them to learn from their past experiences to improve their understanding of risk and improve their risk management practice.

3. The Riskit Method

This chapter presents the Riskit method, i.e., the Riskit process model, as well as the eRiskit application that supports the method. The Riskit method presentation is based on a conceptual framework we have developed earlier for representing processes (Kontio 1995b; Kontio 1998). This chapter first introduces the framework and then presents the Riskit method using that framework.

3.1 Presentation Framework

In our earlier research we have developed and applied a conceptual framework for representing processes in a structured, hierarchical way (Kontio 1995b; Kontio 1998). This framework allows a conceptual decomposition and modularization of process information so that a process can be documented and adapted incrementally. The generic aspects of the process are documented explicitly and separately from how they might be implemented in different situations. Such a structure enhances the reusability of processes as well as it makes it easier to localize the processes.

We have based our process engineering framework on earlier work by several process modeling researchers (Armitage et al. 1995) and on some specific process architecture proposals (Curtis et al. 1992; Feiler & Humphrey 1993) and, most notably, on the component factory framework by Basili et al. (Basili & Caldiera 1991; Basili et al. 1991; Basili et al. 1992a). Our framework extends and details these earlier contributions (Kontio 1998). For the purposes of this work, we have simplified the framework slightly by including the concepts and terminology entity into the paradigm definition.

Our conceptual reference model is presented in Figure 11 and it is discussed in the following. The *paradigm definition* contains the main assumptions, principles, concepts, theories, methods, guidelines, and research methods that define the paradigm under which the process is enacted. In our architectural framework, the paradigm definition allows the documentation of main principles and interpretation rules for the process. The paradigm definition does not have to be comprehensive; it is adequate to document the main principles that correctly characterize the essential aspects of a paradigm.

Part of the paradigm definition is the definition of key concepts and terminology that are used in the process. Such concepts can be represented in a glossary or dictionary, but it is often beneficial to model key concepts more thoroughly using entity relationship diagrams, class diagrams (Awad et al. 1996; Rumbaugh et al. 1991) or other suitable formalism, such as state transition diagrams to represent artifact or process states (Harel 1987). In our Riskit process definition, we have presented the Riskit key terms and concepts as a glossary and as class diagrams that describe the relationships between main Riskit concepts.

The paradigm definition affects all layers of the process reference architecture, as the arrows in Figure 11 indicate. It determines some of the goals and functions for the process to be defined, as well as acts as a guideline at the enactment level. The paradigm definition can be expressed as a collection of statements or as a collection of main literature and references that characterize the paradigm for the process. In the Riskit process definition, we have used

a collection of statements that are associated with explanatory text and references to literature.

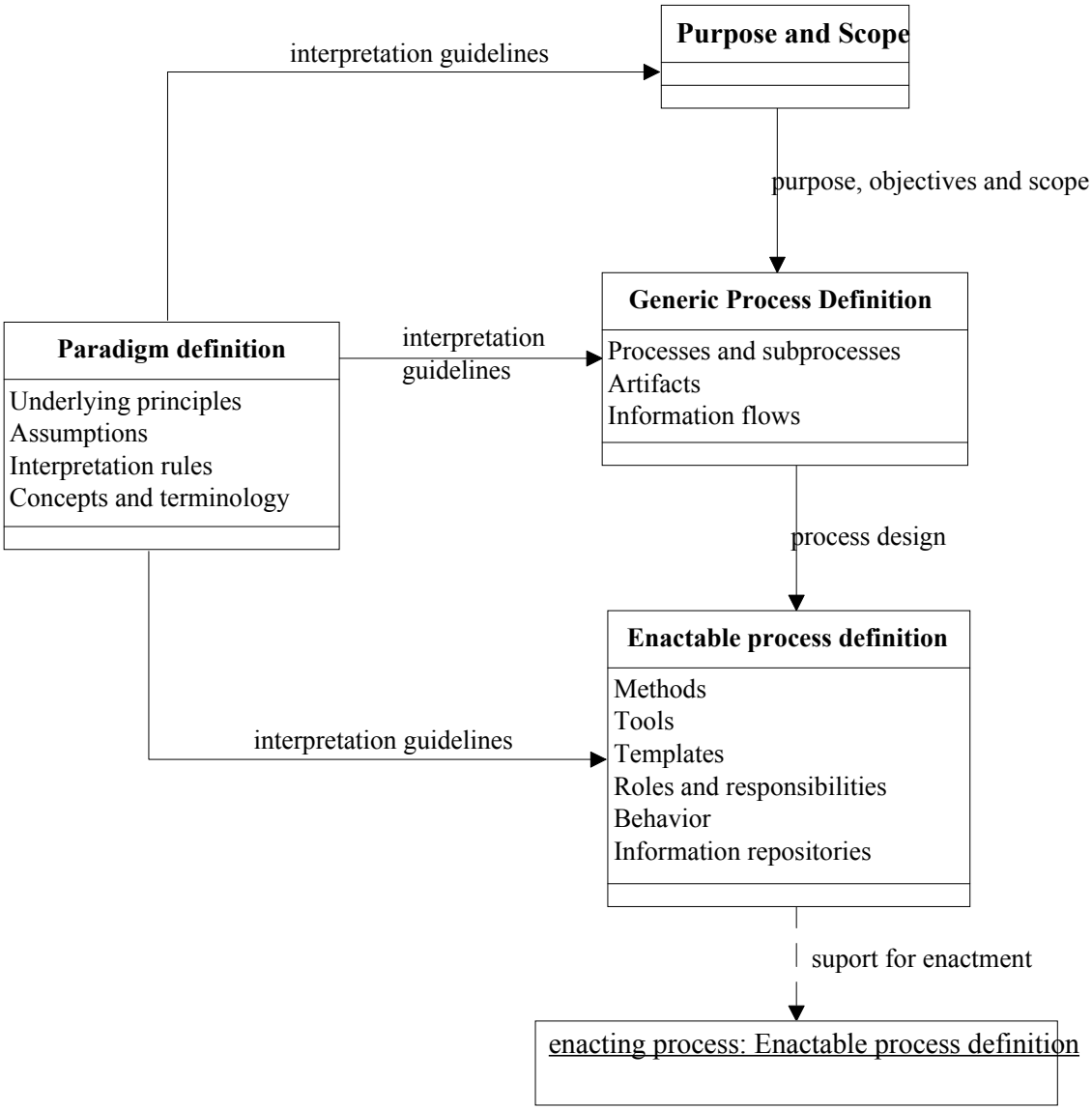


Figure 11: Process modeling architectural framework

The *purpose and scope definition*, also shown in Figure 11, states the goals, main output, and the scope of the process. The goals can be relatively abstract statements of purpose, such as "improve product quality" or "improve development cycle time". These high level goals are broken down to objectives (subgoals) or functions that the process should perform.

The *generic process definition* defines what are the main activities, artifacts, and information flows in the process. There can be several potential alternative generic process definitions that might satisfy the purpose and scope definition and representing them in an organization independent fashion allows comparison and evaluation between them. The generic process definition essentially corresponds to the functional perspective defined by Curtis et al. (Curtis et al. 1992). In the chapters that follow, the Riskit generic process definition has been presented as a dataflow diagram notation (Yourdon 1992), supported by a

process description template (see Table 9) and explanation text. We have adapted our process description template based on contributions in process modeling research (Kaltio 2001; Kontio 1994b; Radice et al. 1985). The symbols used in the dataflow diagram notation have been presented in Figure 12.

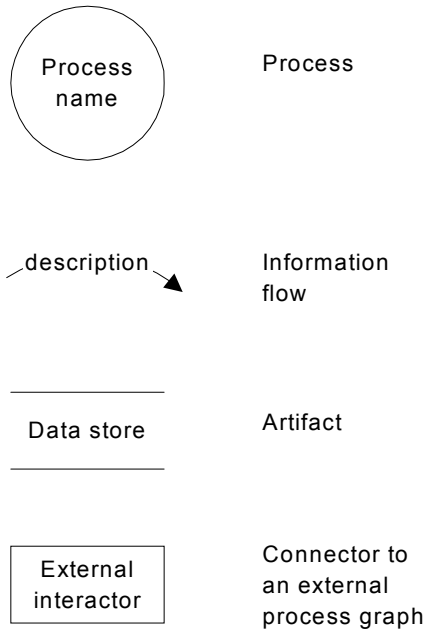


Figure 12: Dataflow diagram symbols used

The *enactable process definition* refines and localizes the generic process definition and describes how the process behaves. A process definition is enactable when adequate information on methods and tools, roles and responsibilities, templates, and behavior is available for process agents. What is "adequate" depends on the types of agents involved. A process engine requires detailed and formal specifications for all of these aspects, whereas people can perform a process with less formal specification. However, the skills, background, and experience of personnel also influence how much detail and formality are required to make a process model enactable.

Methods, tools, and templates define how things are done in a process. A generic process definition might state that a "high level design document" is produced in an "initial design" process. The method definition would indicate what approach, method and techniques are used in the process, the templates would indicate what format and style the output should be, and the tool definition determines what tools are used and how.

Purpose:	Purpose of the process.
Description:	Description of the process and approaches used in it.
Entry criteria	The criteria that is used to initiate the process. The criteria can include logical expressions, such "AND" or "OR". The logical expressions area used, statements are written within square brackets: "[statement]".
Input:	Input information required by the process.
Output:	Output produced by the process.
Methods and tools:	Methods and tools used by the process.
Responsibility:	A person or role that is responsible for the process.
Resources:	List of resource types that are used or participate in the process.
Exit criteria:	Exit criteria used to determine whether the process has been concluded. The criteria can include logical expressions, such as "AND" or "OR".

Table 9: Process definition information template

When roles and responsibilities are defined, they clarify, e.g., who enacts the process, what their responsibilities are, and how they interact. Again, level of detail in clarifying roles depends on the types of agents involved, but it also depends on the repeatability and level of control required from the process.

The behavioral specification can refer to several aspects of the process. It can include the definition of entry and exit criteria of activities, loops, and iterations in the process, alternative process paths, decision points in the process, dependencies of activities, and concurrency rules of activities. Whether all of these items are defined and how precisely they are defined depends on the process modeling objectives and on how much freedom is given to process agents in enacting the process. Experienced developers are able to enact a process with minimal or no behavioral specification, whereas a process engine requires detailed and unambiguous behavioral specification.

The information repositories refer to databases, documents, and other forms of information storage that is used in the process. For the Riskit process, these are described in chapter 4.5 and chapter 4.7.

The Riskit process definition does not contain a detailed enactable process definition, as the process definition presented here is meant to be a generic one that will be customized for each organization. However, we have presented some key methods in detail, provided references to other potentially relevant methods, and provided guidelines for the definition of roles and responsibilities and behavioral aspects of the process. We have included the definitions and examples of templates and methods in the generic process definition chapters in the following when the templates are simple and naturally fit into the description of the process in question.

3.2 The Riskit Paradigm Definition

3.2.1 Main Principles

3.2.1.1 Multiple Goals and Stakeholders

In goal-oriented situations, risk is a relative concept and it is dependent on the goals, expectations, and constraints involved. Therefore, the Riskit method extends the traditional definition of risks by including the goals and stakeholders as essential entities in defining risk, as presented earlier in Figure 1. The concept of risk is characterized by a probability associated with it, as well as the impact involved. Note that the general definition of risk refers to negative consequences, i.e., losses. We have used the term impact to also cover situations and risks that have at least some positive consequences. However, usually the focus of risk management is to manage negative impacts.

The definition and evaluation of impact depends on what the expectations or goals are in a situation. For example, suppose a risk causes the project to finish on May 1st instead of April 1st. If the expectation or objective was that the project should finish on April 1st, the risks that caused this delay was actually a risk. However, if the expectation was to complete project at the end of May and there are no other negative impacts, the valuation of the impact is quite different and it is even questionable whether the risk that happened should be even categorized as a risk.

The concept of a stakeholder in business context originates from organizational strategy research (Freeman 1984; Pouloudi 1999), and it has been used in the information system area to align information system development with corporate strategy (Lacity & Hirschheim 1995) and to support closer cooperation between users, developers, and management (Currie 2000; Lacity & Hirschheim 1995; Papazafeiropoulou et al. 2001). The stakeholder concept is also commonly used in public decision making (Accorsi et al. 1999; Bender et al. 1997). Lyytinen

and Hirschheim were among the first to highlight the link between stakeholders and information system failures (Lyytinen & Hirschheim 1987; Lyytinen 1988). In software risk management literature the importance of stakeholders has been discussed on a high level by several authors (Boehm 1989; Boehm & Ross R 1989; Charette 1989; Hall 1998), but most risk management methods do not explicitly support different stakeholder perspectives (Charette 1989; Fairley 1994; Gemmer & Koch 1994; Groth 1992; IEEE 1997; Michaels 1996) and those that do, often limit the number of stakeholders and assume that consensus can be reached (Pandelios 1996). Boehm's Win-Win approach is the only major risk management approach that focuses on stakeholder goals (Boehm & Bose P. 1994). The Riskit method extends Boehm's approach by maintaining links between risks and stakeholders explicitly. These links are visualized in Figure 1 (see page 5). The Riskit method contains templates and guidelines on how to identify, analyze and document all the elements listed in Figure 1.

Our definition of a stakeholder is based on Freeman's definition (Freeman 1984): stakeholder is any individual, group, organization, or institution who can affect, or be affected by, the software project or its results.

The expectations and goals are dependent on the stakeholders that are involved. Each stakeholder may have a different set of objectives and different priorities for them. Therefore, to value the impact it is necessary to know what the expectations are for each stakeholder. In fact, to obtain accurate prioritization of risks, the impacts should be prioritized for each stakeholder separately to take into account their potentially different priorities and preferences in their objectives.

When risk scenarios are defined, their impact to the project is described through the stated project goals. This allows full traceability between risks and goals and on to stakeholders: each risk can be described by its potential impact on the agreed project goals, and each stakeholder can use this information to rank risks from their perspective.

All projects have more than one stakeholder that is interested in its results, each with different priorities and expectations. The Riskit method recognizes these different stakeholders and risk management is based on balancing the stakeholder expectations in the risk analysis step.

Risks can affect several project goals simultaneously, e.g., cost, schedule, reputation, reliability, and functionality. In such cases, it is necessary to model and address all of these effects for risk analyses to be realistic. Many risk management approaches limit their view to the one or two most important goals at the cost of ignoring the others. The Riskit method views the evaluation of risks as a multiple criteria decision making problem and uses appropriate techniques to evaluate all affected goals when comparing loss scenarios.

While the Riskit method allows each stakeholder to have a unique set of goals, their target levels, and their relative priority, a project normally documents and formally agrees on a single set of objectives for it. Such goals, here called *committed goals*, represent an agreement among the stakeholder on what are the agreed, formal goals of the project. It may be that not all stakeholders' all goals are covered or that their priorities may be slightly different. Even when the committed goals have been defined and stated, it is still valuable to understand and document all the relevant goals for stakeholders as it will help project's risk management activities to prioritize risks more accurately so that all stakeholders' interests and preferences are taken into account.

3.2.1.2 Formal Definition of Risks

Risk is a fuzzy concept and it is necessary to define risk more precisely and formally in order to facilitate detailed and analytical discussion about risks. The common definitions of risk associate several different meanings to risk. It can refer to the *possibility* of loss, to the *actual loss* that would result if the risk occurs, to the *probability* associated with a loss, to a *person* that contributes to the possibility of loss, or to any *factor*, *object* or *course of action* that is associated with a potential loss (Anon. 1992; Anon. 1995a). All of these meanings are conceptually different and a single term fails to differentiate between the underlying meanings.

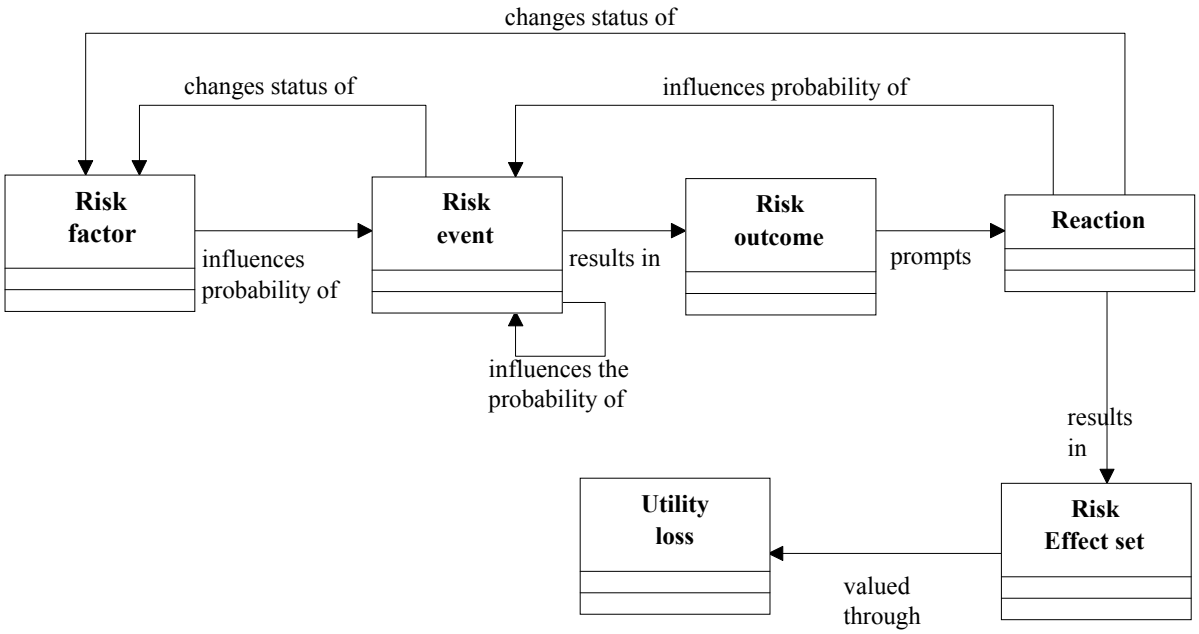


Figure 13: A simplified conceptual view of the elements in the Riskit Analysis Graph

The Riskit method supports unambiguous definition for risks by providing concepts that are more precise to model the different aspects of risks. However, the Riskit method also uses a common, more generic definition of risk as well, to provide a concept that can be used when the risk has not been analyzed into more precise components. Our definition of risk is as follows (Kontio 1997):

> *Risk refers to a possibility of loss, the loss itself, or any characteristic, object, or action that is associated with that possibility.*

The *Riskit Analysis Graph* is the technique that we use as a framework in more accurate risk description. The Riskit Analysis Graph will be presented and discussed in more detail in appendix 3.5, but we have included a high-level representation of the key elements and their relationships in Figure 13. Risk factors describe such characteristics of the environment that affect probabilities of negative events (i.e., risk events) occurring. Risk events document the potential events that might have undesired effect on the project. Risk outcomes express the resulting situation after the risk event but before any reactions are made. Risk reactions model the potential actions that can be taken after the risk event has occurred. Risk effect sets describe the impacts of risk events, including the effect that reactions may have had.

Finally, the utility loss defines the magnitude of the losses – or gains – associated with the risk scenario.

The Riskit Analysis Graph can be seen both as a conceptual template for defining risks, as well as a well-defined graphical modeling formalism.

3.2.1.3 Evaluation of Loss

The Riskit method uses utility theory and some key principles of prospect theory and other relevant theories about human decision making to evaluate losses between different risk scenarios. As we discussed in chapter 2.2, utility loss is a more accurate approach to evaluate and compare uncertain losses. The Riskit method models the impacts of each risk scenario as a set of effects, either positive or negative ones. Each such effect set is evaluated with respect to the utility loss it would cause, compared to the goals set for the project. In simple situations stakeholders can perform such an analysis by giving subjective preferences or rankings, in more complex situations one can use multiple criteria decision making tools to support such assessments, such as the Analytic Hierarchy Process by Saaty (Saaty 1990; Saaty 1992).

The utility loss is evaluated using the stated project goals as reference points. This will help participants to phrase the problem and prioritize risks from the reference point that is the planned and committed result of the project, as the prospect theory suggests (Kahneman et al. 1982).

3.2.1.4 Controlling of Biases

As we discussed in chapter 2.2, there are several potential biases that may threaten the accuracy of risk analysis. The Riskit method has incorporated steps and methods to minimize the potential impact of such biases using three strategies. First, the Riskit training package contains presentations, discussions, and examples on these biases, giving training participants hands-on experience and deeper understanding of them.

Second, we have developed a checklist that can be used during risk analysis to prompt critical review on whether such biases are present in a given analysis. The Appendix E contains this checklist and Table 10 lists which questions address each of the potential biases.

Third, many of the biases are also controlled by the characteristics or features of the Riskit method itself. These characteristics are listed in Table 10.

3.2.1.5 Subjective and Frequency-based Probability

As we discussed in chapter 1.1, software engineering domain is characterized by rapid changes in technology, software projects last a relatively long time, are costly, and each software project has unique aspects that reduce the similarities between projects. For these reasons the concept of frequency-based probability (Draper & Lawrence 1970) in software engineering is seldom practical (Singpurwalla & Wilson 1999). However, the software engineering practitioners have started to measure and accumulate data about the projects and many organizations do have statistical data about their past performance. Such data should be leveraged so that the probability estimates can be based on collected data.

The Riskit approach to probability is to use historical data, whenever available, as a baseline and adjust it on subjective basis to result in a subjective probability estimate that

takes into account the known changes between the current situation and the situations on which the data is based on.

Bias	Control in the Riskit method	Checklist question #
Insensitivity to prior information about probabilities	▪ Risk management database contains information about risk occurrences, lessons learned reports summarize past risk characteristics.	1, 2
Insensitivity to sample size	▪ Risk management database will eventually contain more data, increasing the sample size.	3, 4
Misconceptions about probabilities	▪ No specific support.	5
Insensitivity to predictability of information	▪ Participation of several individuals in assessments. ▪ Explicit documentation of estimates, review, and discussion of estimates.	6, 7
Illusions of validity	▪ Risk management database provides access to data on past projects ▪ History data and lessons learned reports are validated by the Experience Factory organization ▪ Participation of several individuals in assessments. ▪ Explicit documentation of estimates, review, and discussion of estimates.	7, 8
Misconceptions about regression	▪ No specific support.	9
Biases due to retrievability of instances	▪ Project context data is stored and available in the risk management database. ▪ Several people participate in risk management.	10
Biases due to the search strategy	▪ Same as above.	10
Bias related to imaginability	▪ Use of checklists to support risk identification. ▪ Participation of several people or experts in risk management.	11
Illusory correlation	▪ The Experience Factory organization analyzes and validates data and conclusions about past projects.	12
Insufficient adjustment	▪ Participation of several people or experts in risk management.	13
Evaluation of conjunctive and disjunctive events	▪ Risk Analysis Graphs allow accurate modeling of different scenarios, better understanding of them, and more accurate estimates of probabilities. ▪ Dependencies between risk scenarios modeled in Risk Analysis Graphs.	14, 15
Subjective probability distribution	▪ Worst and best case situations described for risk effects.	16
Impact of cognitive dissonance	▪ Risk management Experience Base provides information about project. ▪ Systematic and consistent enactment of the process.	17, 20
Ambiguity impact	▪ Participation of several people or experts in risk management.	
Mental compartmenting	▪ Stakeholder analysis and goal review make different interests explicit.	18, 19, 20
Regret theory	▪ Phrasing of loss evaluation to reflect project goals. ▪ Use of utility theory.	
Magical thinking	▪ Tracking the implementation of risk controlling actions.	
Disjunction effect	▪ Risk controlling action selection strategies support the selection of right actions.	21, 22

Table 10: Approaches in the Riskit method for controlling biases in risk analysis

3.2.1.6 Defined Process

Any project that has ambitious goals is likely to result in high workload for the project management personnel. In such situations, it is typical that high priority is given to urgent, immediate problems and secondary problems or tasks that influence longer-term issues are performed if time allows. Risk management may easily fall into the latter category,

potentially increasing the risk potential of the project drastically. In order to avoid this situation, the organization should require or enforce proper risk management to take place and provide the necessary, cost effective means to do so.

The Riskit method has been documented in detail (Kontio 1997) to support consistent and effective application of the method. The Riskit process definition contains descriptions of all main steps in the process, describes the methods and tools available for the steps, provides guidelines for the artifacts to be produced during the process, and proposes typical roles for the personnel performing risk management.

3.2.1.7 Learning through Experience

Ability to capture, analyze, and package experience is a prerequisite for systematic, planned improvements in software engineering (Basili 1989). The Riskit method has been designed to support systematic, experience-based learning and improvement. The main features that support this objective are the defined process that supports improved repeatability, consistency and fidelity of the process; the Riskit Analysis Graph and its underlying meta-model that supports accurate capture and documentation of risk information; and the associated risk management improvement framework that helps by making the improvement process itself a repeatable and systematic one. The improvement process is presented in chapter 4.

3.2.2 Concepts and Terms

In addition to the previously mentioned key principles and assumptions, the Riskit method also contains a set of terms that are used. These terms are defined in the following.

- The term *risk* in its general meaning is defined as a possibility of loss, the loss itself, or any characteristic, object, or action that is associated with that possibility.
- *Risk element* is defined as any item in the Riskit Analysis Graph (see chapter 3.5 for details):
  - *Risk factor* is a known fact or characteristic that influences some risk event.
  - *Risk event* is an occurrence of an incident with some negative consequences.
  - *Risk outcome* is the resulting situation after the risk event but before any reactions have taken place.
  - *Reaction* is a corrective action taken after the risk has occurred.
  - *Risk effect* is the combined impact of risk event and resulting reactions to goals of the project.
  - *Utility loss* is the harm a stakeholder experiences on a set of risk effects in a situation.
- *Risk scenario* is a combination of risk elements that describe the causes, triggering events and the impact of a risk. Normally a scenario consists of a risk event, risk reaction, and risk effect set.
- *Riskit Analysis Graph* is a graphical formalism used to document risk scenarios in the Riskit method.
- *Risk item*, or "raw risk", is defined as a risk that has not been analyzed and categorized into risk elements or described in the Riskit Analysis Graph.
- *Risk cluster* is a grouping of risk items.

- *Risk controlling action* is a proactive maneuver that is taken before risk occurs (or before it is known whether the risk has occurred).
- *Stakeholder* is any individual, group, organization, or institution who can affect, or be affected by, the software project or its results.
- *Goal* is defined as a characteristic that the project or product should have. Goals in the Riskit method are categorized into objectives, drivers, and constraints (see chapter 3.4.2, page 61).
- *Urgency* of risk refers to the time available until a decision must be made whether to control a risk and how. Urgency is, thus, a function of the delay of risk controlling action impact and the time of risk occurrence. See Figure 21 for details and discussion.

A more general glossary is presented in Appendix C, also including translations to Finnish.

3.3 Riskit Purpose and Scope

The purpose of the Riskit process is two-fold. First, the Riskit process aims at providing project management and project stakeholders with accurate and timely information about the risks and opportunities in the project. Understanding the risks associated with the project is necessary in making informed decisions about the project and its objectives, especially for assessing the potential business benefits, costs, and business risks associated with the project. In essence, the risk management process should help the organization to decide what risks are the most beneficial or profitable for the organization to take.

Second, the Riskit process provides a systematic way of identifying, analyzing and controlling risks that pose potential threats to project objectives so that overall risk profile and risk level is in balance with the business case and objectives for the project. There are two aspects to this activity; on one hand, the purpose is to identify, analyze and control risks to reduce the risk exposure overall, i.e., limit or reduce the probabilities and possible consequences of risks. On the other hand, risk management also attempts to identify and leverage opportunities that may be present and direct project so that it takes calculated risks that have the potential to bring along business benefits.

The scope of the Riskit process can be characterized by domain, temporal scope, and organizational scope. The Riskit method has been primarily developed to support software projects and their risk management in any application domain, e.g., embedded systems development, transaction systems, telecommunications applications, MIS applications, and e commerce, just to name a few. Software engineering in general is the context where all of our method development and empirical studies have taken place. However, apart from having different risk items, we do not see any reasons why the method would not be applicable to any other goal-driven activity, i.e., the method itself should be applicable to R&D work in general, project-based business, construction projects, even marketing campaigns. While such a generalization is a plausible assumption, this research does not intend to validate this assumption.

The temporal scope of the Riskit method covers the very early initiation phases of a software project to its completion. We have not designed the method specifically to maintenance or ramp-down phases of a product lifecycle, but as far as these activities can be considered goal-oriented undertakings, key elements of the method, again, maybe applicable.

In terms of organizational scope, the Riskit method has been targeted to project and program managers and their management teams. The project personnel should also participate in the process to provide the insight and information on which the risk management activity is based on. The upper management is also affected by the process, as they need to understand, support and enforce the risk management process and follow the risk management activities and their results.

3.4 Riskit Process

The Riskit process overview is presented in Figure 14. In this chapter, we will first give a brief overview of the Riskit process and then present a more detailed view of the Riskit process.

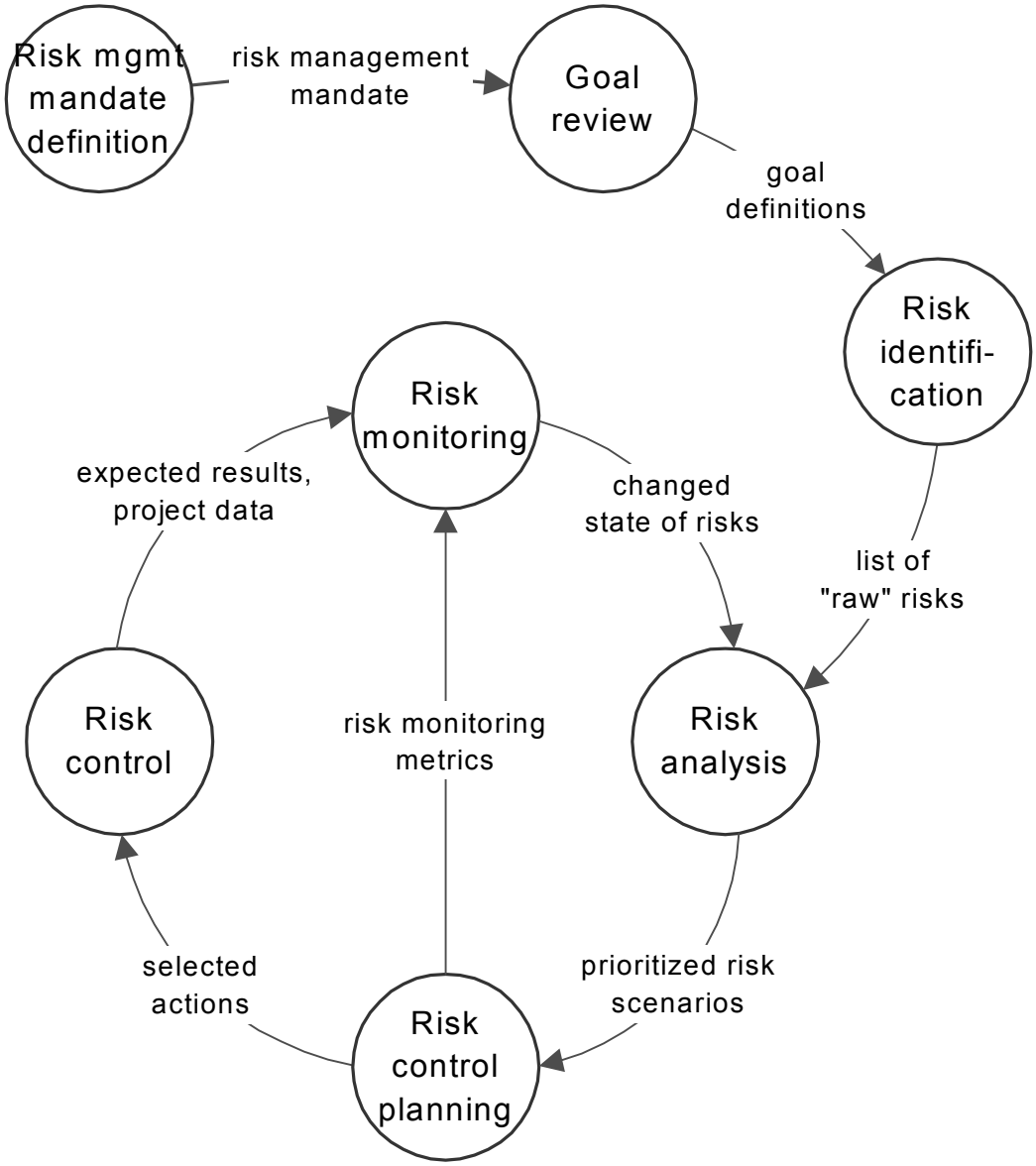


Figure 14: The Riskit process and main information flows

The first step in the Riskit process is the definition of the *risk management mandate*. The purpose of the risk management mandate is to clarify who is responsible for risk management

and how it should be done in the project. The risk management mandate defines the objectives and scope for risk management, assigns responsibilities and authority of risk management, defines what methods, procedures and reports are used for risk management, and how frequently risk management is done. The risk management mandate also documents who are the relevant stakeholders for the project.

The next step in the Riskit process is the *goal review*. Purpose of the goal review step is to understand and, if necessary, refine or revise project goals so that they reflect stakeholder interests and agreements, and make sure that they are well understood and documented. Goal definitions are used later in the risk analysis step so that all risks are linked to goals and their prioritization is done with respect to their impact on goals.

The purpose of the *risk identification* step is to produce a long list of potential risks in a project. These risks are called risk items or raw risks in the Riskit method as they represent un-analyzed ideas of potential threats and they are often presented as single sentences or with a few words. Identification should involve participants from all main areas of the project so that all risk areas are covered. Risk identification can be based on brainstorming, structured discussions, or use of checklists. Raw risks are typically clustered into groups for further analysis and discussion in the risk analysis step.

Risk analysis aims at two results. First, during risk analysis the identified risks are discussed and documented so that they are understood by participants. Different raw risks can be interpreted in different ways, many of them may be redundant with each other, and some may be wrong or inaccurate. Risks can be documented and analyzed using the Riskit analysis Graphs (example shown in Figure 15) or risk information sheets or forms (e.g., (Dorofee et al. 1996; Hall 1998)).

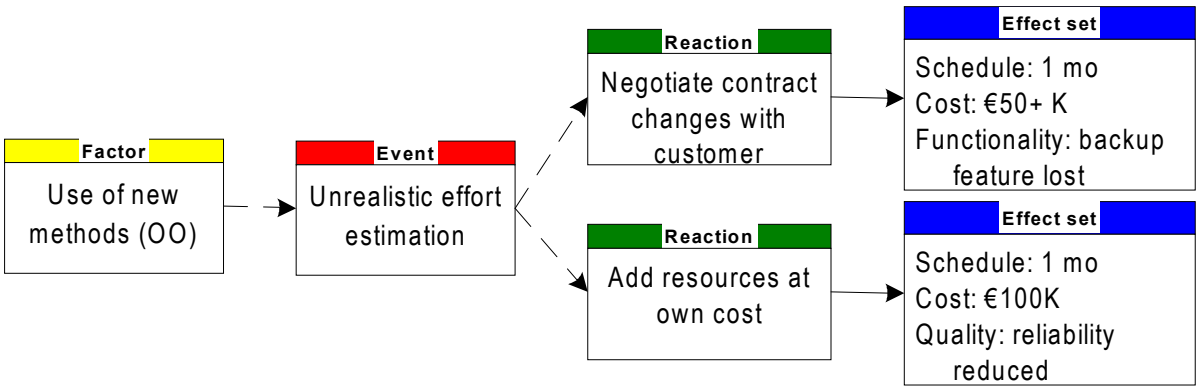


Figure 15: Example of the Riskit Analysis Graph

The second result of risk analysis is the prioritization of risks so that the most critical risks can be controlled. Risk prioritization is based on evaluation of probabilities and utility losses associated with each risk. The prioritization can be based on ratio scale metrics or estimates if such estimates are feasible, but they can also be ordinal scale rankings. In the latter case the Riskit Pareto ranking technique[3] is used to obtain partial prioritization of risks. The evaluation of losses is based on the utility theory, as we have discussed earlier.

[3] Discussed in more detail in chapter 3.4.4.3

Purpose:	Provide project and organization management with accurate and timely information of the risks in a project. Define and implement cost efficient actions to control risks.
Description:	Monitor and manage risks continuously in a project.
Entry criteria	Project planning has been initiated.
Input:	Project authorization information: goals, resources, schedule, and budget. Context and history information about the organization and its process.
Output:	Continually updated information about risks. Defined and implemented risk controlling actions. Experience and data about risks and risk management process.
Methods and tools:	The Riskit process definition. Riskit documentation templates. Riskit Analysis Graph definition and drawing tools. Risk identification checklists. Multiple criteria decision-making tools. Word-processing and spreadsheet software.
Responsibility:	Project manager.
Resources:	Technical personnel. Stakeholder representatives.
Exit criteria:	Project has been completed or terminated.

Table 11: Process definition information for the whole Riskit process

The risk *control planning* step identifies and selects controlling actions for the most critical risks in a project. Several alternative strategies can be used to select appropriate controlling actions, such as focusing on biggest risks, using risk reduction leverage as the criteria, or using stakeholder importance as the selection criteria. The most appropriate and effective risk controlling actions are selected and they are assigned to individuals in the project. Their implementation is considered to be part of the *risk control* step.

The *risk monitoring* step is a continuous activity that monitors risks, controlling actions, their effectiveness, and any changes on project situation that might affect risks.

Many of the Riskit steps can be enacted in parallel and more than one step may be performed in a single session. In addition, it is sometimes necessary to complete the risk management cycle for a critical risk quickly – before other risks have been even analyzed – so that effective controlling actions can be implemented immediately. Nevertheless, the Riskit process provides a systematic frame and guideline on how to identify, analyze, and control risks, even when the process itself is not explicit in a given software project.

Using the template introduced in Table 9, we have given a process definition for the whole Riskit process in Table 11.

While Table 11 presented a holistic view of the Riskit process, the following chapters present a more detailed view of the Riskit process, i.e., its sub-processes, artifacts used, information flows, resources used, and its behavior. We have elaborated the basic process given in Figure 14 and added the main feedback information flows in Figure 16. Each of the processes in Figure 16 can be instantiated several times during the project duration and they may be enacted concurrently. This behavioral aspect of the process is modeled through the entry and exit criteria defined for each process (Radice et al. 1985; Curtis et al. 1992; Kontio 1995b). However, typically the most critical instances of the risk management cycle are the ones enacted in the beginning of the project.

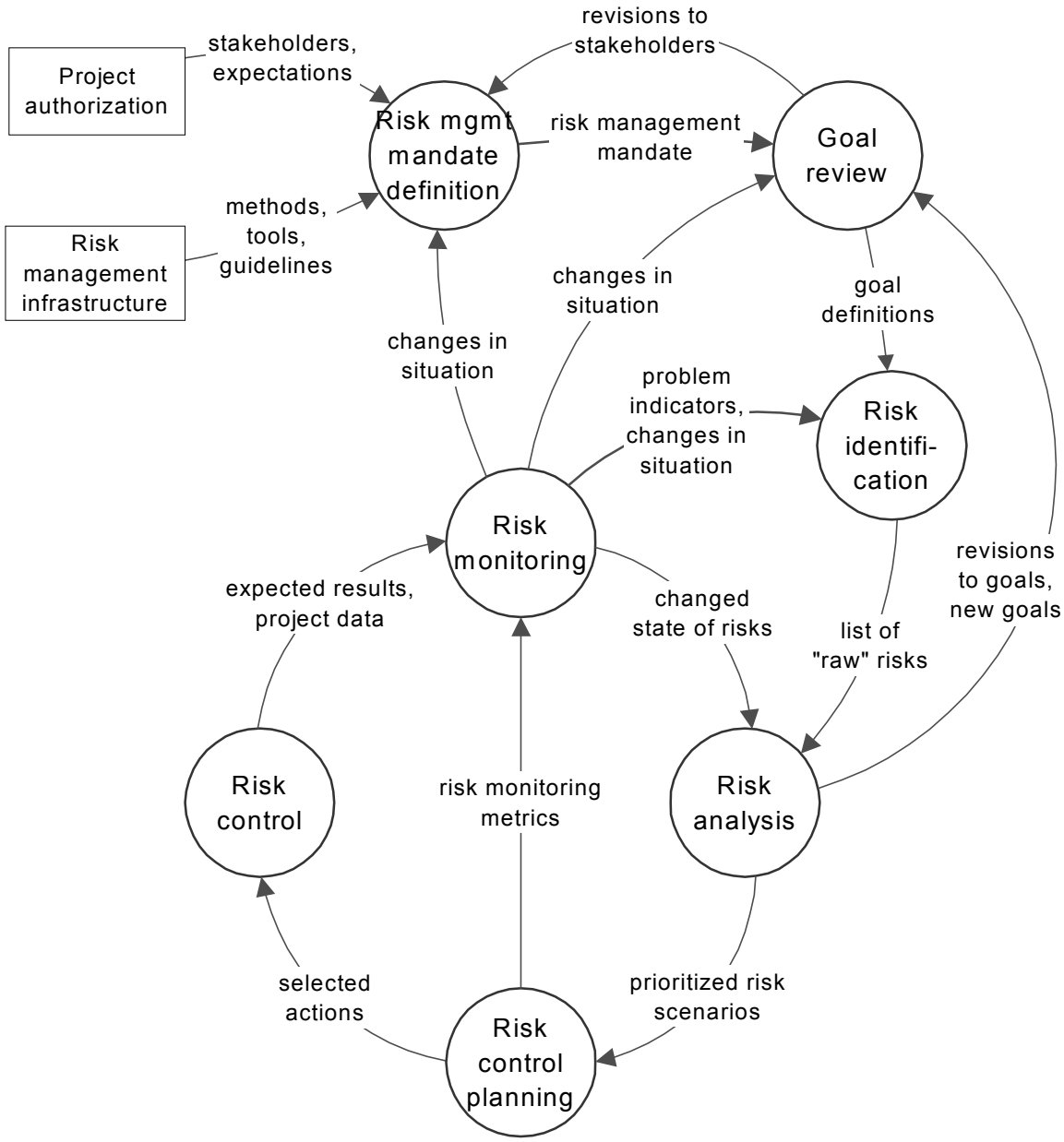


Figure 16: A detailed view of the Riskit process

We have presented summary descriptions of the Riskit sub-processes in Table 12, which presents summaries of the activities in the Riskit process, as well as the main output of each activity. Each process will be defined in more detail in the following chapters.

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	Identify potential threats to the project using multiple approaches.	A list of "raw" risks.
Risk analysis	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.	Completed Riskit Analysis Graphs for all analyzed risks. Ranked risk scenarios.
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.	Selected risk controlling actions.
Risk control	Implement the risk controlling actions.	Reduced risks.
Risk monitoring	Monitor the risk situation.	Risk status information.

Table 12: Overview of outputs and exit criteria of the Riskit process

3.4.1 Risk Management Mandate Definition

The risk management mandate is a project specific statement on the scope of risk management in a project. The responsibility of defining the risk management mandate belongs to the owner of the project, i.e., the entity that authorizes or funds the project or who will use or sell its results. Typically, project owner is the person or group of people to whom the project manager reports, e.g., a project steering group. The process definition information for this process has been presented in Table 13.

The risk management mandate definition process is initiated when any of the three entry conditions listed in Table 13 are met, i.e., when the project is initiated, or when there has been a change in stakeholders or in the overall risk level of the project. Stakeholder changes may be identified in other parts of the process, especially in the goal review and definition process and in the risk identification and analysis processes. The acceptance of stakeholders into the risk management mandate needs to be controlled by project owner and therefore it must be handled by this process. Likewise, if there has been a significant change in the overall risk level in a project, or the risk analysis has revealed that initial assumptions about the risk levels are not valid, the risk management mandate may need to be revised. For instance, additional resources may need to be allocated or specific areas of risk may be given a higher priority or more frequent reporting cycles.

Purpose:	Define the scope and frequency of risk management.
Description:	Defines the responsibility, authority, scope, and focus of risk management in a project.
Entry criteria	[project planning has been initiated] OR [stakeholders have changed] OR [project's overall risk level has changed] OR [stakeholder's risk tolerance has changed]
Input:	Project authorization information: goals, resources, schedule, budget. Organization's risk management policy and practice.
Output:	Risk management mandate.
Methods and tools:	(none defined)
Responsibility:	Project owner or project manager.
Resources:	Project owner, project manager.
Exit criteria:	Risk management mandate documented and approved.

Table 13: The process definition information for the *risk management mandate definition* process

The risk management mandate defines which stakeholders are to be defended in risk management, stakeholders' priorities, which risks may be excluded from project management's risk management scope (e.g., organization management may be willing to take responsibility of some risks without burdening project with any risk controlling responsibility), and how often and on what level of detail risks should be managed. The risk management mandate may also define any other procedures that are not addressed by the existing risk management infrastructure, i.e., the current methods, processes and tools that have been defined for risk management. A template for defining the risk management mandate is presented in Table 14.

Risk management mandate can be documented as a chapter in a project plan or as a separate risk management plan. Several examples exists for an outline of such a risk management plan (Charette 1989; Dorofee et al. 1996; Hall 1998; Newland et al. 1997). The risk management mandate definition process is concluded when all items listed in Table 14 have been addressed and approved.

Risk mgmt mandate attribute	Description	Example
Objectives	Statement of main objectives for risk management.	*"The objective of risk management in this program is to prevent major risks from occurring, keep project owners informed of the risk situation in the program and, when feasible, estimate the size of risks in the program."*
Scope	Definition of scope for risk management: what areas of risk should be covered and what level of detail should be involved.	*"The program is responsible for managing all technical, personnel and project management related risks."*
Risk management authority	Definition of authority or budget available for risk management.	*"The program manager has been allocated 12 person months of developer time for implementing risk controlling actions. Additional risk controlling action expenditures have to be approved by the steering group."*
Accepted risks	Description of risks that the project owners have accepted and are thus excluded from project's normal risk management scope.	*"Management takes responsibility for risks that deal with competitive situation changes and possible corporate reorganization."*
Risk management procedures	Description of the risk management procedures, methods, or techniques to be used.	*"The standard, corporate risk management procedures are followed in the program with following modifications:* • *A dedicated risk identification session is held every two months* • *Top 10 lists and corresponding risk controlling actions are included in monthly reports."*
Stakeholders	Identification of stakeholders and their priority.	*"The stakeholders covered in risk management, in order of their importance, are customer, division management, and sales division."*

Table 14: Risk management mandate definition template and example

3.4.2 Goal review

Risks do not exist without a reference to goals, expectations or constraints that are associated with a project. If goals are not recognized, risks that may affect them may be ignored totally, or, in the best case, they cannot be analyzed in any detail, as the reference level is not defined. Some of a project's goals typically have been explicitly defined but many relevant aspects that influence management decisions may be implicit. Therefore, it is necessary to begin the risk management process by a careful review, definition, and refinement of goals and expectations that are associated with a project. The definition of the goal review process is given in Table 15.

Purpose:	Define project's goals explicitly. Recognize all relevant stakeholders and their associations with the goals.
Description:	Existing goal definitions are reviewed and refined, if necessary, implicit goals are identified and defined. Different stakeholders are identified, their importance or priority defined, and their association and expectation levels with goals
Entry criteria	[project planning has been initiated] OR [new goals or stakeholders are identified] OR [a change in goals or stakeholders has been recognized]
Input:	Project authorization information: goals, resources, schedule, budget. Risk management mandate.
Output:	Goal definitions.
Methods and tools:	GQM (Basili et al. 1994a) Affinity grouping (Brassard & Ritter 1994; Dorofee et al. 1996)
Responsibility:	Project manager.
Resources:	Project owner, project stakeholders, project personnel.
Exit criteria:	Goals are explicitly documented and participants agree with their definition.

Table 15: The process definition information for the *goal review* process

In the Riskit method, we identify three different types of goals. We use the term goal to refer to any of them, i.e., a goal is a general statement of purpose, direction, or objective. When defined more accurately, we have found it useful to classify goals into three categories:

Objective: A goal that has an achievable, well-defined target level of achievement, e.g., "drive from A to B in one hour".

Driver: A goal that indicates a "direction" of intentions without clearly defined criteria for determining when the "goal" has been reached, e.g., "drive from A to B as fast as you can".

Constraint: A limitation or rule that must be respected, e.g., "… while obeying all traffic laws".

The review of project's goals often leads to definition of some additional, previously implicit objectives, drivers, and constraints. The purpose of this step is to produce formal definitions of these issues for the stakeholders that the project manager must satisfy. The goals are expressed using the template presented in Table 16.

As Table 16 indicates, goals are linked to different stakeholders that are associated with a project. This information will later be used in risk analysis to compare and rank risks. If new stakeholders are identified, they are defined and documented as described in the risk management mandate definition process. From the perspective of our process definition, a change in stakeholders initiates a new instance of the risk management mandate definition process.

Goal attribute	Description
Name	Name of the goal.
Type of goal	Objective / driver / constraint
Description	Description of the goal.
Stakeholder(s)	Names of the stakeholders for the goal that are interested in it.
Measurement unit	Measurement unit(s) used for the goal (e.g., $, date, or person-month).
Target value	Target value for the goal. Relevant for objectives and possibly for constraints.
Direction of increasing utility	Definition of whether an increase or decrease in goal value increases the utility. I.e., whether an increase in goal metric is good or bad. Stated as “growing” or “decreasing”.
Required value range	Minimum or maximum value required for the goal, if applicable.

Table 16: Goal definition template

The relationships between goals and stakeholders can also be documented using a stakeholder-goal priority table presented in Table 17. Such a table allows approximate prioritization of goals for each stakeholder: each cell in Table 17 documents relative importance of goals for each stakeholder. It is important to point out that if such rankings are documented for stakeholders, each column should be read and interpreted independently. Priority values *between* stakeholders for a given goal cannot be derived from such information. In other words, goal priority rankings should be interpreted only within a single column, not across columns in Table 17.

As shown in Table 17, the relative priorities between stakeholders can also be documented in stakeholder column headings. This information is initially defined in the risk management mandate definition process.

Stakeholders: **Goals:**	Stakeholder A priority: 1	Stakeholder B priority: 1	...	Stakeholder X priority: 2
Goal 1	1	2	...	4
...	...	...	...	...
Goal n	N/A	2	...	1

Table 17: An example of a stakeholder-goal priority table

The goal and stakeholder priority information is useful information for the risk analysis process as it allows more effective filtering and ranking of risks. Without such information project manager may be forced to make intuitive or undocumented judgment calls regarding which risks are selected for further analysis or how utility losses are prioritized. Note that it is usually adequate to provide ordinal scale partial rankings of these items, either by using predefined categories (e.g., low, medium, high) or defining priorities for these items.

Most important goals are often defined in the project plan or the project contract. However, not all of the goals may be in these documents. For instance, efficient resource utilization may be an important consideration for a contractor but this typically is not

considered a project goal. However, if these goals are real for some of the stakeholders in the project, they must be included in the risk management process. Goals can typically be found in the following areas:

- schedule;
- resources used, most often personnel time;
- cost of development;
- product requirements, which can include both functional and other quality characteristics;
- resource utilization; and
- technical constraints, such as hardware platforms, operating systems, and the use of particular software tools.

The goal review can be considered completed when project manager and stakeholders have reached an agreement on the goals and they are formally defined. However, the goal definition process may often need to be re-initiated as new goals are identified during the risk analysis process.

3.4.3 Risk Identification

The purpose of the risk identification process is to identify potential threats to the project and its stakeholders. Table 18 presents the process definition information for this process. As Figure 16 and Table 18 show, the risk identification process is initially carried out in the beginning of the project as its results are fed into the risk analysis process. The risk identification process is activated again when either of the two other conditions in the "entry criteria" row are met: if stakeholders or goals change or if the project situation changes.

The goal of the risk identification process is to produce a comprehensive list of all reasonable risks to the project. The mental mode of the identification process is to suggest many potential risks, not to analyze them. Analysis and filtering of risks produced will take place in the next step of the Riskit process. There are various techniques that can be used to facilitate effective risk element identification, such as brainstorming, checklists (Boehm 1989; Ropponen 1999; Moynihan 1997; Jones 1994; Honkonen 1999; Barki et al. 1993; Carr et al. 1993; Laitinen et al. 1993; Karolak 1996), critical path analysis, and even simulation and benchmarking (Boehm 1989). Based on our experiences, we recommend that informal techniques, such as brainstorming, are used in the beginning of risk analysis and more formal approaches are introduced gradually. This approach does not create initial bias in risk identification and it introduces formality as participants may start to lose their vigor in identification.

The risk list that is produced should be numbered or coded so that all risks can be traced throughout the risk management process.

There are two possible strategies for concluding risk identification process. The recommended approach is to conclude when no new reasonable risks are identified when alternative identification techniques are used. Such a situation would suggest that the identification process has exhausted all reasonable risks and further effort is no longer cost effective. However, this approach may be costly and subject to participant fatigue. An alternative approach is to set a predefined time limit, such as a single two-hour session, for risk identification. This approach can be justified by arguing that it is likely that most relevant risks are identified in the beginning and if adequate time is allotted, any remaining risks are not likely to be critical. Considering that risk identification is a critical activity and

it is not particularly expensive, we recommend that a conservative approach is used in terminating the risk identification process, i.e., it is better to keep on identifying new risks a bit too long than to stop the process too early.

Purpose:	Identify potential threats to the project.
Description:	Identify a large number of possible threats to the project using multiple approaches.
Entry criteria	[project planning has been initiated] OR [new goals or stakeholders are identified] OR [a change in goals or stakeholders has been recognized] OR [the time interval stated in risk management mandate has elapsed] OR [a significant change is project's situation has been recognized]
Input:	Project authorization information: goals, resources, schedule, and budget. Risk management mandate. Risk checklists, general (Carr et al. 1993; Laitinen et al. 1993) or organization-specific (Boehm 1991). Lessons learned reports from similar projects.
Output:	A "raw", numbered list of risks.
Methods and tools:	Brainstorming techniques. Goal and stakeholder driven identification approaches. Meeting aids. Interviews.
Responsibility:	Project manager.
Resources:	Project personnel. Risk management facilitator.
Exit criteria:	The marginal yield of risk identification approaches zero, even when identification techniques are changed, OR time or effort allocated for risk identification runs out.

Table 18: The process definition information for the *risk identification* process

Purpose:	Understand and prioritize risks.
Description:	Analyze risks and their components so that their probabilities and impacts can be assessed and most important risks recognized.
Entry criteria	Potential new risks are identified.
Input:	A list of risk items.
Output:	A prioritized list of risk scenarios.
Methods and tools:	Riskit Analysis Graph. Multiple criteria decision-making tools. Riskit Pareto ranking technique.
Responsibility:	Project manager.
Resources:	Selected project personnel. Risk management facilitator.
Exit criteria:	Participants agree on the priority of the most important risks.

Table 19: The process definition information for the *risk analysis* process

3.4.4 Risk Analysis

Risk analysis is a process where the "raw" risks from the risk identification process are grouped, filtered, and prioritized. The goal of this activity is to provide detailed descriptions of project's risks so that highest risk scenarios and appropriate risk controlling action can be planned and implemented in the next step of the Riskit cycle. Table 19 presents a summary of the risk analysis process.

Three main activities can be identified in the risk analysis process. First, raw risk items are clustered into sets, second, selected risks are documented as risk scenarios, and third, risk scenarios area ranked. Risk clustering and risk scenario development are iterative processes that interact with each other: developing a risk scenario may prompt revisions in risk clusters and vice versa. These relationships between the processes are represented in Figure 17. These processes will be discussed in the following chapters.

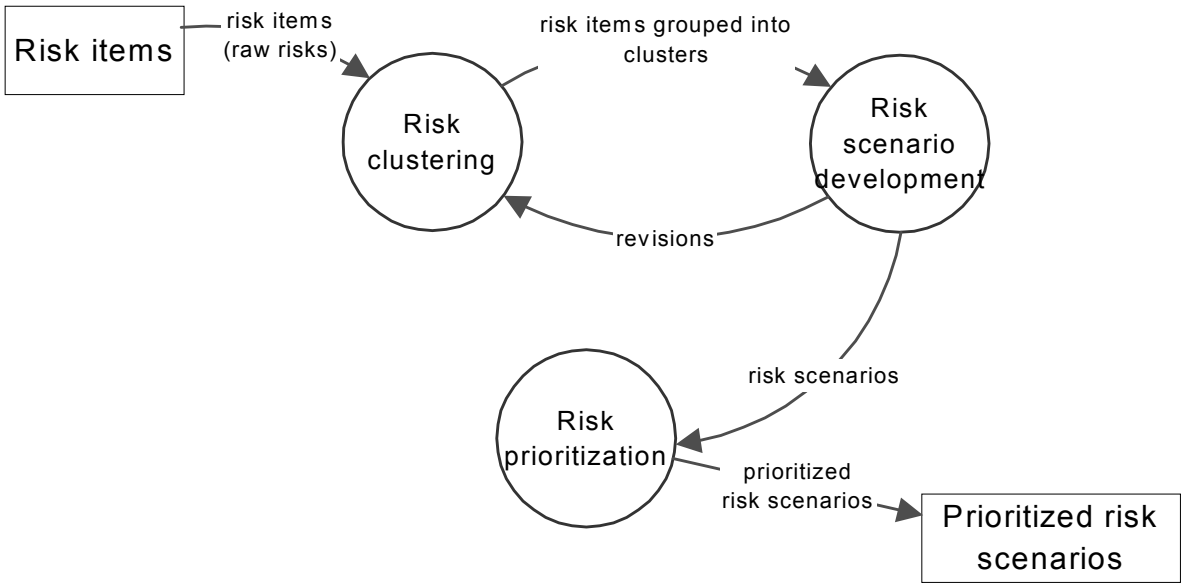


Figure 17: Sub-processes in risk analysis process

3.4.4.1 Risks Item Clustering

As the risk list produced by the risk identification process is an "un-analyzed" list of risks, it can contain redundant and overlapping items, as well as items on different levels of abstraction. If the risk identification process produced many such items, e.g., over 20, these risk items should be clustered into sets that contain similar risk items. This process is called risk item clustering and we have presented the process definition for it in Table 20.

Purpose:	Group “raw” risk items into clusters.
Description:	Group, decompose, merge or delete risk items into manageable clusters.
Entry criteria	Potential new risk items are identified.
Input:	A list of risk items.
Output:	Risk items grouped into clusters.
Methods and tools:	Word-processor, drawing tools.
Responsibility:	Project manager.
Resources:	Selected project personnel. Risk management facilitator.
Exit criteria:	All risks are included in the cluster set and number of clusters is manageable.

Table 20: The process definition information for the *risk item clustering* process

The purpose of risk item clustering is to provide a manageable intermediate step in risk management. The number of risk items produced in the risk identification process can be large and represent risks of different granularity. In many cases, it is meaningful to cluster these items into sets that contain risks that relate to same area or are otherwise similar. Possible criteria for “similarity” include

- type of risk: technical, personnel, organizational, quality, schedule, functionality, product structure, etc. Some of these can be divided further.
- criticality: some risks may be considered obviously critical already at the risk clustering step
- stakeholders: risks may be grouped by stakeholders, i.e., risks affecting mainly a single stakeholder are grouped into one set.

The definition of “similar” is subjective judgment and not overly critical, as all risk clusters will be analyzed further and developed into specific risk scenarios in the next step. Risk clusters mainly provide a temporary structuring mechanism for the “raw” risk items produced in the risk identification process. Analysis that is more detailed will be done in the scenario development process.

3.4.4.2 Risk Scenario Development

Risk scenario development provides the detailed documentation of risks that are selected for analysis. Risk scenarios are documented using the Riskit Analysis Graph (presented in chapter 3.5). One of the three different versions of the graph can be selected based on the level of granularity desired from the analysis, and the time available for the analysis. As a default, we recommend that the “normal” Riskit Analysis Graph be used (see Figure 24, page 90). We have presented a process definition for this sub-process in Table 21.

Purpose:	Develop risk scenarios for main risks.
Description:	Develop risk scenarios for main risks using the Riskit Analysis Graph.
Entry criteria	[risk clusters have become available] OR [new information becomes available and is not compatible with existing risk scenarios]
Input:	Risk items grouped into clusters.
Output:	Risk scenarios for most relevant risks.
Methods and tools:	Riskit Analysis Graph and drawing tool.
Responsibility:	Project manager.
Resources:	Selected project personnel. Risk management facilitator.
Exit criteria:	All selected scenarios are completed.

Table 21: The process definition information for the *risk scenario development* process

As there normally is limited time available for risk analysis, not all risk items from the risk identification process can be included in risk analysis. Therefore, selecting ("raw") risk items from risk clusters is an initial risk prioritization choice, yet this choice is made when the risks are not yet analyzed. To counter the possible bias caused by such an early selection, an adequate number of risk scenarios should be developed. In addition, all risk items should be explicitly decided upon, they should not be left out of the analysis only because they got lost among other risk items. Our rule of thumb is to select most important scenarios from remaining risk clusters and keep on developing scenarios them until several most recent scenarios have not resulted in risk controlling actions that will be implemented. The rationale of this strategy is that if, after careful analysis, additional risk scenarios do not result in cost effective risk controlling action, they are not considered big enough risks by decision makers.

When risk scenarios are developed, the items in relevant risk clusters can be reviewed as candidates for risk elements. As defined in chapter 3.5, risk elements are defined column by column, as shown by an example in Figure 18. The example in Figure 18 represents two scenarios as the event "unrealistic effort estimation" has two potential reactions, "accept delay and added cost" and "allocate more resources", both with different effect sets.

The first step in risk analysis, classifying risks into risk factors and risk events, is based on the risk list produced during the identification process. The categorization is based on the definitions given in chapter 3.5 and results are documented in the Riskit Analysis Graph (Table 36). An example of a Riskit Analysis Graph is given in Figure 18.

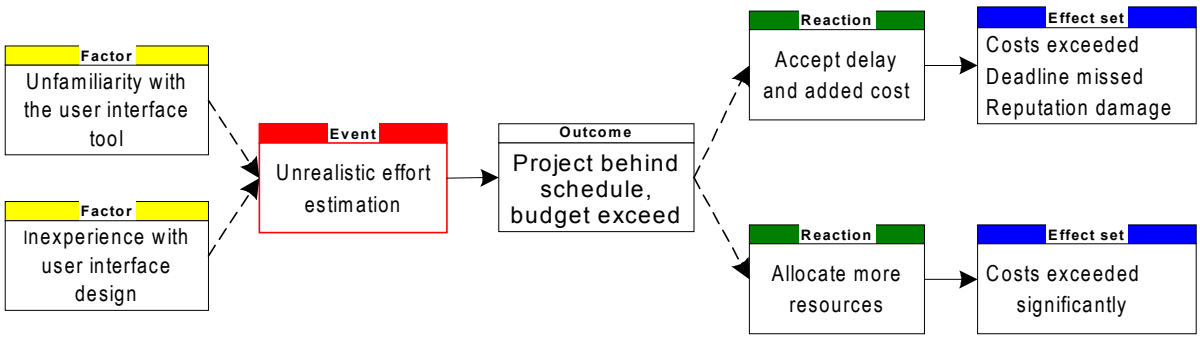


Figure 18: Example of the Riskit Analysis Graph

The Riskit Analysis Graphs can also be expressed in textual form using indentation, as is shown in Figure 19, using the same example as in Figure 18. However, the main disadvantage of this textual representation is that it is difficult – and sometimes impossible – to represent complex relationships between risk elements without duplicating them. For instance, when a risk factor influences several risk events, the textual form may become impractical. In such situations, one solution is to duplicate the risk factor items at each risk scenario. This reduces the visual power of representation and may create consistency problems when graphs are revised. Although we have developed tabular alternatives that avoid the redundancy problem, they seem to increase the complexity of the representation unnecessarily and as of now we are not recommending their use.

Factors:
- Unfamiliarity with the user interface tool
- Inexperience with user interface design

Event: Unrealistic Effort estimation
Outcome: Project behind schedule, exceeding budgets
Reaction: Accept delay and added cost
Effect set:
Costs exceeded
Deadline missed
Reputation damage
Reaction: Allocate more resources
Effect set:
Costs exceeded significantly

Figure 19: Textual version of the Riskit Analysis Graph

The main task of scenario development is not to map each risk item produced in the risk identification process into a Riskit Analysis Graph. Instead, judgment must be used to select scenarios that capture essential and representative future risk scenarios. The following criteria can be used when determining whether a scenario represents an appropriate set of future events:

- The risk event in the scenario represents event instances that are similar in nature and their probability can be estimated.
- The range of potential effects in the scenario is not extreme.
- Potential risk controlling actions are same or similar for all scenario instances.

If there is fuzziness or wide range of possibilities with either of the above criteria, the scenario should be potentially decomposed into two or more scenarios.

Each risk scenario is documented in the Riskit Analysis Graph and, depending on the available time and resources, each risk element is defined using the risk element definition templates presented in Table 22, Table 23, Table 24, and Table 25.

Risk factor attributes	Description
Name	Name of the risk factor to be used as an identifier.
Description	Description of the risk factor.
Normal/assumed level	Description of the “normal” level for the risk factor.
Project’s risk factor state	Description of the risk factors value for the project

Table 22: Risk factor definition template

Risk event attributes	Description
Name	Name of the risk event to be used as an identifier.
Description	Description of the risk event.
Probability of occurrence	Assessment of the probability of the event occurring.
Uncertainty of the estimate	Assessment of the uncertainty in the probability assessment.
Information source	Description of sources of information about the risk event for monitoring the changes in the probability or event occurrence.
Timeframe	Estimate of when the occurrence of risk will take place.

Table 23: Risk event definition template

Risk outcome attributes	Description
Name	Name of the risk outcome to be used as an identifier.
Description	Description of the outcome. A description of the project state after the event but before any other action is taken.
Certainty of the outcome	Assessment of the probability of the outcome if the risk event occurs (when not deterministic).

Table 24: Risk outcome definition template

Risk Reaction attributes	Description
Name	Name of the risk reaction to be used as an identifier.
Description	Description of the reaction. A description of the line of action or procedures that may be carried out if an event occurs.
Rationale	Description of the rationale for taking the action.

Table 25: Risk reaction definition template

The final step in risk scenario development is to estimate effects of scenarios. Effects are collected into sets and associated with each scenario. Effects are described through goals: each goal that is affected by a scenario is included in the effect set description and the effect is stated as a deviation from the goal or, in the cases of a driver, deviation from is expected. Depending on the estimation methods and tools available, the effects can be stated qualitatively (e.g., as textual descriptions or classifications high/medium/low) or quantitatively. Ranges can be expressed as well, if participants consider this necessary. Table 26 presents a template for describing scenario effects.

Note that effects should describe the net effect of the scenario after the event has occurred and reaction has taken place. It is often the case that not all goals are affected by a risk scenario, and sometimes the effects may be positive for some goals (e.g., loss of personnel may reduce costs while delaying schedule and limiting functionality).

Risk Effect set attributes	Description
Name	Name of the risk effect set to be used as an identifier.
Description	Itemized description of the effects. The effect description can be qualitative, i.e., a written characterization of the effect, or a quantitative estimate of the effect. Effects can also be described as ranges. All effects are defined in terms of the goals they affect. If an effect that is not previously documented as a goal is recognized, corresponding goal definition must be made.

Table 26: Risk Effect set definition template

Risks can also be documented using forms or tables to capture information about risks. The main benefit of forms is that they provide a standard structure and template for risk information, making the communication consistent. However, forms are often perceived as inflexible and lacking visual appeal and practitioners, in general, are not motivated to fill in and update lengthy forms. However, risk forms can be easily tailored to each situation and there are several sample forms available in the literature (Brassard & Ritter 1994; Charette 1989; Dorofee et al. 1996; Hall 1998).

ID	Description	Owner	Probability	Loss impact	Priority	Risk status	Controlling actions	Action Status
1	The required subcontractor resources may not be available when needed	J. Boss	High	Med	High	controlled	Negotiate a firm contract and guarantee	initiated
…								
28	The DB interfaces are not well known and may cause a delay	J. Date	Med	Med	Med	controlled	Perform tests ASAP	initiated

Table 27: Example of risk tracking table

The simplest kind of form is one where risks are documented in a table and short description of main attributes is included in each cell. The advantage of this approach is that it allows condense view to large number of risks, but, naturally, information about each risk is quite limited. Table 27 presents an example of such a table. Note that there are several other, alternative attributes that could be used as columns in such a table, such as

- Origin of risk

- Date of most recent review
- History data on the past status of risk
- Date of entry on the list
- Risk status
- Potential controlling actions
- Selected controlling actions
- Action Status

The choice of the attributes is a matter of balancing with the need to keep such tables simple versus including sufficient information about risks.

We have presented an example of a Riskit compatible form in Table 28.

ID:	**Project:**
Owner:	**Date reviewed:**
Priority:	**Timeframe:**
Probability:	**Loss:**

Description:	
Context:	
Factors:	
Reaction:	**Effect set:**
Reaction:	**Effect set:**
Reaction:	**Effect set:**

Risk mitigation strategy:			
Potential risk controlling actions:			
Selected risk controlling actions:	***Action***	***Responsible***	***Status***

Risk metrics:	***Metric/indicator***	***Status/Value***

Table 28: Example of a Riskit compatible risk form

3.4.4.3 Risk Prioritization

As resources for risk management are limited, it may not be feasible to mitigate or analyze all risk scenarios. Instead, one should focus on most important risks and spend relatively more time and resources on their management. In order to do this, it is necessary to rank the risk scenarios. Thus, the final step in risk analysis is to prioritize risk scenarios. The process definition for the risk prioritization process is given in Table 29.

In order to prioritize risk scenarios it is necessary to estimate the probability and utility loss associated with each scenario. These two estimation problems have different kinds of inherent difficulties. Probability estimation is difficult because little historical data may be available and event probabilities, in principle, are unknowable in a changing environment (French 1989; Kontio & Basili 1997). Utility loss estimation is difficult because there are multiple factors to be considered and the exact shapes and forms of stakeholders' utility functions are not known. We will discuss the estimation problems for each aspect of risk separately in the following.

Purpose:	Prioritize risk scenarios.
Description:	Based on the estimates for probability and utility loss for each scenario, prioritize scenarios with respect to their seriousness.
Entry criteria	[Risk scenarios have been completed] OR [new risk scenarios have been defined] OR [new information becomes available and is not compatible with existing prioritization]
Input:	Risk scenarios.
Output:	Partially prioritized risk scenarios.
Methods and tools:	Riskit Pareto ranking technique.
Responsibility:	Project manager.
Resources:	Selected project personnel. Risk management facilitator.
Exit criteria:	Selected scenarios have been ranked as well as available data allows.

Table 29: The process definition information for the *risk prioritization* process

If historical data about risks is available and if it can be safely assumed that the risk situation has not changed from the projects where the historical data was collected from, a frequency based interpretation of probability (French 1986) can be used and past risk occurrences can be used as an estimate of the probability. However, in software engineering context, such data hardly ever exists in adequate volume in order to be statistically reliable and the assumption about status quo is rarely realistic. Therefore, while historical data can be used as input to the estimation process, subjective probabilities, i.e., degrees of belief (French 1986), are often the only source for probability estimates. At the same time, it has been shown that direct query of numerical or verbal ratings of probabilities are not reliable (Tversky & Kahneman 1974) and more systematic approaches are relatively costly (Merkhofer 1987) for most software projects. Unless adequate time can be spent in probability estimate elicitation to control biases (Tversky & Kahneman 1974), we recommend

that scenario probabilities are ranked subjectively, resulting in ordinal scale rankings between scenarios. The use of ordinal scale rankings makes it clear that rankings are based on incomplete information and the impact of possible estimation errors or biases is reduced. If rankings are inconclusive between some scenarios, they can be evaluated further.

Risk scenario probabilities are estimated by ranking them based on their subjective probability of occurrence. We recommend that a pair-wise ranking approach (Saaty 1990) is used instead of using predefined ordinal scale values, due to potential problems with their interpretation (Tversky & Kahneman 1974; Merkhofer 1987). Using this approach the risk scenario ranking is done as follows:

1. Participants are asked to individually rank scenarios in decreasing order of subjective probability. To ease this process and to increase the accuracy of rankings, a pair-wise comparison approach can be used, such as the AHP method (Saaty 1990).
2. Resulting rankings are compared and discrepancies highlighted.
3. Discrepancies are discussed and resolved by consensus.

If consensus cannot be reached in a discussion, two strategies can be used to resolve the issue. First, the ranking of the debatable items can be increased so that a conservative interpretation is used: it is better to err on the side of caution than to assign too low a ranking. Second alternative is to group debatable items into same ranking category, i.e., collapsing rank categories into one category. However, in practice the pair-wise ranking approach may be too time-consuming and, due to time constraints, ordinal scale rankings can be used. In such cases, it is still useful to check the correctness of the rankings by selecting some pairs and asking participants to confirm whether the stated rankings apply to the pairs picked.

When a scenario contains only one probabilistic element, i.e., a single risk event followed by deterministic outcome, reaction, and effect set, the probability of the risk event and the risk scenario are the same. Probability estimation becomes more difficult when a scenario includes probabilistic elements, such as alternative outcomes or reactions. If reliable numerical (ratio scale) estimates for probabilities can be elicited and the probabilistic elements in the scenario are disjoint, the scenario probabilities can be calculated using straightforward conditional probability calculations. However, if such estimates are not available – which we believe is often the case – scenario probabilities must be estimated separately for each scenario. Given the increased complexity of such a situation, the estimation process should be carefully conducted so that complexity does not reduce estimation reliability unnecessarily.

The resulting list of scenario probabilities will be a partially ordered set, where some scenarios may be assigned into a same probability rank.

Estimation of the utility loss for each scenario is also constrained by the time availability for the analysis. In principle, utility losses between scenarios could be derived using various multiple criteria decision making tools (Saaty 1982; French 1986; French 1989). However, given the possible fuzziness involved in the effect sets and cost of applying such methods we recommend that similar ranking approach is used to rank utility losses between scenarios.

Utility losses of scenarios are ranked separately for each stakeholder. However, if stakeholder goal priorities are identical or very similar, such stakeholders can be ranked together. Such a joint ranking should be done cautiously because even when the goal priorities are similar, the underlying utility functions may be quite different. Merging stakeholder views, therefore, is a compromise that may sacrifice accuracy of stakeholder rankings.

Once the rankings for probabilities and utility losses have been obtained for all scenarios, they can be ranked. The concept of expected utility loss can be used to prioritize scenarios if both probability estimates and utility loss estimates have been estimated using distance or ratio scale metrics (Fenton 1991). In such a case the *expected utility loss* of a risk scenario can be calculated by the following formula:

expected utility loss(RS) = probability(RS) * utility loss(RS)

where "RS" indicates a given risk scenario.

However, this formula can rarely be used due to difficulties and costs involved in obtaining the distance or ratio scale metrics for the factors in the formula. Therefore, we have developed an alternative scenario ranking technique that can deal with ordinal scale estimates for probability and utility loss, yet provide reliable rankings of scenario risks. We call the ranking technique *Riskit Pareto ranking technique*. The Riskit Pareto ranking technique uses the probability and utility loss rankings of scenarios and searches for scenarios that are Pareto efficient over other scenarios, i.e., scenarios that are on the Pareto-efficient frontier[4] w.r.t. utility loss and probability ranks that have been used. Risk scenarios that are on the Pareto efficient frontier are not worse on either probability or utility loss estimate than any other risk scenario. This approach can be visualized with a simple table, as shown in Table 30: scenarios are positioned on the Riskit Pareto ranking table according to their rankings w.r.t. probability and utility loss. A scenario's Pareto efficiency over other scenarios can be easily assessed in the table: it is Pareto efficient if no other scenarios are in cell above it or left of it.

	Risk scenario probability				
Risk scenario Utility loss	*rank 1*	*rank 2*	*rank 3*	...	*rank n*
rank 1	scenario 1	scenario 2		...	
rank 2			scenario 3	...	
rank 3	scenario 4	scenario 5	scenario 6	...	
...	...	...	...	...	...
rank m		scenario 7		...	

Table 30: Risk scenario ranking table using Pareto-efficient sets

Using the Riskit Pareto ranking technique results in a partial ranking of risk scenarios, i.e., priorities for some scenarios can be defined but some scenarios' relative priority remains unknown. While the complete prioritization of scenarios would be desirable, the input data leading to the prioritization does not normally allow it.

In Table 30 scenario 1 is Pareto efficient over all other scenarios. The remaining scenarios can be only partially ranked based on the available information. The priority between scenarios 2 and 4 cannot be established but one can say that Scenarios 2 has higher priority

[4] Pareto-efficient frontier consists of alternatives that are Pareto optimal over other points in a set. An alternative *a* is considered Pareto optimal over *b* when $\forall$ *i* so that $a_i >= b_i$ and $\exists$ *i* so that $a_i > b_i$ for all i = 1, 2, 3, … n, where n is the number of criteria involved (French 1986; Keeney & Raiffa 1976).

than scenarios 3, 5, 6, and 7; and that scenario 4 has higher priority than scenarios 5, 6, and 7. The significance of these partial rankings is that they guide the focus of risk management to scenarios that have been reliably prioritized over other scenarios, given the information available. The risks should be considered for risk controlling action planning in their order of priority.

Note that Table 30 may tempt some users to rank all risk scenarios on the same diagonal as "equal" or "indifferent" and use this information to derive further rankings on the scenarios. According to this view, using the Table 30 as an example, one would assume that since scenarios 3 and 5 are "indifferent" and scenario 5 is higher risk than scenario 7, scenario 3 is higher risk than scenario 7. Such an interpretation would also seem intuitively correct, as people easily associate equal distance between the rank categories in the Riskit Pareto ranking table. However, we would caution against such an interpretation because ordinal scale metrics cannot convey enough information to assume transitivity of the "indifferent" relationship between scenarios. Since transitivity cannot be assumed, such logical conclusions are not justifiable[5].

The risk scenario rankings that are produced are stakeholder dependent. If more than one stakeholder is supported in the analysis, a corresponding number of utility loss rankings should be performed. However, if stakeholders have identical or similar goal priorities (see Table 17), their utility loss rankings can be merged to save time and reduce complexity of results[6]. This stakeholder view can be used in the risk control planning process to decide whether controlling actions for a risk scenario should be taken, what controlling actions should be taken and who should be covering the costs for them.

Once risk scenarios have been prioritized, even though partially so, the partially prioritized list of scenarios can be given as input to the next process in the Riskit cycle – risk control planning – and the risk analysis process can be terminated.

3.4.5 Risk Control Planning

The goal of risk control planning activity is to determine which risk controlling activities are necessary to take. The main issues in risk control planning are the identification of which risks pose greatest threats and the selection of appropriate risk controlling actions to mitigate them. The process definition for the risk prioritization process is given in Table 31.

Accomplishing these goals requires that both risk scenarios and risk controlling actions can be ranked or quantified. In most cases, neither task is trivial and we recommend the use of systematic approaches for ranking risks and risk controlling actions. Thus, risk control planning involves two main activities: defining possible risk controlling actions and selecting cost-effective risk controlling actions to be implemented. While these two activities are discussed sequentially in the following chapters, they are very much linked to each other. In fact, they should be seen as concurrent activities with continuous information exchange and incremental refinement.

[5] It is not safe to assume that relationships between scenarios 3 and 5 is that of "equivalent" nor that the relationship would be transitive. The priority of scenarios 3 and 5 is unknown and thus it cannot be used to draw conclusions on other priorities. For example, it would be wrong to conclude that because scenario 2 has higher risk than scenario 3 and because scenario 2 "is kind of same as" scenario 4, therefore scenario 4 has higher risk than scenario 3.

[6] It should be noted that this may cause error in rankings: although goal rankings may be similar, stakehoders' utility functions may be different and result in different utility rankings.

Purpose:	Propose and select cost effective risk controlling actions.
Description:	Define, prioritize, and select risk controlling actions for the risk scenarios that have been considered most important.
Entry criteria	Important risk scenarios have been identified.
Input:	Partially prioritized risk scenarios.
Output:	Selected risk controlling actions. Risk monitoring metrics.
Methods and tools:	Riskit element review. Riskit controlling action taxonomy.
Responsibility:	Project manager.
Resources:	Selected project personnel. Risk management facilitator.
Exit criteria:	All selected risk scenarios have been addressed.

Table 31: The process definition information for the *risk control planning* process

3.4.5.1 Defining Risk Controlling Action

Once the high-risk scenarios have been selected, possible controlling actions are proposed for each of them. Identifying possible controlling actions is a creative process and can be carried out in a free format manner. However, in order to ensure consistency and adequate consideration for all possible options we have developed two complementing techniques that can be used to support the identification of potential risk controlling actions. These two techniques are called the *Riskit element review* and the *Riskit controlling action taxonomy*. We will introduce both approaches in the following.

The Riskit element review is based on the risk elements presented in the Riskit Analysis Graph. This technique simply calls for a focused review of all risk elements in a scenario and prompts participants to consider ways to influence the elements by controlling them, finding alternatives, or preventing them. This review can be supported by questions presented in Table 32.

We have synthesized and detailed a high level taxonomy of risk controlling options from contributions by Boehm and Charette (Boehm 1989; Charette 1989; Charette 1990; Boehm 1991; Charette 1992). This taxonomy is presented in Figure 20 and discussed in the following.

The first set of options in Figure 20, *no risk reducing action* means that an organization does not take any immediate action to prepare for risk or to reduce risk. This option does not reduce the risk itself but may provide more information as time goes on. This option is recommended when there is not enough information to make a decision or if risks are too small to justify any other risk controlling action. Further action can be taken if new information motivates it. This option can be broken further into three options.

The *wait and see* option can be used in two situations. First, it is a good option for all risks that are considered small enough not to require any other action. Second, it can also be considered when there are no inexpensive ways of obtaining additional information and a major part of the risk is in the uncertainty of the of risk size of risk. In other words, the ranges of estimates of risk are wide and management has no special reason to believe that higher risk estimates are probable. This option, in fact, would be the same as the reactive

strategy we discussed earlier. Clearly, using this option to cover high uncertainty risks is, to say it simply, risky. A conservative approach would be to use some of the other options for high uncertainty risks.

Riskit element	Possible focusing review questions
Risk factor	• Can some risk factors be eliminated? • Can the situation described by some risk factors be improved or corrected? • Could the influence of current risk factors be reduced? • What other factors might compensate for influence of current risk factors?
Risk event	• What could be done to reduce the probability of risk event occurring? • Can there be a trial run? • Is training required? • Should the technology be evaluated, a prototype developed? • Can we learn from other people of projects?
Risk outcome	• Can alternative outcomes be created, e.g., more people assigned or trained?
Risk reaction	• What other reactions might be possible, can we do now to make them available? • Are more effective reactions possible? • Should we do more than just contingency plans?
Risk effect	• Can we compensate effects by some other means? • Can we protect some goals by some specific actions? • Are all goals equally critical?
Utility loss	• Are all expectations realistic? • What effects are not critical for utility loss? • Are there ways to reduce long-term utility loss?

Table 32: Supporting focus questions for Riskit element review

Buying information is an option that is used when the management does not have enough information to decide what to do about a risk and there is a possibility to obtain more information. In principle, it is only a temporary option that results in a new decision as the information becomes available. After additional information becomes available, some of the other options are selected. Buying information can take many forms. Sometimes information can be literally bought from outside sources, such as market research organizations or by hiring a consultant that knows about the area that risk is relevant to. However, more typical way of buying information is to develop prototypes, run simulations, initiate feasibility studies, or conduct performance tests.

Defining risk monitoring metrics is an option that should be selected for all or most risk scenarios, regardless of the other risk controlling actions produced. Risk monitoring metrics may include existing process or product measures but they can also include new metrics, or even informal information items, such as “observing personnel morale” or “monitoring database technology developments”. We recommend the use of GQM as a systematic method for defining such metrics, with modifications to include any information items as “leaves” in GQM trees, not just traditional metrics (Basili 1992; Basili et al. 1994a).

The second main option in Figure 20 is *contingency planning*. It means that recovery plans are made for a risk scenarios but no further action is taken. Strictly speaking, contingency plans have rather marginal effect on risk reduction as they mainly buy time and

marginal effort in advance. However, since they change the mode of risk control towards prevention this option has been listed separately from the previous option group. Contingency plans describe the actions that will be taken if the risk occurs. Plans are made and approved and they are put on hold to be used if risks occur. Contingency plans help organization make sure that there is a way to recover from the risk. Contingency plans can also be looked from another angle; they are, in effect, a way to detail the reactions and effects in risk scenarios.

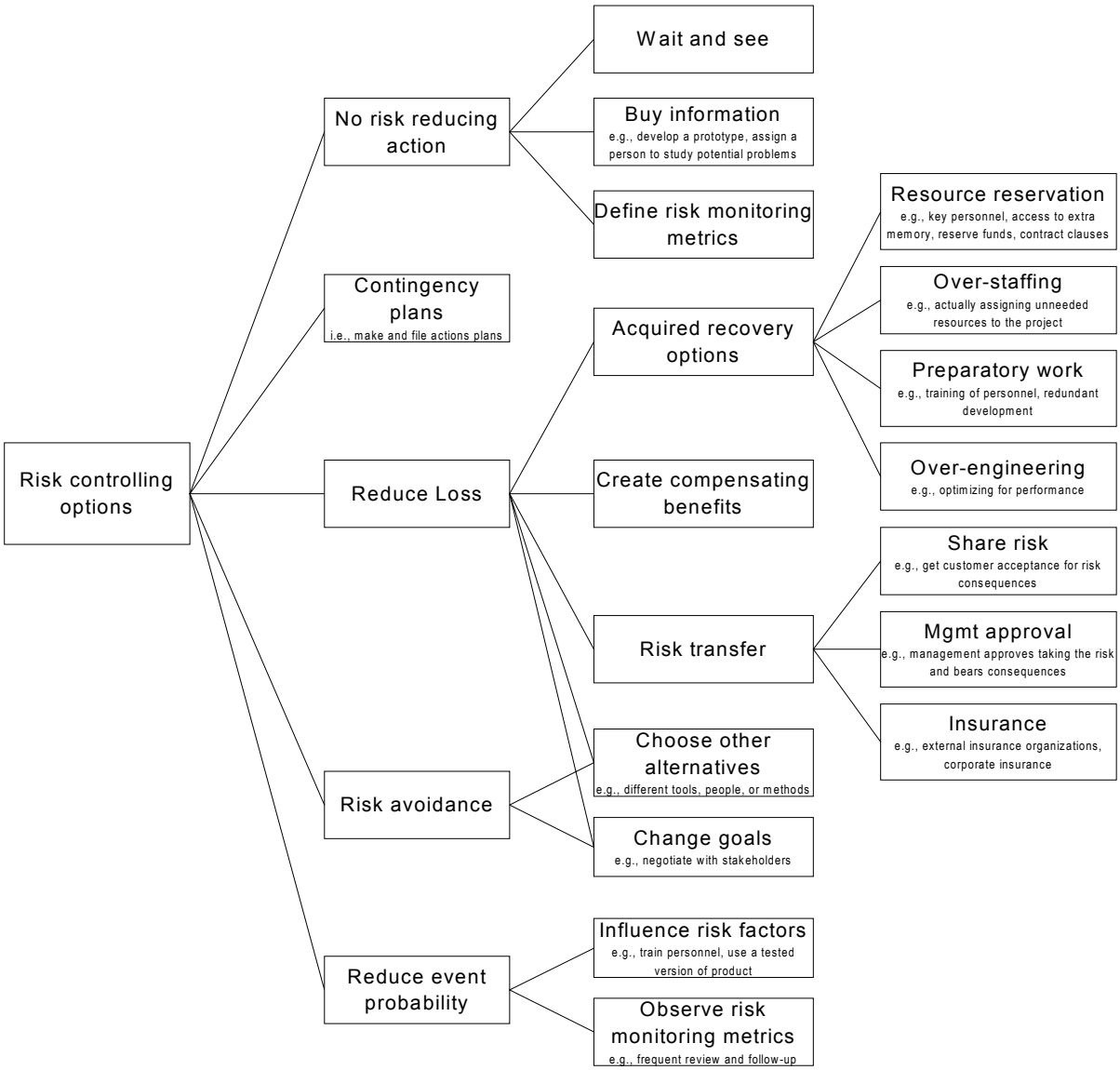


Figure 20: Risk controlling action taxonomy

The options under the term *Reduce loss* in Figure 20 refer to risk controlling actions that are aimed at mitigating the damage, i.e., effects or utility loss, caused by a risk. This option group has been divided further into five main groups, the two last ones are common to risk avoidance option group as well. The *Acquired recovery options* refers to a set of actions that buy options that can be used to limit the loss. They typically have a cost associated with them. The *Resource reservation* option refers to a situation where some resources are

reserved for limiting the impact of risk if the risk occurs. Resources can be human, computer or financial. Resources are not used before the risk occurs but they are reserved so that they can be "called to rescue" when necessary. The second sub-option, *over-staffing*, may be introduced to make sure that more than one person knows enough about each area in the project. Whereas resource reservation does not necessarily affect the project budget, over-staffing is likely to do so. The third sub-option in this group, *preparatory work*, refers to some action that is done in preparation of risks. It is akin to contingency plans but the main difference is that some work is done just in case: if the risk does not occur this preparatory work will have been wasted. Note that preparatory work itself normally does not reduce the probability of risk event occurring, it just mitigates the effects. Finally, *over-engineering* means implementing some features in the product or design so that there will be alternative ways of action if the risk occurs. For instance, over-engineering could mean that extra effort is spent during design or coding to make sure that alternative system architecture or compilers can be used.

The second option under reduce loss category is called *create compensating benefits*. This means actions that reduce the utility loss that the effect set of a scenario would otherwise cause. For instance, if a risk scenario would result in two-month schedule delay, the utility loss could be reduced by monetary compensation, free training, or having technical personnel at a customer site to resolve problems locally during the delay period.

The third option under reduce loss category is called *Risk transfer* and it includes three different options. *Sharing risks* means obtaining conditional approval for risks from stakeholders or project owners. Sharing can happen, e.g., with customers or subcontractors of the project. Again, a critical issue is to analyze the risks well and communicate their significance to all stakeholders. *Management approval* is similar to sharing of risk but instead of negotiating with stakeholders, organization management takes the responsibility of risk. Using this option normally means that the existence of risk is acknowledged and but there are no feasible ways to reduce risk to an acceptable level from the project perspective. An example of such a risk might be a development of a Windows 95 compatible software version to be able enter the market when Windows 95 was released: at early stages of development there may not have been enough technical and schedule information about Windows 95 to justify development, yet efficient enough risk controlling action might have meant missing the window of opportunity. In such cases, management can authorize the project to proceed in a line of action without spending time on controlling some "management approved" risks. Finally, the third option in this category is *insurance*, i.e., using external or corporate resources to insure for some risks. As real insurance options are rare in software development and as corporate, internal insurance schemes are, in effect, a variant of management approval, this option is rarely a realistic option.

Risk transfer should be used cautiously and limited to specific, clearly defined conditions or situations. Otherwise, it can become a "free ticket" for project management not to worry about risks. This may require contractual negotiations or explicit agreement what risks are transferred and what are not. Even if risk transfer option is used, other risk controlling options are often applied.

The two options under *risk avoidance*[7] are also shared by reduce loss option. It contains two sub-options. *Choose other alternatives* refers to an actions where alternative approaches,

[7] Although risk avoidance could be considered a special, extreme case of either reduce loss or reduce event probability, it has been given its own category for two reasons: it is frequently mentioned by other sources

methods, tools, resources or technologies are used. Each such alternative contains some characteristics that may contribute to some risks in a project, changing them may thus change the risk portfolio in a project. If decision about such alternatives are not yet made, the risk management process can easily contribute to the decision making process. If such decisions have already been made, it may be possible to consider revising such decisions, if that would prove to be the cost effective risk controlling action.

Changing of goals is a potential and often effective risk controlling action. As we have pointed earlier, the definition of loss, and therefore risk as well, depends on the goals defined for the project. If these goals have been initially unrealistic, the correct and cost effective risk controlling action may be to change these goals. This requires negotiation with stakeholders and project owners and such negotiations should be supported by the results of the risk management process. While changing of goals may be a tempting, easy way out of a risky situation, it is clear that management and stakeholders do not want to see it applied too often.

Finally, the last option category in our taxonomy is *reducing event probability*. We have divided it into two options: *influence risk factors* and *observe risk monitoring metrics*. Influencing risk factors can address any risk factor included in a risk scenario and can propose improvements in risk factors that reduce event probabilities. For instance, if "inexperience in user interface design" is a factor for an event "user interface not accepted by users", the action to improve inexperience in user interface design might be to provide training for personnel, and the use of a user interface design tool might be a factor that compensates for the inexperience.

It is important to point out that both of the techniques presented here, the Riskit element review (Table 33) and the Riskit controlling action taxonomy (Figure 20) are not meant to be comprehensive or normative guides to arrive at an optimal list of risk controlling actions. Instead, they are meant to act as supporting tools that augment the risk controlling action planning and extend the search space for controlling actions. The most critical aspect of risk controlling action planning is the involvement of project personnel and their ability to innovate effective actions.

3.4.5.2 Selecting Risk Controlling Action

Once the potential risk controlling actions have been identified, the next task is to select most effective ones to be implemented. It is recommended that more risk controlling actions are proposed than can be effectively implemented. This serves to confirm that the coverage of risk analysis and risk control planning has been adequate. If all proposed risk scenarios are selected to be targeted for risk controlling actions, it may indicate that risk scenarios not included in the risk control planning may need to be reconsidered. If all proposed risk controlling actions are implemented, this signals that not enough risk controlling actions were proposed and some beneficial actions may have been missed.

In the Riskit method, we use five criteria for selecting the risk controlling actions:

- Ranking of risk scenarios.
- Risk controlling action effectiveness.
- Resource availability.
- Stakeholder importance.
- Urgency of implementing the risk controlling action.

(Boehm 1989), and it does represent a slight paradigm shift in thinking about risk controlling actions. Therefore, it is justifiable to separate it.

We will discuss these each in the following.

The first criterion, *focusing on highest risk scenarios*, is an obvious one, i.e., mitigate the highest risk scenarios. The highest risk scenarios are recognized by the risk scenario ranking done previously. The number of risks to be mitigated can be determined subjectively, be based on a predetermined threshold or criteria, or be based on a Pareto diagram and on the point of diminishing returns (Michaels 1996). The most effective risk controlling action for each scenario is determined by estimating how much alternative actions reduce the expected utility loss. Given that expected utility loss is usually expressed as an ordinal scale rank between risk scenarios, the reduction of expected utility loss often remains a subjective judgment.

The focus on high-risk scenarios, however, is a slightly simplistic selection criterion. It does not account for the relative *efficiency of risk controlling actions* nor acknowledge possible resource constraints for implementing the actions. In principle, the risk reduction leverage proposed by Boehm (Boehm 1989) takes the effectiveness of proposed risk controlling actions into account. Within the context of the Riskit method, the risk reduction leverage should be applied to the utility loss of risk scenarios:

$$\text{risk reduction leverage} = \frac{\text{Expected utility loss}_{before} - \text{Expected utility loss}_{after}}{\text{Cost of risk controlling action}}$$

In other words, the reduction in expected utility is divided by the cost of the action that caused the reduction. However, the problem with this formula is that ordinal scale rankings of utility losses do not allow such a formula to be used. Even when distance or ratio scale estimates for utility loss were available, the accuracy of these estimates and those of cost estimations for controlling actions may make the formula impractical to use. In practice we recommend that the search for optimal risk is not even attempted, instead, relying on the principles of bounded rationality proposed by Herbert Simon (Simon 1979), subjective judgment is used to select risk controlling actions that are considered effective enough in the light of available information.

The third selection criteria are *resource constraints*. This can refer to the risk management budget if it has been defined or to the amount of available resources and skills. These constraints may rule out some otherwise effective risk controlling actions. For instance, training all personnel fully on a new method might have a very high-risk reduction leverage and reduce some key risks significantly but it may be unfeasible due to the cost and time delay involved.

Stakeholder importance and perspectives may also influence the selection of appropriate risk controlling actions. If stakeholders' risk scenario rankings differ, they may also have different preferences for implementing risk controlling actions. The process described here enables stakeholders to identify where their interests differ and where they are similar. For risk controlling actions that compete for limited resources, stakeholders should negotiate and, if possible, consider taking responsibility or covering the cost of some risk controlling actions that primarily control risks that are most relevant to them.

Finally, the *urgency of implementing the risk controlling action* may strongly influence on what risks to mitigate and how. The risk controlling action urgency depends both on the time of the risk event occurrence and the time delay in implementing the risk controlling action, as Figure 21 shows. The time of risk event occurrence, naturally, influences the urgency. However, the risk controlling action impact delay is often omitted from risk timeframe

analysis, although it can have a big impact on a situation as some risk controlling actions have long implementation delays. As time goes by, some risk controlling actions become infeasible if they are not considered early enough.

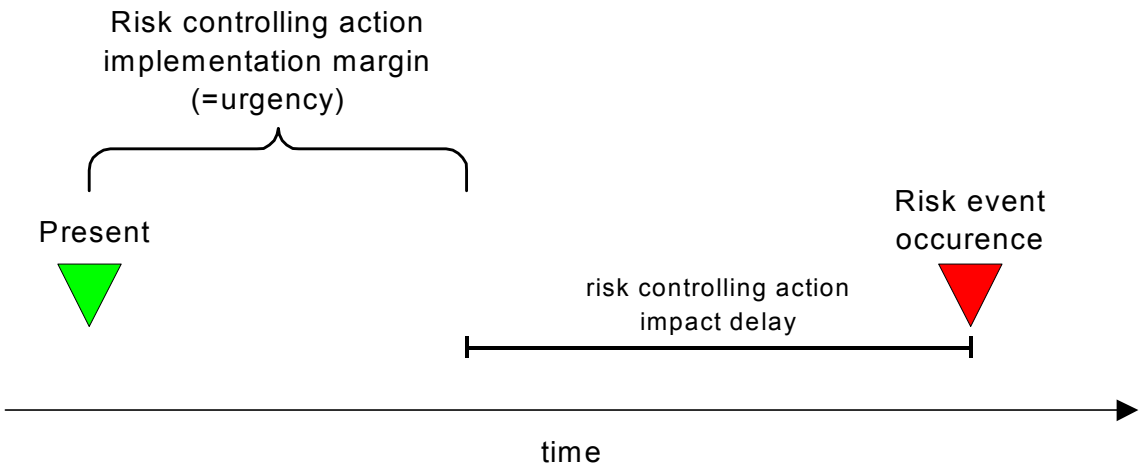


Figure 21: Risk controlling action urgency

We will use an example to illustrate the situation. Let us consider a risk event of "system architecture not defined by milestone 3". Whether this risk occurs or not will become clear at or near the milestone 3 which, in this example, is six months away and, therefore, not urgent. However, the two candidate risk controlling actions are "reuse the simpler architecture from previous system" and "recruiting of an experienced system architect". While the previous system's architecture can be reused with a short notice (perhaps within weeks), the recruiting and induction of new architecture expert could take several months. Therefore, the recruiting action has a higher urgency than reuse of old architecture.

The risk controlling actions are normally identified and defined based on risk scenarios that they are intended to mitigate. When such actions are defined, it is often the case that potential risk controlling actions overlap or have compound effects, or can even have counter-effects between them. Therefore, risk controlling actions should be seen as independent entities that can influence – i.e., reduce or increase – several risks when they are implemented. Such impacts should be considered when risk controlling actions are evaluated and selected. The eRiskit application supports this by allowing linking of controlling actions to many risks, a manual application of the method requires that such multiple or compound impacts are separately addressed, e.g., by dedicated step in the process or by using unique identifiers for risk controlling actions and forms and action tables.

The five criteria presented here for selecting risk controlling actions should be considered when deciding what action to take. While the importance of these criteria may vary between situations, based on our experience, the risk ranking and risk controlling action urgency are often the most important criteria. In any case, professional judgment must be used to conclude what actions to take, as the criteria themselves does not necessarily provide unambiguous conclusions.

The risk control planning process can be concluded when all selected risk scenarios have been addressed by the process and a decision has been made whether to implement controlling actions for them or not.

3.4.6 Risk Control

Once the risk controlling actions have been defined and selected, they become a part of project management. Their actual implementation is a project and organization specific management issue and the Riskit method itself does not provide detailed support on how this is done. However, the main requirements from this process are described in Table 33.

Purpose:	Implement risk controlling actions.
Description:	Implement the risk controlling actions defined by the risk control planning process.
Entry criteria	A risk controlling action has been selected for implementation.
Input:	Selected risk controlling actions.
Output:	Implemented risk controlling actions. Problem reports if problems arose in implementation.
Methods and tools:	(none defined)
Responsibility:	Project manager.
Resources:	Project personnel, external resources as needed.
Exit criteria:	Selected actions have been implemented.

Table 33: The process definition information for the *risk control* process

Note that the risk control process can be initiated as soon as the first risk controlling action has been selected for implementation. Should the planning for all actions for all scenarios take longer, the implementation of selected actions, of course, does not need to be postponed.

3.4.7 Risk Monitoring

The *risk monitoring* process is a continuous process that monitors the status of the project and the status of risk monitoring metrics. The risk monitoring process is defined in Table 34. The risk monitoring process is initiated as soon as the actual work in the project starts. In practice, however, the process is activated after the first cycle of risk management has been carried out, as risk identification and risk analysis largely perform the functions of risk monitoring during the first cycle.

Although the risk monitoring process has been defined as a continuous process, in practice the project status and risk monitoring metrics are reviewed at some frequent intervals. This frequency is defined in the risk management mandate, but our experience indicates that weekly or biweekly reviews are normal. The time interval can be adjusted based on the risk management needs of the project.

In practice, the actual enactment of the process may take place in a project management meeting where some other issues are also discussed. However, we recommend that the risk monitoring activity is a dedicated activity that is consciously performed with care. If needed, the monitoring process can immediately lead to launching of risk identification or risk analysis processes as required in the same session, or separate session can be scheduled if necessary.

Purpose:	Monitor the project and risk situation.
Description:	Continuously monitor risk monitoring metrics and the possible changes in project situation.
Entry criteria	Project has started. The process may be enacted on predefined frequencies.
Input:	Definitions for risk monitoring metrics. Risk management mandate. Goal definitions. Riskit Analysis Graphs.
Output:	Status reports.
Methods and tools:	Organization measurement program or database.
Responsibility:	Project manager.
Resources:	Project personnel.
Exit criteria:	Project has been concluded or terminated.

Table 34: The process definition information for the *risk monitoring* process

3.5 Riskit Analysis Graph

We introduced a simplified version of the underlying conceptual model – or meta-model – of the Riskit Analysis Graph components in Figure 13. The model in Figure 22 presents the same model having some additional information included, i.e., including multiplicity rules between elements. This meta-model represents the underlying, conceptual elements and their relationships. Each rectangle in the graph represents a risk element and each arrow describes the possible relationship between risk elements.

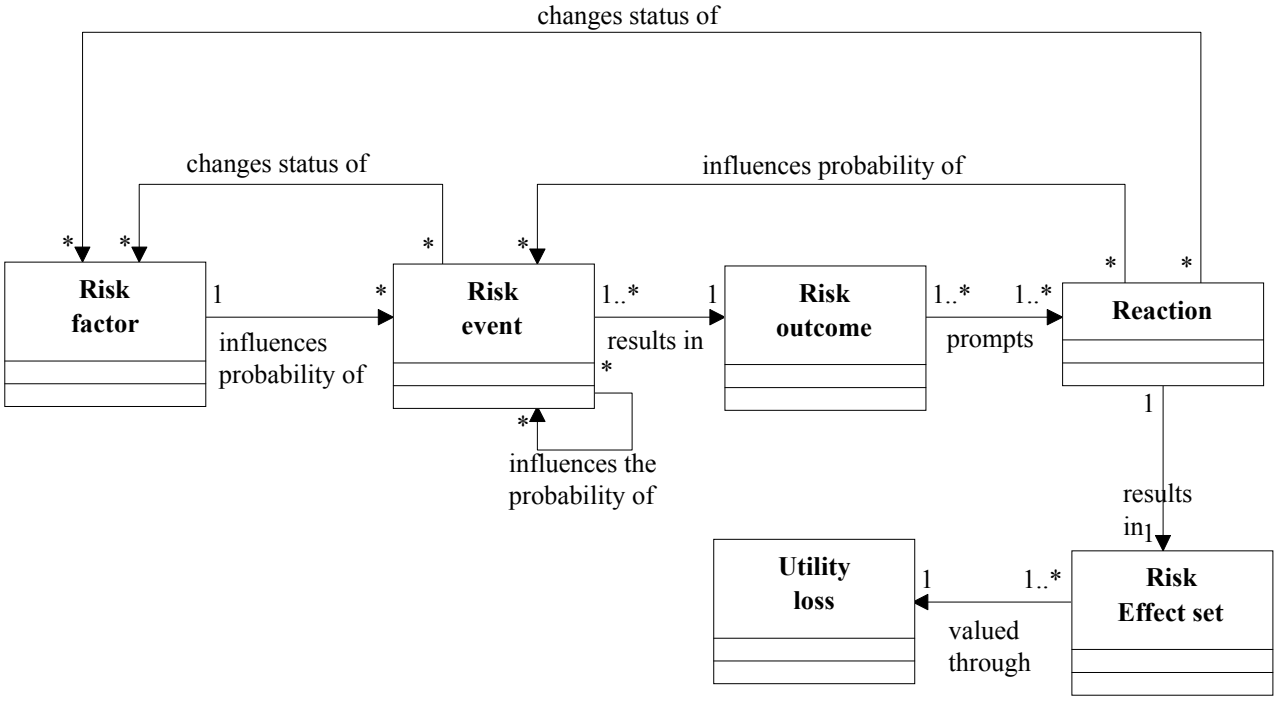


Figure 22: The full Riskit Analysis Graph

A risk factor is a characteristic that affects the probability of a negative event (i.e., risk event) occurring. A risk factor describes the characteristics of an environment. Consequently, in the Riskit Analysis Graph a risk factor does not have probability associated with it, it describes a relevant environment characteristic as it is or will be[8]. Examples of risk factors are listed in Table 35. Risk factors that are documented typically increase the probability of risk events occurring, but they may also reduce them, i.e., they are *success factors* for a project (e.g., "the development team recently developed a similar application").

The purpose of risk factors is not to document all possible characteristics that may influence a risk event as there may be an infinite number of such factors. Instead, a risk factor should document main assumptions of project environment and, especially, characteristics that are different from the assumed, "normal" situation. This interpretation of risk factors enables explicit documentation of main assumptions and deviations from these assumptions.

A *risk event* represents an occurrence of a negative incident – or a discovery of information that reveals negative circumstances. Risk event is a stochastic phenomenon, i.e., it is not known for certain whether it will happen or not. This uncertainty can be characterized by a probability estimate associated to the risk event. Examples of risk events are listed in Table 35. Each risk event can be influenced by many risk factors but a risk event does not have to have a risk factor associated with it. A risk event can also influence the probabilities of other events or even influence risk factors.

The next element in Figure 22 is called *risk outcome*. It represents the situation in a project after the risk event has occurred but before any corrective action is taken to reduce the effects of a risk event. Examples of outcomes are listed in Table 35. The purpose of the concept of risk outcome is to document the immediate results and situation after the risk occurs. Based on the risk outcome description, different reactions can sometimes be considered more objectively and creatively than directly from a risk event.

When a risk event occurs, the resulting risk outcome is rarely accepted as such. Instead, organizations react to the situation to reduce the negative impact of the risk event. These corrective reactions[9] are an important part of understanding what is the overall impact of the risk event to the project domain. Thus, each risk outcome is associated with one or more risk reactions: a *risk reaction* describes a possible action that can be taken as a response to risk event and resulting risk outcome. If only one risk reaction is described, it is deterministic: it will be taken if the event occurs. If more than one reaction is described, they represent alternative lines of actions. Risk reactions can influence the probabilities of risk events. If the influence is stochastic, they have a similar relationship as a risk factor has to a risk event: they change the probability of an event. Examples of risk reactions are also listed in Table 35.

The *risk effect set* represents the final impact of a risk event to the project. In other words, it documents what characteristics of the project were affected, taking into account the impact

[8] In practice it is possible that some risk factors are probabilistic, i.e., it is not known whether they are true for the environment or not. For instance, if new people are recruited for a project, it may be possible that a factor called "inexperienced personnel" becomes true. Such a situation is modeled by defining a risk event that influences a risk factor, i.e., risk event would be called "recruiting results in inexperienced personnel" and it would have a relationship to a factor called "inexperienced personnel".

[9] Note that we use the term "reaction" to action that is taken after the risk event occurs, as opposed to "risk controlling actions" that are taken before risk events occur.

of reactions. Effects are described through the explicitly stated goals for the project. Examples of different effects on goals are listed in Table 35.

While the risk effect represents the impact the risk had on each project goal, the concept of *utility loss* captures how severe the overall impact of effects is. The concept of utility loss is based on the utility theory, a concept widely used in economics and decision theory (Von Neumann & Morgenstern 1944; French 1989). The use of utility theory allows the simultaneous consideration of multiple criteria and consideration of several stakeholders. Furthermore, it is likely to result in more realistic evaluation of the losses as the utility functions of stakeholders are generally believed to be non-linear (Friedman & Savage 1948; Boehm 1981) and there may be points of discontinuity in them. We have sometimes used the term "pain" as a synonym for utility loss as the concept of utility may appear too theoretical for practitioners.

Risk element	Software Engineering Examples	General Examples
Risk factor	• inexperience of personnel • use of new methods • use of new tools • unstable requirements[10]	• a high cholesterol diet • living near a fault line of earth's plates (e.g., San Francisco) • slippery driving conditions (rain, snow)
Risk event	• a system crashes • a key person quits • extra time spent on learning a method • a major requirements change	• a doctor's diagnosis of a patients heart problem • an earthquake • a car accident
Risk outcome	• system out of operation • personnel and competence shortage • work behind schedule • new work required	• a diagnosed heart disease exists • some buildings and roads destroyed • a crash scene: untreated personal injuries, damaged vehicles
Risk reaction	• system operational after delay, back up data restored • recruiting process initiated, staff reassigned	• treatment of heart problem • reconstruction of roads and building • treatment of injuries, purchase new car
Risk effect	• added cost $50K • two-month calendar delay • some functionality lost • reputation as a reliable vendor damaged	• hospital stay, cost of medical care • cost and inconvenience of reconstruction, loss of human life, medical expenses • medical costs, permanent injury effects, raised insurance premiums
Utility loss	• The perceived harm experienced by a stakeholder, e.g., the board of directors, CEO, or personnel	• The net effect of pain, lost time and expenses as felt by individuals

Table 35: Examples of risk elements

[10] Note that this is different from "a change in requirements", which would be a risk event. When defined as a factor, "unstable requirements" refers to the characteristics of the situation.

The multiplicity (i.e., cardinality) information about risk element associations is included in Figure 22, using the UML class diagram notation and syntax[11]. A symbol in the beginning of an arrow indicates how many outgoing associations are allowed or required. Correspondingly, a symbol at the end of the association arrow indicates how many associations can be linked to an element.

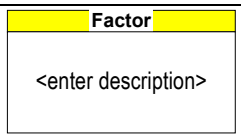
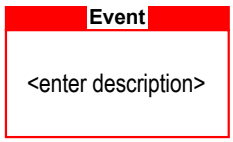
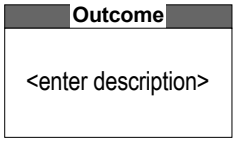
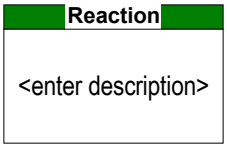
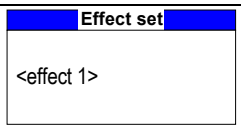
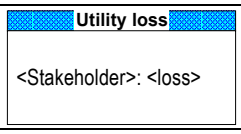
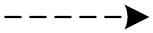
Symbol	Definition
	**Risk factor** (yellow banner). Represents risk factors. Risk factors name is entered in the symbol. The factor should be named so that its influence is unambiguous, e.g., one should name a factor "**limited** CASE experience" instead of just "CASE experience".
	**Risk event** (red banner). Represents risk events. Event name is entered in the symbol and the probability estimate of the event can be entered in the symbol as well.
	**Outcome** (gray banner). Represents the situation after the risk event has occurred but before reactions are carried out. Can be omitted.
	**Reaction** (green banner). Represents the actions that may be taken after the risk event has occurred. Descriptive name of the reaction entered in the symbol. The reaction symbol can be omitted from the graph for null reactions (i.e., when the reaction is "no reaction").
	**Effect set** (blue banner). Effect of a risk scenario to the situation. Each effect is described or quantified w.r.t. explicitly stated project goals. The effect is described as a deviation from the expected effect. If a goal is not affected, it is not listed.
	**Utility loss** (light blue banner). Documents the utility losses for each stakeholder. Can be omitted from the graph.
	Deterministic connector. Represents a certain relationship between risk elements in the Riskit Analysis Graph.
	**Stochastic connector**. The causality between risk elements is either probabilistic or is based on a decision to be made later.

Table 36: Riskit Analysis Graph symbols

The Riskit Analysis Graph uses specific symbols to represent risk elements. The allowed symbols in the Riskit Analysis Graph are defined in Table 36. The banners of the symbols are color-coded to support easier recognition of risk elements[12]. The Riskit symbols can be drawn manually or with any drawing tool. However, we have implemented a drawing

[11] The multiplicity symbols are interpreted as follows:

1	Exactly one association leaves or enters the class.
*	Any number of associations leave or enter the class
1..*	At least one association leaves or enters the class.

[12] The color of the symbol is mentioned in parenthesis in the explanation column in Table 36 so that colors are distinguishable even when this document is viewed on black and white media.

template on a MS-Windows -based drawing tool (Visio Corp. 1995), which contains the Riskit symbols and thus supports easy creation and editing of Riskit Analysis Graphs.

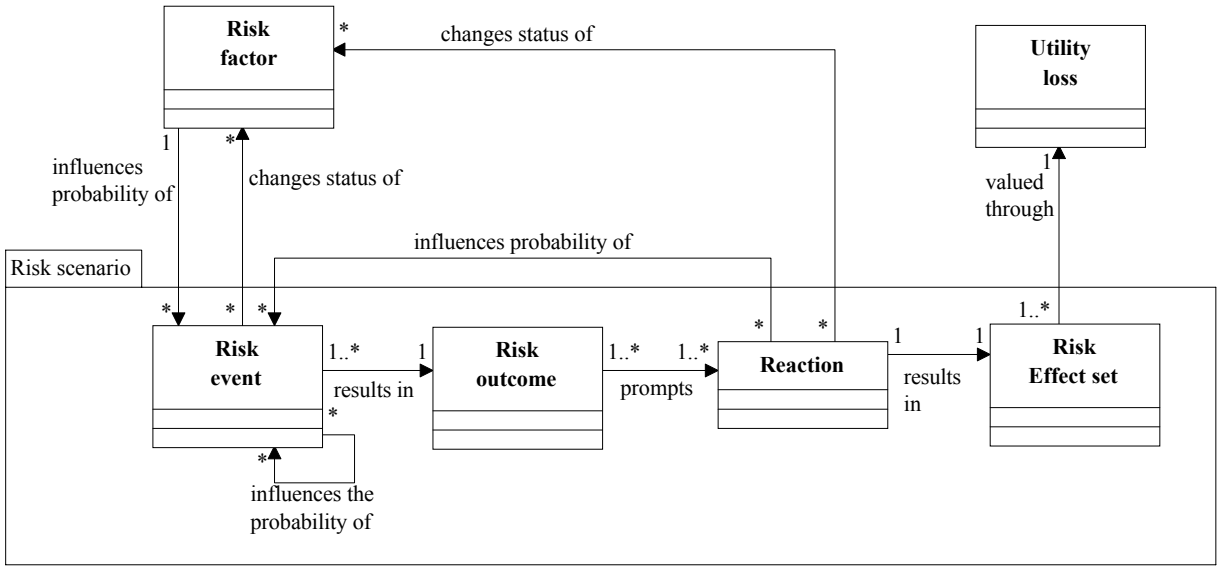


Figure 23: Definition of a risk scenario

Each class of symbols in the Riskit Analysis Graph is drawn in the same vertical column in a graph. In other words, if several risk scenarios are represented in the same graph, all factors are in the same vertical line (column), followed by risk events in the same column, etc. An exception is a case where some risk events only influence risk factors, i.e., they have been created to model probabilistic risk factors. These risk events can be placed towards left of the risk factor column to keep the main part of risk scenario more legible.

The utility loss is estimated for each relevant stakeholder. Thus, each risk effect set has at least one utility loss estimate associated with it.

The previous definitions introduced individual risk elements in the Riskit Analysis Graph: the term is used *risk element* to refer to any of the components presented above, i.e., risk factor, risk event, risk outcome, risk reaction, risk effect and utility loss (pain). We use the term *risk scenario* for any unique event-outcome-reaction-effect combination. Risk scenario is marked in Figure 23 with a dashed rectangle. The key attributes of a risk scenario are its probability, its set of risk effects and, its set of utility losses.

There are several possible ways to use the Riskit Analysis Graph. The *full Riskit Analysis Graph* is based on the underlying conceptual model of risk elements, as presented earlier in this chapter in Figure 22. However, our earlier evaluations with the method indicated that such a complete graph may be laborious to edit and complex to view in practice (Kontio et al. 1996). Therefore, a simpler version of the graph can normally be used. In this normal version of the Riskit Analysis Graph the risk outcome is not explicitly modeled, it is implicitly included in the risk event, as is shown in Figure 24. This is called the *normal Riskit Analysis Graph* and it is the default version of the graph. The consequence of this simplification is that when there is more than one possible outcome for a risk event, these should be modeled as separate events.

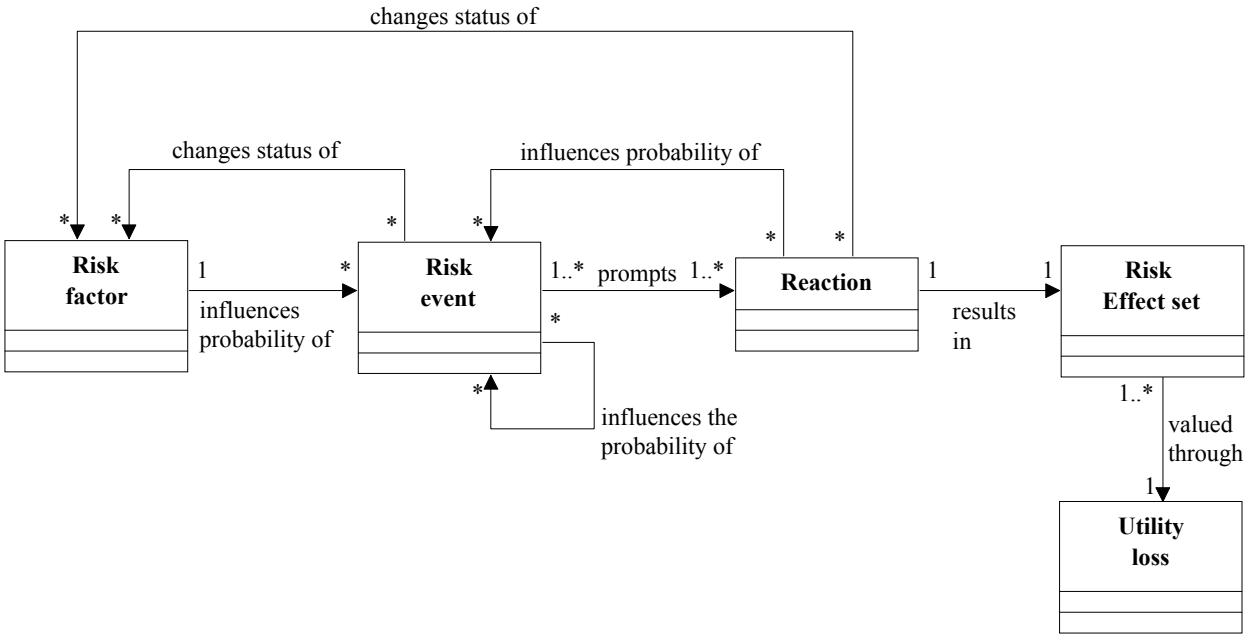


Figure 24: The "normal Riskit Analysis Graph"

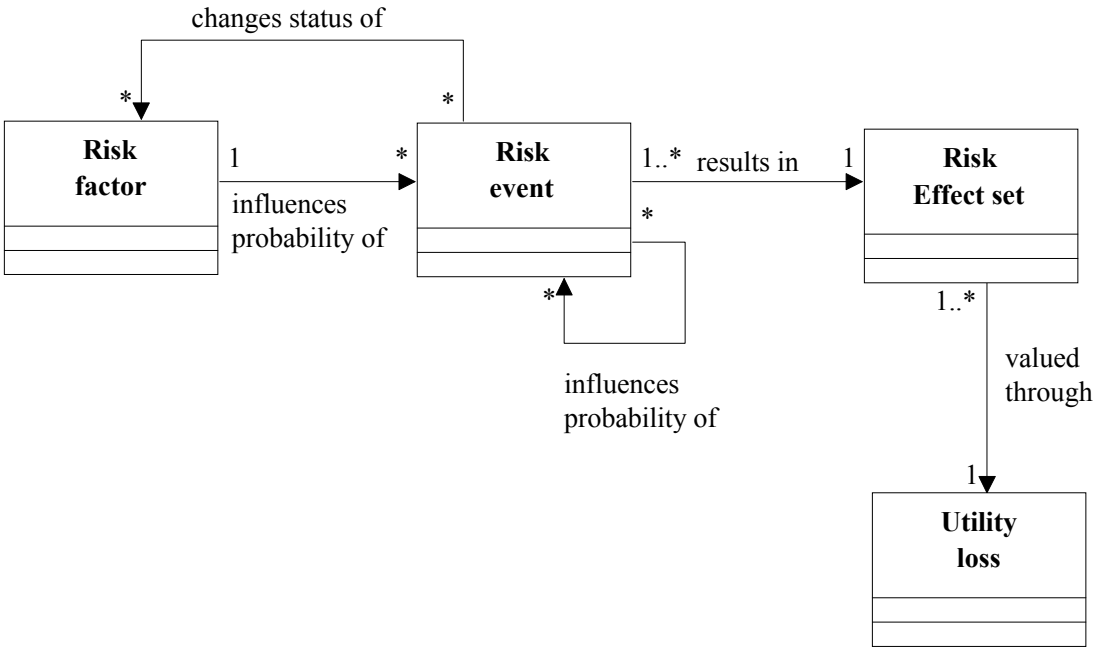


Figure 25: The "simple Riskit Analysis Graph"

We have also defined an even simpler version of the Riskit Analysis Graph, as shown in Figure 25. In this simplified form of the graph, the reaction element is implicitly included in the effect set element. This further simplification of the graph can be used when there is no need to model and analyze different alternative reactions and when there are reasons to minimize graph size and complexity. In a situation where the simple Riskit Analysis Graph is

used and several reactions need to be modeled, they could be modeled as different effect sets. However, it is important to point out that the simple Riskit Analysis Graph makes a potentially central aspect of risk scenarios implicit. Thus, if alternative reactions need to be considered, we recommend that the reactions are explicitly modeled.

We have developed two template sets to support easy, software-based manipulation of Riskit Analysis Graphs. First, we have made the Riskit Analysis Graph symbols available as editable objects in the Microsoft Office environment, primarily aimed for use with PowerPoint presentation software.

We have also created a template for Microsoft VISIO drawing tool (Microsoft 2000). All the basic symbols of the Riskit Analysis Graph are included in the template and the VISIO tool. VISIO supports drag-and-drop editing of items, connectors adjust automatically to changes in object locations, and additional information can be easily entered into diagrams. The VISIO template as well as the PowerPoint template is available for downloading through the Internet[13].

Figure 26: Picture of RiskitFrames and a demonstration of their use

The Riskit Analysis Graph modeling can also be supported by physical teamwork aids called RiskitFrames. The RiskitFrames are laminated Riskit Analysis Graph elements, which can be written upon with whiteboard markers and wiped clean. They have magnetic strips or adhesive sticky tape on the back so that they can be attached and reattached to most surfaces in meeting rooms. RiskitFrames are available in two sizes, A4 and A5.

The RiskitFrames are intended to be used in meetings where risks are discussed and analyzed. The team can synthesize and discuss the risk scenario physically on whiteboard, and the results can be documented either by taking a digital photograph or by using VISIO, PowerPoint or eRiskit software to capture the risk scenario into electronic form. The RiskitFrames have been commercialized into a product.

[13] Download details available at http://www.rdware.com

3.6 Roles and Responsibilities

Each organization can implement risk management in several different ways. In this work, we have chosen to use explicit and systematic risk management and part of systematic risk management is a clear definition of roles and responsibilities. Several authors have listed or outlined roles for risk management personnel (Dorofee et al. 1996; Hall 1998; Michaels 1996; Ritchie & Marshall 1993; Williams et al. 1998; Newland et al. 1997), but none of these sources have identified the need for a risk management process owner. The concept of a process owner had been introduced by the business reengineering field (Harrington 1991; Hammer & Champy 1993; Hammer 1996) in order to allocate authority for cross-functional activities in an organization and to accumulate expertise for specific areas. As our approach is based on continuous, systematic learning, we have adopted the role and responsibility definitions from risk management and process management fields and synthesized roles and their responsibilities for software risk management. These roles are listed in Table 37 and discussed in the following.

Project owner, also sometimes called sponsor, acts as a business owner for the project, pursuing the business benefits for the organization. Project owner controls the resources, prioritizes and resolves business-related issues and conflicts, and supervises the overall progress and success of the project. From the point of view of risk management the project owner acts as the management representative that can support and require risk management to take place in the project. As with many other management practices, risk management will only be effective if higher management understands its importance, supports it, and requires it to happen. In practice, this means that the project owner needs to actively monitor risks in the project, review their status, and see how well the risk controlling actions work to reduce the risks.

Project manager's responsibility is to make risk management happen in the project. This requires that the basic agreement on how, when and by whom risk management should take place, i.e., defining the risk management mandate (see chapter 3.4.1). In addition, responsibilities and actions for risk management need to be organized and delegated, training, support, encouragement, and empowerment given to project personnel. Naturally, the project manager also needs to actively involved in the risk identification, analysis, and control, as well as monitoring the status of risks and reporting about risks to project owner and other stakeholders.

Project participants need to become familiar with the risk management practices and methods and develop a proactive attitude for risk management. They need to seek and communicate potential risks to each other and contribute to the risk management process. Project personnel are usually involved in implementing the risk controlling actions and monitoring the risk status.

Risk management process owner's main responsibility is to provide the risk management process, support it, and collect feedback to improve it. Risk management process owner's responsibilities responsibilities will be discussed in more detail in chapter 4.6.

Risk management facilitator can also be used to support risk management, as shown by several examples (Dorofee et al. 1996; Getto & Landes 1999a). The facilitator can be defined as someone helps risk management activities by providing advice, consulting with methods and templates, proactively proposing the next tasks, ensuring that the results are documented, and tracking status of agreed actions. The facilitator can work for both project manager and/or process owner as his contributions serve both of their interests: the project manager

receives help in applying effective methods for risk management, and the process owner can make the deployment of the risk management process easier and obtain feedback on its use.

Role	Risk management responsibilities	Typical positions taking the role
Project owner	•Require and encourage risk management plans and activities to take place •Monitor risks, their status, and review effectiveness of risk control	•Middle/senior R&D managers
Project manager	•Establish risk management procedures for the project •Ensure that risk management plan exists and is followed •Report project risk status to project owner and other stakeholders	•Manager •Team leader •Senior software engineer
Project participant	•Participate in risk management activities as required •Volunteer information about risks to project's risk manager	•Software engineer
Risk manage-ment process owner[14]	•Define and establish the risk management infrastructure for the organization •Support risk managers •Monitor the status of risk exposure •Collect feedback for improvement	•Quality and process development organization specialist •Finance and control function specialist
Risk manage-ment facilitator	Support projects and risk management by •conducting risk identification and analysis sessions •making the information about methods tools and guidelines available •providing proactive guidance on the risk management process •Ensure that the risk management process is followed •Ensure that risk information is documented and accessible to those that need it	•Internal consultant / trainer •Quality manager •Quality engineer
Risk manager	•Plan and execute the risk management process in the project •Consolidate risk status information and report it to project manager •Support all personnel in risk management activities •Provide feedback to risk management process owner for process improvement	•Quality manager •Quality engineer •Project assistant
Risk owner	•Defines and implements controlling action for the risk he/she owns •Tracks and reports the status of risk and impact of controlling actions	•Software engineer in the project

Table 37: Roles and responsibilities in risk management

[14] Discussed in more detail in chapter 4.6.

Risk manager is a role that is responsible for actually running the risk management process in a project, i.e., conducting sessions, analyzing their results, reporting risk data, and coordinating risk management activities. Risk manager also provides feedback to risk management process owner n the risk management process itself.

Finally, risk owners can be nominated to track, control, and report on specific risks. Such a role gives special emphasis and responsibility for a given risk or group of them and dedicates this responsibility to an individual. This has been found to be an effective way to create sense of responsibility for such risks.

Clearly, not all organizations need all these roles as separate individuals, and some roles might not even be necessary. However, we believe that most of these roles contribute to sound risk management practice and they should be allocated to job positions and individuals.

3.7 The eRiskit Application

This chapter presents the main features, architecture, and the information content of the eRiskit application. A more detailed description of the application is given in Appendix B.

3.7.1 Overview and Main Functionality

The eRiskit application has been developed to support the use of the Riskit method by several project participants, and to capture risk management data and raw experience for improvement purposes. It is an Internet browser-based application that supports users in all steps of the Riskit method by allowing the documentation, analysis, and tracking of risk related information. The software also provides workflow guidance to users, i.e., all main screens have pointers to next steps in the Riskit method. However, experienced users can maneuver in the application without using these process cues as well.

The application has been designed to run on the Intranet and to be accessible by standard Internet browser, such as Netscape or Microsoft Explorer. The user interface metaphor is similar to web pages and users can access the application without installing any special software on their workstation.

The software supports risk information entry both through forms and through graphical editing of Riskit Analysis Graphs. Figure 27 shows an example of the graphical editor. Information entered in textual form can be viewed in graphical form and vise versa. The system also allows saving of incomplete graphs so that they can be completed over several sessions or even by several individuals.

The application allows the tracking of risk status information. Status of risks and their controlling actions are kept up-to-date in risk element and risk controlling action forms and summary reports can be printed or viewed as required.

The software has also been built to support easy language adaptation, i.e., terms shown to user are kept in a single location, allowing cost-efficient translation to local languages.

The eRiskit application also has functionality to support clustering of raw risks, and converting them to risk elements (called “elementization” in the tool). Raw risks can be entered at a meeting or imported to the application. They can be selected and converted to different risk factors or risk events, an action that instantiates the corresponding risk element for further editing. This functionality avoids manual entry of information that is already in the system.

The system supports several user groups and their different access levels in the system. In addition, the underlying database can contain information about several projects, yet each user only is able to see the projects that they have been granted access to.

The risk scenario prioritization is done in separate screens that support ordinal scale rankings of losses and probabilities. Loss rankings can be performed for each stakeholder so that stakeholder priorities for risks can be obtained and analyzed.

Finally, the application has extensive online help that provides context sensitive guidance for most screens.

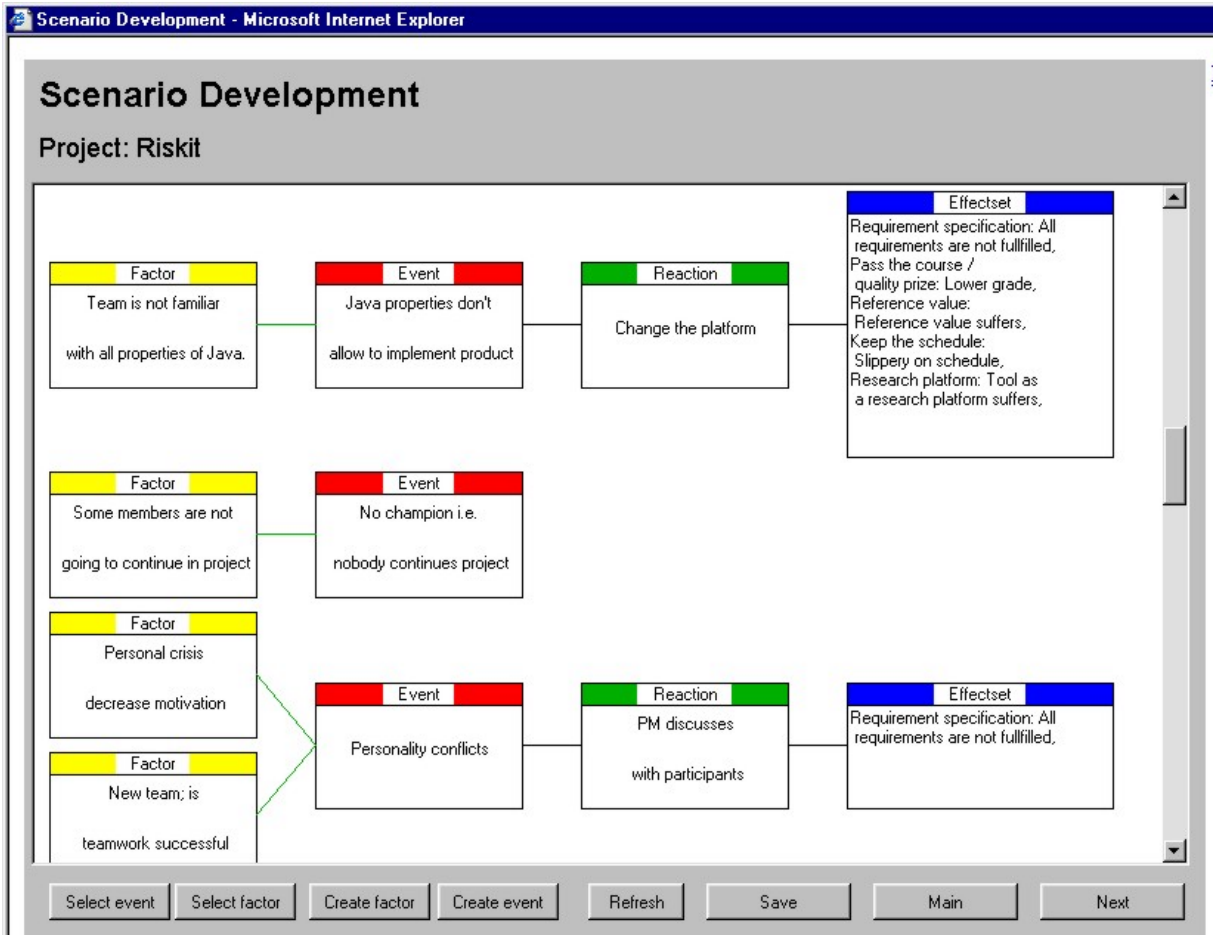


Figure 27: Example screen from eRiskit: graphical risk scenario development

3.7.2 Software Architecture

The applications consist of the client software (applet), server software (servlet) and the database management system (DBMS), and the middleware between them.

The user connects to the server by opening to a specific Internet address with his browser with Java capabilities (Netscape Navigator 4 or Microsoft Explorer 4). Java applet on the web page contacts the servlet on the server, which in turn connects to the database and retrieves the data. Then the information is presented to the user's browser. This describes a mandatory architectural constraint.

The application has been separated from the database product used so that the underlying database can be changed if necessary.

3.7.3 Database Content

We have also identified four main types of risk management empirical data types that can be collected and utilized to support these two main goals: context information, risk management process information, risk element information, and risk monitoring information. We will introduce each of these in the following.

Context information refers to such information that determines the circumstances and setting where the project and its risk management are carried out. Context information is relevant for all software engineering measurement data, but it is particularly important for risk management. The probability of a risk event is often influenced by many factors. By capturing as much as possible of the risk management context information we make it easier to interpret risk management data in the future. From risk management perspective, context information can be further classified into three types. The *organization context information* describes the overall context of the project, that is, what is the application domain, what is the level of personnel experience and training, what methods and tools are used, reporting procedures, organizational structure, etc. The definition of project context information collection procedures is generally relevant to the whole organization, not just to risk management, and thus it is the responsibility of the software measurement program to implement the necessary data collection procedures.

The second subtype of context information is *project information*, which defines the project itself and it includes the definition of the goals, customers, schedule, and constraints of the project. It also includes the definition of the risk management mandate for the project: The risk management mandate is a project-specific statement of the scope of risk management in a project. It defines which stakeholders are to be defended in risk management, stakeholders' priorities, which risks may be excluded from project management's risk management scope and how (e.g., organization management may be willing to take responsibility of some risks without burdening project management with any risk controlling responsibility), and define any other procedures that are not addressed by the risk management infrastructure.

The third context information type is *risk management infrastructure*, which deals with the risk management principles, methods, tools and practices that are available and implemented in the organization. We have adopted Hall's risk management evaluation framework as a tool to document and capture risk management infrastructure for each project (Hall 1995). Hall's framework provides a consistent way to take a snapshot of the risk management infrastructure during a project's duration.

We will discuss how these data are stored in the eRiskit database and discuss the main structures of the database in the following. The high-level database schema is presented in Figure 28.

Project context data is captured in a separate entity to allow the documentation of contextual information about the project, as well as changes in the context. However, the current implementation of context data in the application is simply an informal text field entry.

Stakeholders and goals play an important role in the application. Each stakeholder can be associated with several goals and each goal with several stakeholders. Stakeholders can also be active in several projects.

Each project is associated with a risk management mandate. In fact, risk management mandates can be updated and the old mandates are archived to provide a record of changes for experience capture.

Information about risks is captured in risk elements, according to Riskit method. Controlling actions are associated to one or more risks, i.e., they risk controlling actions and their impacts can be modeled correctly even when an action influences many risks.

The utility loss rankings are associated to each stakeholder and, on the other hand, to risk reaction, which acts as the unique identifier for the effect set of each scenario.

Risk management action entity captures the risk management process enactment data, i.e., what Riskit process activities were performed, how much time they took, who was involved etc.

All key entities in the database have mechanisms to capture changes in their values during their life cycle. This is shown by the state entities in Figure 28, i.e., the system captures the history of changed values for risk factors, risk events, risk effects, loss rank, stakeholder, goal, and project entities.

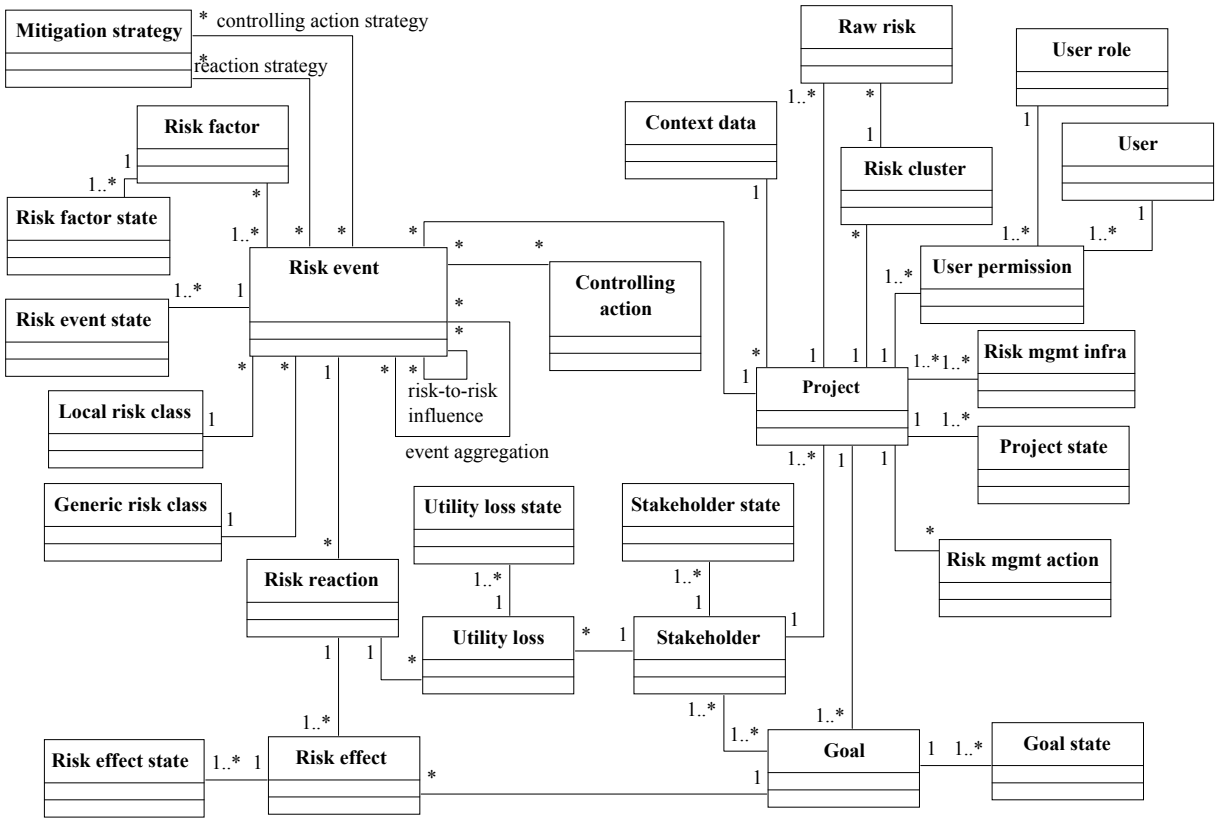


Figure 28: eRiskit database schema

3.8 Conclusions of the Riskit Method and eRiskit Application

The Riskit method was developed to be a theoretically sound, comprehensive, and practical approach for software engineering risk management. It combines contributions from software engineering management, management science, and psychology. The Riskit process is documented in detail, there are several practical tools, and templates that help apply it in practice.

Based on our literature survey, the Riskit method is the first risk management approach that integrates stakeholders and their goals into risk analysis at an operational level, maintaining these links as situations change. The use of stakeholders in risk management in the Riskit method is intended to be normative (Donaldson & Preston 1995), i.e., the stakeholders and their goals should determine how losses, and thus risks, are evaluated in a project.

In addition, Riskit also introduces utility theory and prospect theory, as well as other potential bias controlling approaches, to practical risk management work. None of the reported and published methods reviewed in this work address these issues.

The Riskit method also uses a graphical notation to capture risk definitions formally during the risk analysis process, attempting to combine graphical, intuitive representation with more formal documentation of risks.

The eRiskit application is a prototype implementation of a software tool that provides operational support for applying Riskit, as well as captures risk management information for process improvement purposes. Based on our survey of literature and commercial tools, it is the most comprehensive implementation of a software tool that supports risk management in distributed environment (the Internet), supports the linking of risks to stakeholders and goals, and captures history data on risk management.

As a whole, the Riskit method and the eRiskit application differ from the mainstream risk management approaches in software industry. Based on the analysis of published reports on risk management practices, most organizations seem to use biased risk management approaches without having any explicit means to control them. As a result, the Riskit method has potential to improve the industrial software engineering practice by providing a more sound and consistent risk management method for software projects.

4. Riskit Management Improvement Framework

In this chapter, we present the risk management process improvement framework we have developed. This framework has been built on the concepts and foundations of Basili's Experience Factory (Basili 1989; Basili 1993; McGarry et al. 1994). We will first review the relevant research in risk management process improvement and the Experience Factory in general, and then present the risk management process improvement framework, also coined the Riskit Experience Factory, using the same process representation framework that we introduced and used in chapter 3.

4.1 Review of Relevant Work in Risk Management Experience Capture

There are a limited number reports on work on utilizing data and experience from past projects in software engineering risk management literature. Several organizations have used simple spreadsheets and databases to maintain less formal risk registers and recommended attributes have been proposed in several papers (Hall 1998; Karolak 1996; Newland et al. 1997).

Some aspects of Boehm's work implicitly assumed that data from past projects is available if simulation and cost models are used for estimating risks (Boehm 1989). He also mentioned factors of cost models as possible risk monitoring metrics. Charette has presented an outline of items that should be defined for a project to initiate risk management (Charette 1990). He has also given examples of what should be measured and how this data can be graphed for risk management purposes. However, neither one of these approaches can be considered a systematic way to capture or utilize risk management experience.

SEI has collected data from risk assessments they have carried out during the last few years. Their goal seems to be to support analysis risks and their relationships using lexical analysis on the qualitative descriptions in the database (Monarch et al. 1996). It also seems that frequencies of risks in the database have been used to indicate what are the most common risks. While this research may be potentially useful, we see two major limitations in it. First, the risk database is based on the results of risk assessments. Risks in the database represent risks that some organizations *perceived* as risks using the SEI risk assessment approach. These perceptions may not necessarily be accurate. Participants may have incorrect perceptions of risks, either through lack of knowledge or through a possible bias that the SEI method may have caused. The second concern is that we believe that risks are always sensitive to the situation. Without adequate information on the context where risks were originally recorded, it is difficult to assess whether risks identified in other situations are worth considering in "our" situation.

Hall has defined and implemented a risk database while working at Harris Corporation (Hall 1995). Risks from three projects were collected (Hall 1996) and used for analysis in evaluating Hall's risk management maturity model. Hall has also collected survey data on the levels of risks management practices in various organizations (Hall 1995). Hall's maturity model, presented in Table 38, is a static and staged improvement model, i.e., it contains predefined stages that characterize the maturity of risk management. While the model is

useful as a conceptual framework and it may help organizations structure and plan their risk management infrastructure, we have not found reports or evidence of its validity to support its use as a normative model. Furthermore, the model is not defined in detail to be used consistently for process assessments.

Dimensions		Risk Management Evolution Framework				
		1- Problem	2 - Mitigation	3 - Prevention	4 - Anticipation	5 - Opportunity
Process	Identify	Not seen as positive	Risks are assessed	Risks are volunteered	Risks are sought out	Chances to do better
	Analyze	None	Prioritize risks	Analyze source of risk	Quantitative values used	ROI is calculated
	Plan	None	Action plan is discussed	Action plan is documented	Action plan is executed	Action plan is revised
	Track	None	Monitor critical risks	Monitor all risks	Monitor triggering events	Correct deviations
	Control	None	Discussions increase awareness of what could be improved	Written evaluations document what could be improved	Written evaluations are analyzed and documented as lessons learned	All feedback is tied to improve the process
Infrastructure	Policy	No written standards	Report risks at reviews	Commitment to process	Commitment to metrics	Reward for innovation
	Communi-cate	Lack of communicatio n regarding risks	Risks gathered from lower levels and not communicated to higher levels	Communicate risks within the program team	Between the program team and the customer	Between the program team, customer and the end-user
	Commitment	Upper management	Quality assurance	Management	Employees	Customer and end-user
	Resources	None	Minimal schedule allocation	Minimal schedule and budget are allocated	Sufficient schedule, budget and some resources	Optimal schedule, budget and resources are allocated
	Training	No training	Basic risk concepts	Risk management process	How to quantify risks	How to manage risks
Implementation	Participants	Program manager	Program manager and key technical staff	Program team with a single risk champion	Program team and customer, and a few risk champions	Program team, customer and end-user, with many risk champions
	Procedures	Ad hoc	Verbally stated	Documented	Updated milestones	Living document
	Methods	Ad hoc	Risk surveys	Risk taxonomy	Risk management form	Risk metrics graphs
	Tools	Ad hoc	Top 10 risk list	Risk database	Technical performance measures	Automated risk analysis
	Metrics	None	Defined	Collected	Analyzed	Reported

Table 38: Hall's risk management evolution framework

The MITRE Corporation has developed one of the most comprehensive risk management support systems reported (Garvey et al. 1997). The Risk Assessment and Management Program (RAMP) is a web-based repository of information, examples, guidelines, templates, and links to further information within MITRE's intranet. Its main functionalities include query functionality to find information about projects matching certain criteria, producing

typical risks that these projects have faced and links to individuals that worked in these projects; and downloading lessons-learned reports (called templates within RAMP) about risk areas, or projects. The RAMP is kept up-to-date by automatically prompting content owners to update their data and using the user interactions to augment the database content. The RAMP architecture is based on an underlying database with some static web content. This allows flexible queries on practically all information within RAMP. While much of the information content is updated in a bottom-up fashion, the RAMP support team also conducts analyses and interviews with users to identify new information and insights. MITRE also has a comprehensive risk management process that is integrated with RAMP (Willhite 1998).

The RAMP can be considered a "bottom-up Experience Factory", i.e., most of the experience is entered into the system by individual users, and only occasionally steps are taken to consolidate the information. This approach is cost effective, but it naturally lacks the planned and systematic aspect of experience collection of the Experience Factory.

We also considered several software process improvement frameworks and standards as potential platforms for our process improvement framework. Such frameworks include the SW-CMM (Paulk et al. 1993a), ISO-9001 (ISO 1991b), SPR assessment method (Jones 1994), Bootstrap (Koch 1993) and SPICE (Anon. 1998c). However, for our purposes they are somewhat limited for three reasons. First, risk management is only one of many topics in these models and, consequently, they contain only rudimentary descriptions and requirements for it. Second, these models are static improvement models that describe some required practices to be in place. Given the limited detail on risk management this static improvement paradigm would not result in sufficiently detailed description of the risk management improvement framework. Third, given the limited experience and widespread of biased methods in the industry, it would be difficult to construct a static, normative model for good risk management practices. In addition, the static models have limited empirical validation, especially regarding the risk management section, to support their correctness.

In summary, it seems that software risk management data and knowledge is rarely systematically collected and utilized in the industry. However, the collection and utilization of experience is necessary for improvements in the risk management technology.

4.2 Experience Factory and Quality Improvement Paradigm

As we discussed in chapter 1.3.2, we are using the Experience Factory (EF) and Quality Improvement Paradigm (QIP) as the platform on which the risk management improvement framework is built upon. The EF and QIP provide an established way to build an improvement framework that utilizes experience from on-going projects and, therefore, they are suited to risk management process improvement.

The EF and QIP, however, do have some limitations as well. First, as such they do not provide specific support and guidance on risk management and, therefore, we have adapted and instantiated them for risk management context. Second, learning from in-house experience is a relatively slow improvement approach as it takes time and effort to accumulate experience, analyze it, and then deploy it back to practice. However, given that there are few empirically based contributions in software risk management field, the value of such contributions is likely to be high, and delays are, therefore, acceptable. Third, EF and QIP can be considered relatively costly structures for process improvement: SEL has reported that Experience Factory has cost 11% of software development budget (Basili et al. 1995). We have attempted to keep the overhead specific to risk management as low as possible and

developed automation tools to support labor-intensive aspects of the process in order to reduce the cost of our improvement framework. Fourth, while the EF and QIP have gradually become more widely used in the industry, they are not nearly as commonly used or known as, for instance, the SW-CMM and the SPICE model. As none of these limitations is critical, we have selected the Experience Factory and the Quality Improvement Paradigm as the basis for our improvement framework.

Basili's improvement paradigm consists of two main components, the Experience Factory that describes and organization implementation model for the paradigm and the QIP Cycle that describes the improvement process. They are both presented in what follows.

The Experience Factory developed at the University of Maryland is one of the major paradigms for process improvement (Basili 1985; Basili 1989; Basili et al. 1992b; Basili 1993; Basili & Green 1994). Experience Factory is a way of organizing software development into two distinct organizations, each specializing in its own primary goals: the project organization focusing in delivering the software product and the Experience Factory, focusing in learning from experience and improving the software development process. The main principles of the Experience Factory concept are the following:

- Separation of responsibilities between product development and process improvement.
- Systematic capture and accumulation of knowledge into the Experience Base.
- Continuous learning from own experience through measurement, collection of experience and analysis.
- Systematic reuse of accumulated knowledge through packaging and dissemination of this knowledge.

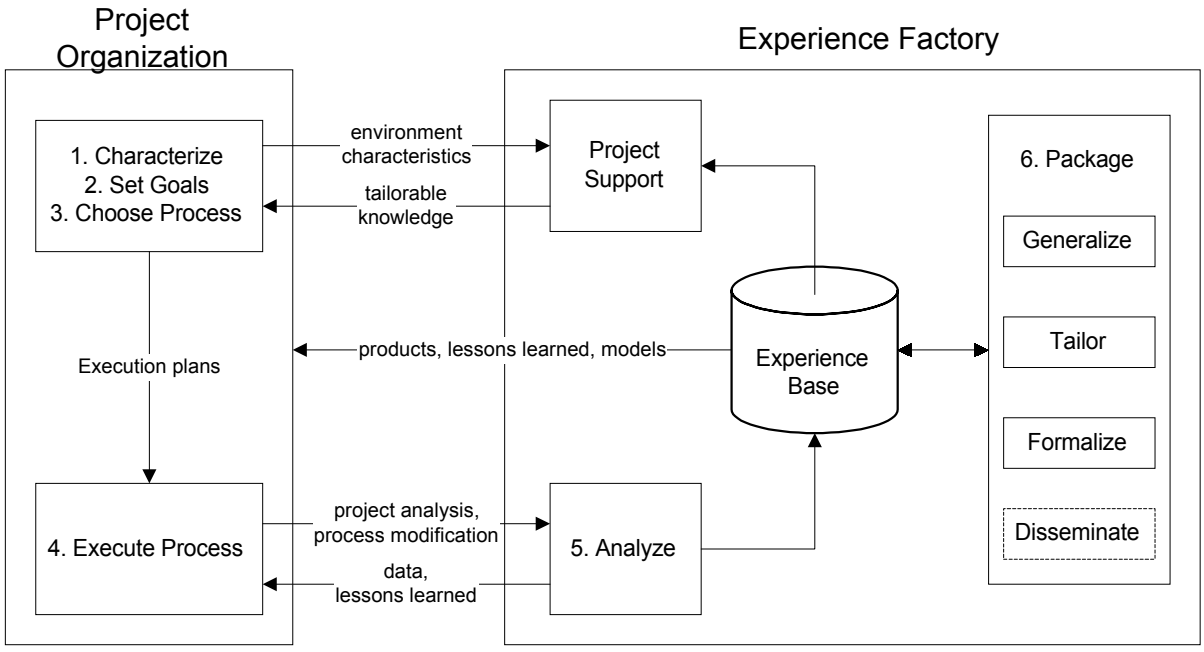


Figure 29: The organization of the Experience Factory

Figure 29 presents an overview of the Experience Factory. The roles of project organization, responsible for product development, and Experience Factory, responsible for

process improvement, are separated. However, they interact to support each other's objectives.

The Experience Factory is based on a quality improvement process that is called the Quality Improvement Paradigm (QIP). Note that in this work we differentiate the concepts of the QIP as a paradigm and the QIP cycle that represents the improvement process. In literature, the term QIP is often used to refer to both QIP cycle and to the paradigm itself.

The QIP Cycle is an approach for systematically defining quantifiable goals for process improvement, selecting appropriate process models for the process and collecting and analyzing the process to package the experiences. QIP Cycle is usually described as containing six steps[15] (Basili & Rombach 1987; Basili et al. 1994b):

1. **Characterize**. Understand the environment based on the available data, models, experience, and insights. Establish baselines with the existing processes in the organization and characterize the criticality of these processes.
2. **Set goals**. Based on the characterization of the environment, set quantifiable goals for the project and organizational performance (and improvement). The reasonable expectations and based on the baselines provided by the characterization step.
3. **Choose process.** Based on the characterization and goals, choose the processes, tools and techniques appropriate for the project, making sure that they are consistent with the goals and constraints set for the project.
4. **Execute**. Perform the process constructing the products and providing data about the progress.
5. **Analyze**. At the end of each specific project, analyze the data and the information gathered to evaluate the current practices, determine problems, record findings, and make recommendations for future projects.
6. **Package**. Consolidate the experience gained in the form of new or updated models, documents and other forms of knowledge and store this knowledge in the experience base, and disseminate information in the organization.

We have presented the steps of the QIP Cycle as processes in Figure 30 using a DFD notation.

The QIP cycle can also be seen as consisting of three different functions: planning, execution, and learning, as shown in Figure 31. Figure 31 also shows the project internal feedback cycle that is used for project control.

The QIP and Experience Factory have been primarily used in the Software Engineering Laboratory (SEL) at the NASA Goddard Space Flight Center, where it has been evolved and improved for 20 years (Basili et al. 1992b). However, it is also being adopted by several other companies. The SEL represents a unique example of systematic data collection and process improvement in the software engineering community. The SEL was the first recipient of the IEEE Computer Society Software Process Achievement Award in 1994 to "recognize its outstanding achievement in software process improvement" (McGarry et al. 1994).

[15] The original version of the QIP (Basili & Rombach 1987) gave a project manager's view on defining a project. A revision to the QIP (Basili et al. 1994b) added the sixth step ("package") and shifted the focus to emphasize an experimenter's view.

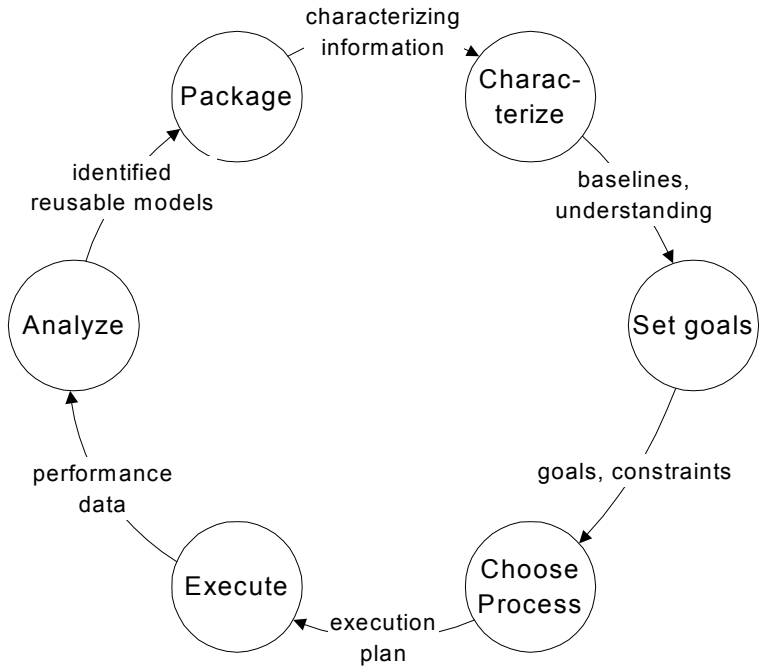


Figure 30: The Quality Improvement Paradigm expressed in data flow diagram

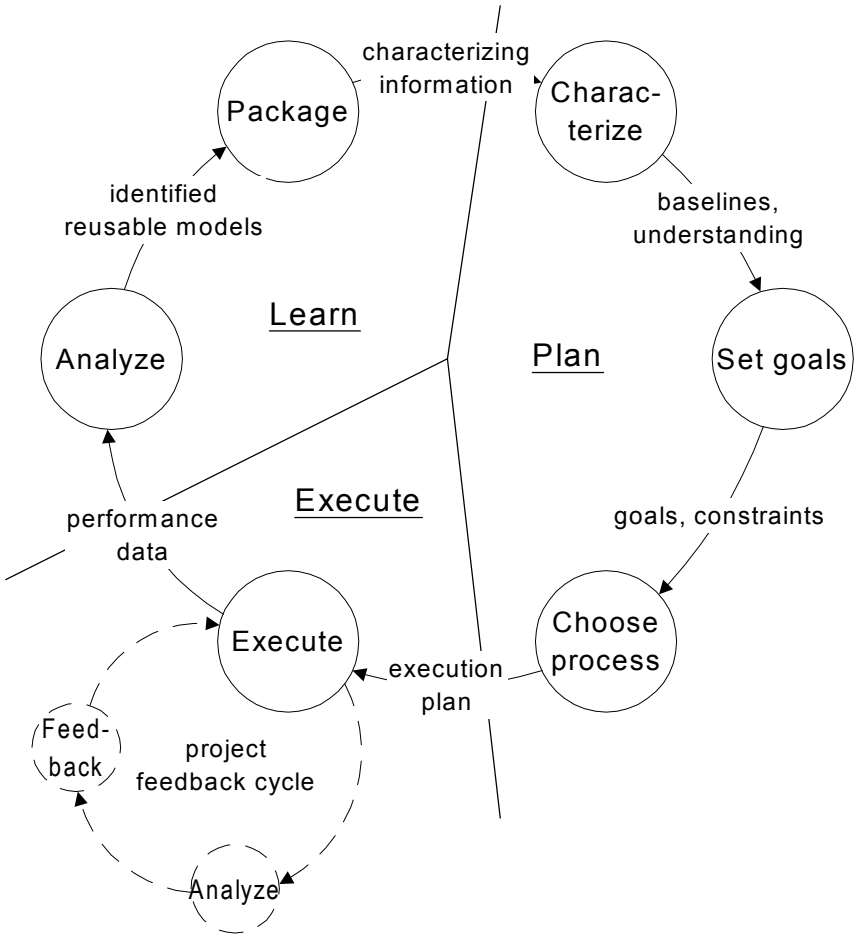


Figure 31: The planning, execution and learning phases in the QIP cycle

We have grouped the "paradigm attribute definitions" into three groups: assumptions, methods, and research methods. We have numbered them sequentially for easy reference. The QIP paradigm definitions are based on previous work that has alluded to such principles (Basili & Rombach 1988; Basili 1989; Basili & Rombach 1991) and on our own interpretation and analysis of QIP. In the following, we have referred to the source where we have derived the paradigm principle. In the case of Basili and Rombach 1988 paper, we have also included an explicit reference to the principle mentioned in their paper (Basili & Rombach 1988).

We have identified the following assumptions as ones that capture essential aspects of the QIP and EF:

1. Continuous learning is essential for all evolutionary fields, such as software development.
2. Continuous, sustained improvement is not possible without understanding of the current situation and environment.
3. Measurement and modeling are essential for understanding and learning ((Basili & Rombach 1988): M1, M12).
4. All knowledge is potentially reusable and, therefore, should be explicitly represented ((Basili & Rombach 1991); (Basili & Rombach 1988): P9).
5. Improvement and organizational goals must be explicitly stated and measured ((Basili & Rombach 1988): M3, M4).
6. All software development knowledge must be localized. We do not yet have universal models for software quality or productivity but if and when such universal models are identified, they will need to be localized as well ((Basili & Rombach 1991), (Basili & Rombach 1988): P7, P9). The following are refinements of this principle:
   - 6.1. Knowledge is reusable within the same domain it was initially formulated. If it is reused in other domains or situations, the success of this reuse is strongly dependent on the understanding of the similarities and differences between the situations ((Basili & Rombach 1988): M13).
   - 6.2. An organization must build up its own understanding of its products and processes, based on measurement, modeling, and analysis (Basili & Rombach 1991).
   - 6.3. The measurement and modeling objectives vary and actual metrics and models are dependent on these objectives ((Basili & Rombach 1988): M3, M8, M9).
   - 6.4. Improvement objectives are specific to each organization and, among other things, depend on business goals and strategies, competitive situation, organization's current strengths and weaknesses, customer needs and preferences, and the technologies available ((Basili & Rombach 1991); (Basili & Rombach 1988): P8).
   - 6.5. The type and characteristics of the software process depend on the organizational and improvement objectives ((Basili & Rombach 1988): P6).

We have identified the following methods as ones that are used within the QIP paradigm.

7. The QIP Cycle represents an effective method for creating localized knowledge for software development. The main principles of the QIP Cycle are the need to understand current situation, need to formulate goals, choose implementation plans based on experience, continuous measurement during execution, explicit

analysis and validation of experiences, and packaging of knowledge ((Basili & Rombach 1988): P4, P5).

8. The Experience Factory represents an effective model for implementing a quality improvement system that aims at creating localized knowledge. The main principle of the EF is the need to specialize in the analysis and packaging of knowledge ((Basili & Rombach 1988): P1, P3).
9. The Experience Base, as a part of the Experience Factory, represents an effective way to document, accumulate, and distribute localized knowledge.
10. The GQM method (Basili 1992; Basili et al. 1994a) represents an effective method for defining metrics that are goal and situation dependent.

Research Methods

11. Software engineering research must be empirical, i.e., theories must be validated by observations, experiments, surveys, and data collection. (Basili & Rombach 1991).
12. Good experimental design improves the confidence and usability of results from experiments.
13. Both qualitative and quantitative techniques will need to be used in software engineering research ((Basili & Rombach 1988): M5).

The improvement paradigm statements above characterize the common principles and assumptions for EF and QIP as we have been able to interpret them from the literature and from our experience in implementing them. Each implementation of EF and QIP, however, may have additional or different assumptions and principles, depending on the context and objectives of the situation.

4.3 The Risk Management Improvement Paradigm

In addition to the general paradigm definition statements presented for the QIP, we have adapted corresponding statements to our risk management process improvement framework. These principles are based on the suppositions we presented at the end of chapter 1.1, conclusions we presented in chapter 2.5, and the requirements we have presented for the risk management method and risk management improvement framework in chapters 1.3.1 and 1.3.2. They are presented in the following.

Risks are influenced by a variety of contextual factors. These contextual factors must be identified and understood in order to understand and improve risk understanding and risk management process. It follows that situation specific experience and knowledge should be collected and analyzed to improve risk management practice, instead of solely relying on outside sources for risk management capability development.

Explicit documentation of risks is essential for communication and understanding about risks, as well as for learning from experience. If risks are well documented and this documentation is captured and archived, this raw experience can be used to improve the knowledge and methods in risk management.

It would be desirable to be able to observe and record the impact of risk management actions accurately, so that it could be used to assess what approaches are effective and what are not. Unfortunately, in practice this is impossible due several constraints, as will be discussed in chapter 5. For instance, risk management actions are based on subjective information about risks and it is difficult or impossible to assess whether these subjective

estimates have been accurate (constraint C-2 in chapter 5); several other factors influence the success of a project and it is difficult or impossible to isolate the impact of risk management actions (C-3); and each project and its risk portfolio are unique (C-4).

Nevertheless, even though the accurate measurement of risk management impact is difficult, it needs to be measured as well as possible in order to obtain feedback on the method. Due to difficulties in this measurement, triangulation and qualitative data and analysis should be used to augment experience that is collected.

4.4 Risk Management Improvement Process Purpose and Scope

In the risk management context, the purpose of risk management improvement process is to (i) improve knowledge about potential risks to support risk analysis, and (ii) improve the risk management process itself. We will discuss these in the following.

First, the goal of improving knowledge about potential risks to support risk analysis is stated in Table 39, using the GQM approach[16]. By improving our knowledge about potential risks, we aim at more accurate estimations of frequency of occurrence and losses inflicted in future projects. Even though some risks are likely to be unique to each project, some risks are more common in a given domain and organizations and they are more likely to occur in future projects as well. Recognizing such risks will help organizations prepare for them and perform their projects better.

Analyze	Risks that threaten the projects,
In order to	*Describe* and *understand*,
With respect to	Type, Frequency, Associated controlling actions, Situational factors
From perspective of	Risk management process owner and Executive management.
in the context of	Software development projects and programs.
because	Some risks are likely to be common in a given domain and organization, recognizing such common risks will help prepare for then better.

Table 39: GQM goal for improving knowledge about risks

Second, the goal of improving the risk management process itself is based on both internal and external information. The external influence refers to monitoring advances in risk management technology and practices; as well as on requirements presented by standards, competitive situation, customers, and partners. These define both what is technically feasible and how important risk management is from a business perspective. The internal perspective tries to describe and understand current practices: what works well and where there are problems; assess the current process against available benchmarks; and appraise new techniques and approaches to see whether they work in a given context. For instance, one

[16] Note that syntaxt and meaning of the verbs in the purpose field ("in order to") were presented in section 1.6 (Table 4).

could seek to find out what methods and approaches were effective in risk identification and analysis, what controlling actions were implemented, and how effective they were in reducing risks. Table 40 presents the GQM goal for risk management process improvement.

Analyze	Risk management process,
In order to	*Describe* and *understand* best practices, *Assess* current process against best-of-breed in industry and state-of-art, and *Appraise* new approaches in this context.
With respect to	Overhead and ease of use, Effectiveness
From perspective of	Risk management process owner and Executive management.
in the context of	Software development projects and programs.
because	Some risks are likely to be common in a given domain and organization, recognizing such common risks will help prepare for then better.

Table 40: GQM goal for improving the risk management process

Both of the goals presented in Table 39 and Table 40 are high-level, generic goal statements. Any specific improvement action or study will need to refine these objectives to a more concrete level and the chapter A.2 provides some guidelines for this.

4.5 Risk Management Experience Base

A central element in the risk management improvement process is the risk management Experience Base. In this work, we have outlined the content and structure of such Experience Base, assuming that it is a part of a larger Experience Base that serves the process improvement purposes in general. The content of the Experience Base is presented in Figure 32 and we will discuss the content and structure of each component in the following.

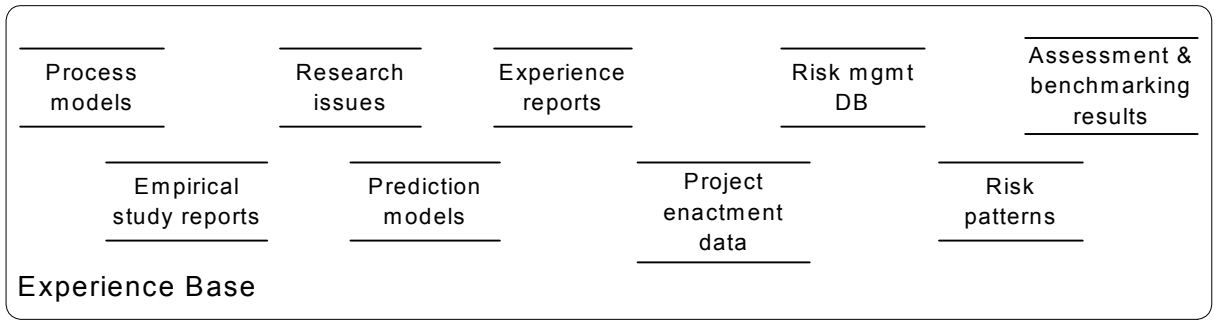


Figure 32: Content of the risk management experience base

The Experience Base contents have been broadly discussed in various publications. Basili categorized EB contents by the representation formalisms: equations, histograms, lessons learned reports, and models or algorithms (Basili et al. 1994b). Oivo and Basili have also proposed a detailed, object oriented representation schema for some software engineering models (Oivo & Basili 1992). The type of knowledge in an Experience Base can be

categorized by several different dimensions, including information content and semantics, type of knowledge, representation formalism used, and storage media. Furthermore, some important attributes could be associated with each type of knowledge, such as level of confidence in the knowledge, accuracy, and precision. In this work we will discuss the main content of different experience repositories in the Experience Base and refrain from discussing other details, except for the risk management database, which will be discussed in detail in chapter 4.7.

We have implemented such a prototype of such an Experience Base and populated it with data and knowledge from our empirical studies and findings. The Experience Base is not open for public but we are using it as a development environment to improve the improvement framework and the Experience Base further.

4.5.1 Process Models

Process models are explicit representations of the activities, information flows, artifacts, agents, and resources taking place in an organization. As models in general, process models are abstractions that contain the necessary knowledge and guidelines to enact a process. Process models can be documented in manuals, in web pages, or process modeling environments, each using different combinations of graphics and textual information. The current practice in process modeling is shifting from text-based manuals to hyperlinked, web-based process models, sometimes partially generated automatically by a process modeling tool (Kontio 1995b; QPR 2001).

There are several methods available for constructing process models (Culver-Lozo & Gelman 1993; Curtis et al. 1992; Dutton 1993; Frailey et al. 1991; Huckvale & Ould 1993; Humphrey & Kellner 1989; Kellner 1996; Kontio 1998; Madhavji et al. 1994; Ould 1992; Seaman & Basili 1994), and several representational schemata have been proposed for process models, both formal, i.e., suitable for computer execution or analysis (Abdel-Hamid & Madnick 1991; Alho et al. 1996; Bandinelli et al. 1991; Barghouti 1992; Catron & Ray 1991; Chen & Tu 1994; Conradi et al. 1991; Crowley & Silverthorn 1991; Derniame & Gruhn 1994; Dowson 1987; Fernström 1993; Kaiser et al. 1990; Lehman 1986; Minkowitz 1993; Oivo & Basili 1992; Rombach 1989; Taylor et al. 1993; Warboys 1989), and informal, intended to be used by people in modeling the process (Brandl 1991; Christie 1993; Frailey et al. 1991; Huckvale & Ould 1993; Kaltio 2001; Lai 1991; Ould 1995; Radice et al. 1985; Singh & Rein 1992). As the selection of such notation is sensitive to each situation and the topic is beyond the scope of this work, we refer to some papers that specifically address process modeling notation selection for guidance on this issue (Armenise et al. 1993; Armitage et al. 1995; Christie 1994; Finkelstein et al. 1994; Kellner 1989; Kellner & Rombach 1991; Kontio 1994b).

4.5.2 Empirical Study Reports

Empirical study reports document the empirical study objectives, arrangements, the data and information produced, analysis methods used, and the conclusions drawn. The empirical study reports can be published articles or internal reports. Examples of such reports on the Riskit method are the various refereed papers (Freimut et al. 2001; Getto & Landes 1999a; Getto & Landes 1999b; Kontio & Basili 1996; Kontio & Basili 1997; Kontio et al. 1998) and internal reports (Englund 1997; Kontio et al. 1996) referenced in this report. It is also important to include sufficient documentation about the context of the study so that other

people accessing the report can assess how applicable the findings might be in different domains and situation.

Empirical study reports are useful for two purposes. First, they provide a pool of knowledge that people can utilize when they are considering alternatives or solving problems in their projects. Second, they can act as samples and templates for new empirical studies that either replicate the study or perform similar, related studies.

4.5.3 Research Issues

We are proposing the research issues are stored as a separate item in the Experience Base. As the QIP Cycle is frequently executed in the organization, its each instance provides an opportunity to obtain empirical data and to test models or hypotheses. Projects are often faced with several practical constraints and it is not always feasible to make arrangements for the most urgent and obvious research questions. In such situations the research issues list can support the identification of alternative, relevant, but perhaps less interfering or less costly research issues to be included. The Research issues list can also be reviewed frequently so that it represents the organization's prioritized view of most relevant research questions.

We are proposing the following outline for the research issues list:

- Background and motivation
- Articulation of research question(s)
  GQM statement of the objective
- Proposed empirical study arrangements
- References to other relevant work
- Keywords describing the research issue

As the research issue list grows large, it will be useful to define taxonomy of research topics that will allow structured browsing of the list.

4.5.4 Prediction Models

Prediction models are methods, tools, or algorithms that can be used to estimate cost, effort, duration, or size of software development. Several estimation models have been presented and used in industry (Albrecht 1979), and(Boehm et al. 1995; Fenton 1991; Kemerer 1993; Zuse 1997) they are supported by software tools that make the calibration, data entry and use of the estimation tools easy.

Common requirement to all estimation approaches is that estimation models need to be calibrated to the organization and context they are used. This requires measuring past projects' performance and adjusting the parameters of the models based on them until models can be used. Calibration should be done frequently in the future as well to adjust to changing environment and to improve accuracy of these prediction models.

4.5.5 Experience Reports

Experience reports are lessons-learned accounts and summaries written by project participants. They differ from empirical study reports in several aspects: (i) they do not need to have an empirical study design and specific data collection mechanisms included, (ii) they are not written or synthesized by analysts external to the project, and (iii) data, information and analysis is not necessarily based on scientific principles or statistical methods. While

these constraints limit the internal and external validity of the findings, experience reports can still act as a valuable source of information and help discover good practices and potential problem areas.

Experience reports should be written during the project, or shortly after it has ended. It is beneficial if more than one person from the project participates in the writing of the report so that broader range of experiences can be included. Use of teamwork or brainstorming techniques can help elicit participants' experiences effectively (Brassard & Ritter 1994; Scholtes et al. 1996). The lessons-learned report should be a normal closing procedure for the projects so that such reports are created and entered into the Experience Base regularly.

Providing a template for the experience report content will make the writing of the report easier, improve the quality and coverage of the reports, and make the report contents more consistent. Thus, such a template should be provided to all projects, a simplified example is provided in Table 41.

1. Introduction
2. Project Description
   2.1. Project Context and Background
   2.2. Project Objectives
   2.3. Project Organization and Resources
   2.4. Key Processes, Methods, and Tools Used
3. Project Performance and Results
   3.1. Project Output and Deliverables
   3.2. Project Enactment Data
4. Problems and Root-cause Analysis
5. Good Practices Identified
6. Proposed Research Issues
7. References to Project Documentation and Data

Table 41: Example outline for a lessons-learned report

4.5.6 Project Enactment Data

Project enactment data refers to measurement data and information that is captured and stored during the project. This data can include measurements about the software products and other artifacts produced in the process, measurements about the process, or measurements about other information entities involved in the process. The process elements introduced in Table 72 can act as a checklist for identifying potential objects of study for project enactment data.

The definition of project enactment data should be based on a measurement program and the process and issues involved in establishing such a program have been widely reported in the literature (Basili & Green 1994; Clapp 1993; Daskalantonakis 1992; Grady 1992; Hall & Fenton 1997; Offen & Jeffery 1997; Pfleeger & McGowan 1990; Pfleeger 1993; Pfleeger 1995) and the GQM method can be used as a method for defining goal-driven metrics in such a program (Basili 1992; Basili et al. 1994a; Rombach 1991; van Solingen & Berghout 1999).

While measurement goals tend to be situation specific, some generic goal classes can be identified. We have synthesized the following generic measurement categories (Grady 1992; Zuse 1997):

- Track project progress.
- Build up a basis for project estimation.
- Track and analyze defects.
- Track project costs.
- Provide information for cost accounting and invoicing.
- Provide information for process improvement, i.e., accumulate experience.
- Identify complex and error prone modules.
- Validate the benefits and disadvantages of methods, tools, and practices.

The actual methods, approaches, and tools for establishing measurement programs for project enactment data are beyond the scope of this work. We refer to the literature cited and assume the existence and availability of such data in this work. From the point of view of this work, the establishment of project enactment measurement system and database is beyond the scope of this work. However, many, if not most, software development organizations have measurement systems in use and such system and their data can be used to improve risk management knowledge.

4.5.7 Risk Management Database

Risk management database contains information about potential risks, occurred risks, planned and implemented controlling actions, and about the risk management activities performed in projects. In this work we have defined such a database in detail in the form of eRiskit application and will discuss the content of it in chapter 4.7.

4.5.8 Risk Patterns

Risk patterns are risks and related risk controlling actions that have been found to be common in a given domain. In this context, risks include elements of risk scenarios, i.e., Riskit Analysis Graph elements: risk factors, risk events, outcomes, reactions and effect sets. Associated to them there may be risk controlling actions.

In simplest form, risk patterns can be localized checklists, checklists that list risks that are common in the given domain and context. Such localization can be based on published risk checklists[17] and adapted based on personnel and project experience. More advanced forms of risk patterns include combinations of Riskit Analysis Graph risk scenarios, and association of potential risk controlling actions to such scenarios.

4.5.9 Assessment and Benchmarking Results

Assessment and benchmarking results contain results of process assessments or benchmarking efforts. Process assessments can include formal assessments to obtain a quality certificate, such as ISO 9001 (ISO 1987; ISO 1991b), process assessment frameworks, such as the Capability Maturity Model (Paulk et al. 1993a; Paulk et al. 1993b), SPICE (ISO 1998c; ISO 1998b), or Bootstrap (Haase et al. 1994; Koch 1993; Kuvaja et al. 1993), or internal audits and assessments. Such assessments usually result in a profile or score that

[17] Such checklists were references in section 3.4.3 (Barki et al. 1993; Boehm 1989; Carr et al. 1993; Honkonen 1999; Laitinen et al. 1993; Moynihan 1997; Ropponen 1993)

describes the current capability of the assessed unit, and list recommendations for improvement.

Benchmarking is an activity where an organization compares its own processes or methods with another organization in order to identify and share good practices and find solutions to problems. Benchmarking is a common practice in industry and there are several guidelines available for conducting benchmarking projects (Bean & Gros 1992; Bendell et al. 1997; Spendolini 1992). The form and output of benchmarking exercises varies quite a lot, depending on the objectives and scope, available resources, and the conventions used by the participating organizations. Therefore, the format of the benchmarking reports varies in the experience base. However, it is recommended that a benchmarking report template is developed to support higher consistency and comparability between benchmarking results.

Assessments and benchmarking are important additional measurement data that help compare organization's performance to other organizations in the industry. They provide both quantitative benchmarks, such as comparable measurement data between benchmarking partners and process maturity model score, as well as qualitative information about the strengths and weaknesses of the assessed processes. This information can be used to guide improvement efforts and empirical studies.

4.6 Roles and Responsibilities

We have adopted the Experience Factory concept into the Riskit method, using the main principles and structures as presented in the previous chapter. This Experience Factory adaptation is called Riskit Experience[18] Factory. The overall structure of the Riskit Experience Factory is presented in Figure 33. In the following, we will present an idealized description of the components and their interaction in the Riskit Experience Factory.

As shown in Figure 33, the project organization is responsible for the planning phase of the project, i.e., characterization, goal setting, and choosing the process for the project. The Experience Factory organization supports this phase by providing relevant baselines, process facilitation, risk identification support, and risk analysis support to the project. This is based on the context information and description of the project that is exchanged between the project organization and the Experience Factory. The project organization can also independently exploit the risk management experience base to obtain the risk management process definition, risk identification checklists, lessons learned reports and other guidelines that may be available.

As the project is enacted, the project enactment data is captured and stored into the experience base, as well as the data about the risks. The project organization also reports problems, relevant observations, and improvement suggestions to the Experience Factory for further analysis or resolution. The Experience Factory provides baselines, analyzed data and suggestions and relevant lessons learned reports to the project organization in order to support the project's risk management activities.

[18] Even though this is slightly in conflict with the terminology we introduced in chapter 4.2, we are using the term "experience" instead of "knowledge" in this context for two reasons. First, the the Riskit improvement Factory produces raw experience (experience (i)) and it is synthesized into empirical knowledge (experience (ii)), thus the ambiguous interpretation of the word "experience" is actually appropriate. Second, as the concept is build upon Basili's Experience Factory, we wanted to be compatible in naming the framework.

As the project progresses or is concluded, the analysis step is performed by the Experience Factory, often involving project organization as well. Relevant experiences and data are analyzed and documented into the experience base.

Finally, in the package step the Experience Factory generalizes these experiences from the context of a single project, tailors them to general context of the organization, formalizes and documents the knowledge, and disseminates it to the organization.

Within the Riskit Experience Factory the task of risk management process improvement can be identified as a separate set of responsibilities. Using the terms of the Experience Factory, the people with this responsibility could be called Experience Factory risk management analysts. Alternatively, we can use the concept of risk management process owner to refer to the responsibilities of such an analyst. In this chapter, we provide a more precise definition of risk management process owner responsibilities, after reviewing literature definitions of process ownership.

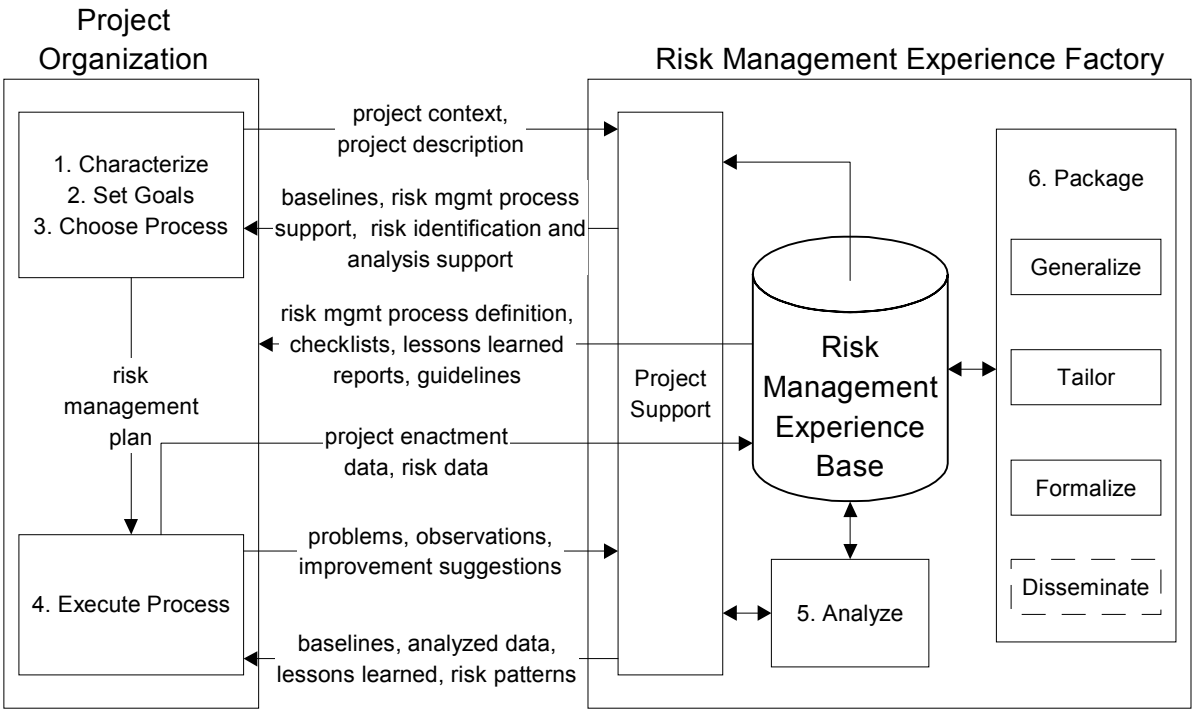


Figure 33: Riskit Experience Factory

The concept of process owner has been introduced by the business process reengineering community and it is currently used broadly in industry. There are several different ways to define process owner responsibilities (Harrington 1991; Hammer & Champy 1993). A more recent definition by Hammer describes process owner's responsibilities as providing the means to perform the process through *design:* deciding what is done in a process, documenting it and training the process performers; *coaching*: helping process performers in difficulties, coordinating process performance work when problems arise; and *advocacy*: soliciting the needed resources for performers, tools, facilities etc. (Hammer 1996). We have adapted the risk management process owner responsibilities as follows:

- Model and document the risk management process
- Plan and manage risk management process improvement projects

- Organize process development within the process: experience capture, analysis and packaging
- Ensure the effectiveness of the risk management process
- Support risk managers in projects
- Plan and define the training for risk management
- Plan, propose and maintain the tools and methods for risk management
- Monitor and survey advances in risk management technology

We have presented the specific responsibilities of a risk manager and risk management process owner in each Riskit step in Table 42.

Riskit Step	Risk manager responsibilities	Risk management process owner responsibilities
Risk management mandate definition	Characterize project w.r.t. risks Identify stakeholders Define the risk management plan	Provide examples and consultation to risk manager Provide risk management training
Goal review	Define goals	Plan and facilitate goal review sessions
Risk identification	Schedule risk identification sessions	Provide checklists Plan and facilitate risk identification sessions
Risk analysis	Schedule risk analysis sessions	Plan and facilitate risk analysis sessions
Risk control planning	Schedule risk controlling action planning sessions	Plan and facilitate risk controlling action planning sessions
Risk control	Implement risk controlling actions and track their status	Coaching and support, if needed
Risk monitoring	Monitor risks and status in the project	Monitor overall risk level

Table 42: Risk manager and risk management process owner responsibilities in the Riskit process

4.7 Summary of the Risk Management Improvement Framework

The framework presented in this chapter is an instantiation and adaptation of the Experience Factory and Quality Improvement Paradigm for risk management process improvement purposes. Based on our literature survey, this model is the most comprehensive risk management improvement framework presented in the literature.

In our work in the development and improvement of the Riskit method, we have used and applied the Riskit Experience Factory in practice and used and enacted all process steps and Experience Base content. However, this practical use of the method has serious limitations that prevent us from using this experience as an empirical case in this work. First, we have used the framework in conducting risk management research in many organizations and, therefore, there is no formal and physical instance of such Experience Factory so that all data and information could be shared between parties that have been involved. Second, we have not used empirical study designs, case study arrangements, nor collected specific data for evaluating the framework. Thus, the framework has not been scientifically evaluated and our findings about it are subjective ones. Given these limitations it is obvious that additional,

independent evaluation of the proposed framework should be conducted to assess its correctness, feasibility, and value, as well as to improve the framework.

While we have developed the risk management process improvement framework as a complement to the Riskit method, we propose that the framework can be used as an improvement model for other risk management methods as well. The basic steps, Experience Base contents, and roles defined for participants are likely to applicable to any risk management approach. However, the Riskit method has several distinct characteristics that make it a particularly useful method for leveraging the improvement framework. First, the conceptual data model of the Riskit method, in the form of Riskit Analysis Graph, captures comprehensive information about risks during the risk management cycle. This information can be used in the Experience Factory analysis step to gain deeper and more solid insight into risks in a given project context. Second, the Riskit process has been integrated and linked to this improvement process explicitly, making it easier to integrate and use the two processes. Third, the eRiskit application is a concrete tool that automates and extends the capture and archiving of raw experience during the risk management process, significantly increasing the potential for experience based learning and improvement.

Given that risks, as well as risk management processes, need to be adapted and localized to have best impact in each organization, we believe that improvement frameworks are needed to bring forth advances and improvements in risk management practice. Yet, there are very few reports of risk management improvement frameworks in use. It is therefore important to motivate organizations not only to introduce risk management into their software organizations but also to highlight the importance of continuous and systematic improvement in risk management. Research community should pursue ways to make risk management and risk management process improvement as cost effective as possible so that practitioners will be able to apply them under the constraints of real projects.

5. Empirical Evaluation

In this chapter, we describe the challenges of conducting empirical studies in software risk management, justify the research approaches we have selected, and present the empirical studies and their results.

Empirical work in software engineering is particularly challenging for various reasons: resource limitations often force empirical studies to be conducted on limited scope or artificial environments (Curtis 1980); software projects are often large and under time pressure, limiting the number of data points and empirical arrangements to be used (Glass 1995b); in industrial context it is difficult to control sufficient number of parameters (Tichy 1998); technologies change frequently in software engineering, potentially making research results obsolete by the time sufficient scientific evidence has been collected (Tichy 1998); constructs in software engineering are often large and complex, making their evaluation and control in experiments difficult (Fitzgerald 1991); and the software engineering research community lacks a history of empirical research (Glass 1995b; Tichy 1998). Despite of these difficulties, attempts to increase the frequency of empirical studies will improve the validity and generalizeability of research results.

When designing our empirical studies, we were concerned with some fundamental difficulties that are particularly relevant for performing empirical studies in risk management. More specifically, we identified the following challenges (Kontio & Basili 1997):

C-1 The correct phrasing of a risk is subjective and depends on the situation and participants. A person may mentally cluster a group of related risks into one risk, whereas another person may address them individually. This makes the comparison of risk data difficult at best and meaningless at worst.

C-2 The real values for probability and loss are not known, or even knowable. Estimating risk requires predicting the probabilities and impacts of future events. As situations inevitably change, even historical data on past risks cannot give correct estimates for risks. This makes it difficult to evaluate the impact of risk management methods as we do not have access to "real risk data". This constraint requires us to use indirect measures to evaluate risk management methods.

C-3 Each set of events occurring in a project is unique and not repeatable. Risks are sensitive to the characteristics of the project and its environment in time. In practice, it is impossible to identify or control all factors that influence the risk portfolio of a project. This makes the comparison of different empirical studies and data difficult as specific characteristics of a situation cannot be factored out. This constraint motivates us to use single case studies in our empirical studies.

C-4 Risk management method cannot be separated from the object of study: if a method results in some action, the state of the system irrevocably changes. Only one scenario of risks and events is available from any system. This constraint

leads us to measure the results of a risk management method as snapshots of reality, i.e., comparisons over time may decrease the validity of results.

C-5 *Risks are probabilistic phenomena. A single occurrence of a risk, whether predicted or not, cannot be used to draw any conclusions about the accuracy of our risk analysis methods. We should have a large number of data points to counter the probabilistic effect. As this is often unrealistic (constraints C-3, C-4 and C-7), we should utilize the limited number of studies more effectively by using qualitative research and analysis methods.*

C-6 *Introduction of a risk management method changes the behavior of participants in a system. This limits the validity of results from the empirical studies. While this is a common threat to validity in many empirical studies, it is particularly relevant to risk management due to sensitivity of risk identification and analysis tasks. It is likely that awareness of experimental interest in risks increases the sensitivity to identify risks and, possibly, introduces bias in risk analysis.*

C-7 *Software projects have relatively long cycle times and are costly. It is not feasible to set up real projects just to experiment with a management method. This limits the number of data points we will be able to obtain in a given time.*

The above constraints limit the empirical study design options available in risk management. From a traditional scientific perspective that relies on controlled experiments with statistically adequate number of data points, these limitations may seem so severe that reliable empirical studies cannot be performed. However, while these constraints are severe, they do not prevent us from applying systematic, scientific principles in our empirical studies. Recognition of these constraints allows us to design such empirical studies that provide more reliable results than anecdotal case descriptions.

A critical aspect of any empirical research is the selection and definition of research constructs that are to be studied. A construct in this meaning[19] refers to a concept that is used to abstract relevant information about a phenomenon (Judd et al. 1991; Rudestam & Newton 1992). A construct is a formal and precise representation of a phenomenon, it can be defined unambiguously, and metrics or variables can be defined to measure it in empirical studies. Figure 34 identifies four principle sources of construct types of empirical study designs for in the context of risk management methods: a method's impact on the system, its inputs and outputs, its characteristics, and its interaction with the system.

What we are ultimately interested in a method is its impact on the system, i.e., its environment. This aspect is presented in Figure 34a in red circles. In risk management context, this would mean, e.g., a risk management method's business impact. While we hypothesize that there is such an impact, we are unable to measure it, due to effort and time constraints of our research as well as the general constraints C-3, C-5, and C-7. Additionally, one of our method development requirements was to develop a complete risk management method. The resulting method we developed is, thus, a comprehensive one, combining several individual approaches into a single approach and an argument is made that the method as a whole makes a contribution. Consequently, it is difficult to test the method as a whole in a controlled setting and isolation, and testing of individual characteristics provides limited view of the method.

[19] Note that we used the term construct in different meaning in chapter 1.6.

The second source of constructs, evaluating a method's inputs and outputs, involves the study of what the method requires and produces, as is presented in Figure 34b using red circles. A risk management method uses people and information as inputs and produces various kinds of outputs, such as identified risks, ranked risks, corrective action plans, and implemented actions. Different risk management methods can be compared based on these factors. However, because of the effort and time constraints of our research as well as the general constraints C-2, C-5, and C-7, it is difficult to draw any strong conclusions about methods based on their inputs and outputs. Therefore, the input and output -based constructs should be supported by other empirical study constructs.

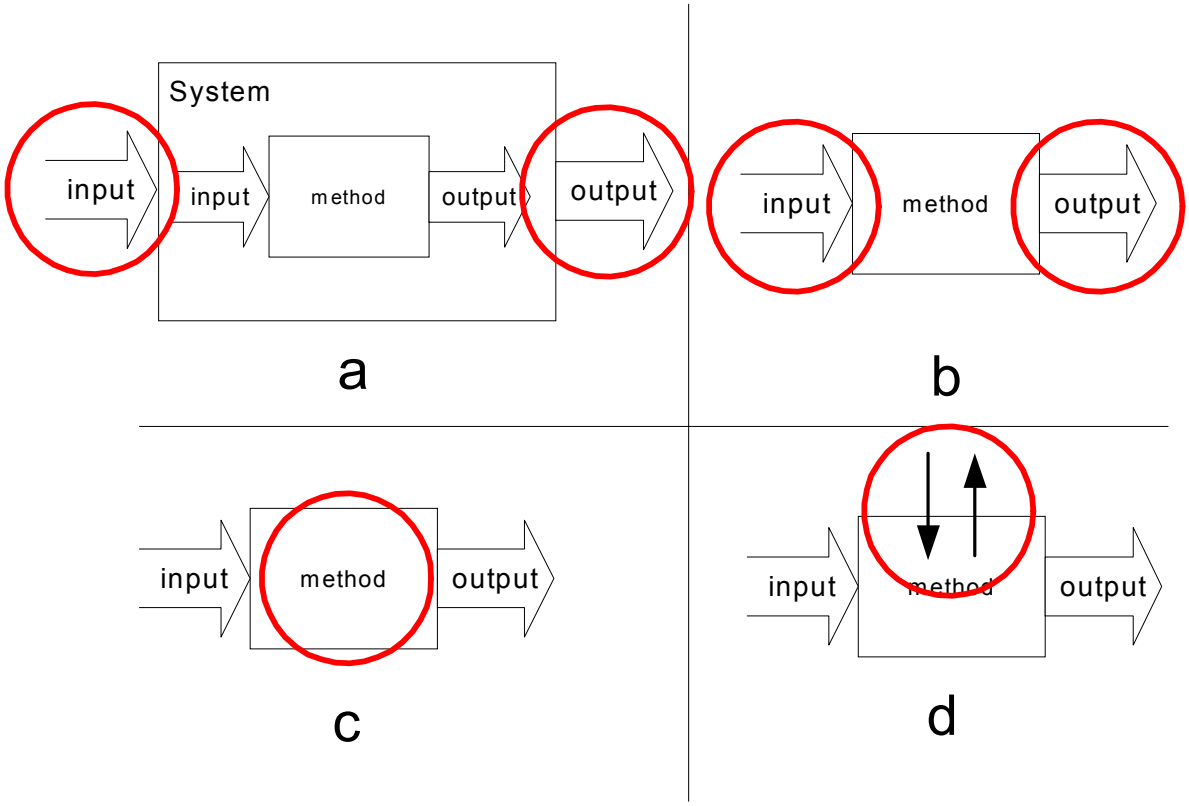


Figure 34: Principal construct sources in method validation

The third source of constructs, evaluating a method's characteristics (Figure 34c) is based on using measurements and observations about the method in order to obtain more information how the method works. This is done with the assumption that some of this information can be used to evaluate the method's usefulness or effectiveness. Examples of such characteristics are complexity of the method, availability of support material, level of detail in the method, coverage of the method, and measurement and analysis of the process and its intermediate results. We are using some method characteristics as indirect measures of effectiveness.

Finally, the fourth construct source, a method's interaction with the system, depicted in Figure 34d, can be used to study how method is used within the system. In risk management, this aspect is of particular value as risk management method's indirect goal is to improve the understanding of risk portfolio. For instance, we will use "credibility" as one empirical study construct as a way to measure how much confidence in the results the method creates.

In summary, our empirical study design will not be based on evaluating the impact of the Riskit method on the system (Figure 34a) due to various constraints we mentioned. Instead, the primary evaluation criteria will be based on evaluating the Riskit method's input and output (Figure 34b), complimented by observations and data about the method itself and its characteristics (Figure 34c), as well as measuring the impact of method's interaction with the environment (Figure 34d). The research strategy used in this work was to perform several case studies to provide sufficient depth of experiences for analysis, while conducting the case studies in several organizations to improve the generalizeability and validity of the findings. Additionally, we conducted one experiment the study the feasibility and usability of the Riskit Analysis Graph.

The empirical studies reported in this chapter had different goals and provide different angles in the empirical evaluation of the risk management concepts and methods produced in this work. Table 3 in chapter 1.3.3 listed the empirical studies and presented their objectives.

The first study reported in this work was actually the last one that was performed in terms of time. It focused on exploring current industry needs in risk management and as such, its purpose was to provide an updated view on the industry needs we had initially developed in the beginning of our work. We used a focus group technique to elicit industry representatives' views on current problems and challenges in risk management, and asked them to evaluate the perceived benefits and problems of the main results of our research contributions.

The second study, an exploratory case study with NASA, was the first application of Riskit in a real-world project. The project that used the method was very small and only one person used the method. The main purpose of this study was to obtain practical feedback on the method to develop it further. The empirical study findings, due to small number of data points and small size of the project, cannot be generalized, but the study itself gave us valuable information for the further development of the method. The empirical study design also was used as a reference point and example in constructing later studies.

The third study at Hughes Corporation and NASA was planned to be the first comprehensive study of the Riskit method in practice. A detailed plan was made for the study, project participants were committed, and training was organized and the study was off to a good start. Unfortunately, political and budgetary changes beyond the control of the participating organizations, let alone the project itself, caused a major restructuring of the overall program in which the study was to take place. Therefore, the study had to be cancelled after the initial two main steps of the Riskit cycle. Despite these limitations the Hughes study is included in this set as it contributed to the redefinition of the goal review and risk identification steps in the method.

The fourth study actually includes two empirical studies that were conducted simultaneously using the same empirical study design. DaimlerChrysler and Nokia each used the Riskit method in their projects and we were able to monitor these projects and capture information about their risk management process and experiences. These were the first studies that actually provided some information about the feasibility of the full method in large-scale projects.

The fifth study was performed by IESE in a project at Tenovis. The study aimed at characterizing the usefulness and cost-effectiveness of the Riskit method. The study indicated that the Riskit method is feasible and its users pointed out several characteristics

that were considered beneficial in practice. The study also produced several concrete improvement suggestions for the further development of the method and associate tools.

The sixth study reported in this chapter was an experiment that was conducted with students at the Helsinki University of Technology to compare different risk modeling techniques and to characterize their strengths and weaknesses. Three of the techniques were based on Riskit Analysis Graphs, one was based on a Riskit compatible form, and the fifth one was the SEI risk statement approach (Dorofee et al. 1996). The Riskit Analysis Graphs were selected as the objects of study because they represent the most visible and concrete aspect of the method to the users, they are central to the method and its underlying conceptual model, and because arranging a controlled experiment for them was more feasible than for the other characteristics of the method. We used non-parametric statistical analysis methods to discover patterns in the data and, even though the sample sizes were limited, several interesting findings surfaced, as reported in chapter 5.6.

The seventh item listed in Table 3 was the implementation of the eRiskit application. We have not included it as an empirical study in this chapter as it is described in chapter 4.7 and Appendix B. However, the implementation of the whole Riskit method, its process steps, the underlying conceptual models, and all practical dependencies between information entities is, in effect, a semi-formal proof of feasibility of the underlying conceptual model of risk in the Riskit method and, thus, could be considered as the seventh empirical study.

We are aware of several other situations, projects, and organizations that have been using the Riskit method in practice. However, since these cases have not been performed with sufficient scientific principles or we have not had access to the experience and data related to these studies, they are not included or reported in this work.

5.1 Study 1: Focus Groups

This chapter presents the objectives and results of a study that used the focus group method to study industry needs for risk management.

We chose the focus group method as the technique to conduct the study. The focus groups method is an approach to conduct interview surveys, using semi-structured discussions to identify issues or concerns (Edmunds 1991; Ghauri et al. 1995; Stewart & Shamdasani 1990; Templeton 1994). Focus group sessions consist of a series of meetings with three to twelve representative participants, who discuss and evaluate objects of study. The sessions are facilitated to follow a predefined, similar structure so that the sessions stay focused on the agreed theme. Focus group sessions produce mainly qualitative information about the objects of study. The benefits of focus group are that they produce candid, sometimes insightful information, and the method is fairly inexpensive and fast to perform (Widdows et al. 1991). However, the method shares the weaknesses of many other qualitative methods – biases may be caused by group dynamics and sample sizes are often small – and, therefore, it is difficult to generalize the results (Judd et al. 1991).

Focus groups are widely used in political science and consumer goods market studies to test product concepts or to evaluate political platforms or campaigns (Neter & Waksberg 1964; Widdows et al. 1991). The method has also been used in organizational studies to provide feedback on how to develop business services (Baker 1991). However, it seems that it is rarely used in software engineering research.

5.1.1 Objectives, Design and Practical arrangements for the Focus Group Study

Several surveys have been made about the risk management needs in industry. Ropponen et al. Have identified several common risks and characterized risk management practices in Finnish companies (Ropponen 1993). In their further study they classifies common risks and found a relationship between risk management practice and specific risk categories (Ropponen 1999; Ropponen & Lyytinen 2000). Several studies have also been made to identify common risks in industry. The VTT study identified and categorized common risks (Laitinen et al. 1993), the SEI risk questionnaire is based on a taxonomy synthesizing several projects' experiences (Carr et al. 1993), as has been done by Jones (Jones 1994).

While all these studies are valuable contributions towards better understanding of risks, these studies do not provide insight into why and how corporations seek to improve their risk management practices, what they intent to achieve with better risk management, and what are the impediments preventing more effective risk management approaches from being used. Furthermore, the Riskit method and the eRiskit application have some specific characteristics that, naturally, have not been addressed in these earlier studies or surveys. Therefore, we conducted a study that aimed at clarifying the following main issues:

- What are the risk management needs or impediments against risk management in the industry?
- How well do the specific characteristics of Riskit and eRiskit satisfy these needs or resolve impediments?

The focus group study was conducted jointly with a company[20] that had an interest in exploring similar questions from commercial perspective.

[20] R & D-Ware Oy.

The study objectives were formulated into a set of GQM statements. The first objective, stated in Table 43, was descriptive research goal, primarily aimed at capturing the current industry needs for risk management. We chose the affinity grouping technique (Brassard & Ritter 1994) to elicit the focus group participants' view on this issue. The participants were asked to spend a few minutes writing their responses on notes on the following question:

- *What are the most relevant problems in implementing risk management in your company?*

The responses were read aloud, briefly discussed if needed, and posted on a wall in a conference room. While the posting was being done, participants grouped the notes into categories so that similar issues were in the same group. Each participant had a unique, numbered set of notes so that the originator could be traced.

Analyze	Risk management needs in industry
In order to	*Describe* near- and long-term needs, growth potential, and *Understand* what are the most beneficial technology areas.
With respect to	Most important, current problems, and most important development areas.
From perspective of	Corporate decision maker, risk mgmt process owner, and technology provider.
in the context of	Product development programs, software development, and project-based business.
because	Understanding the industry needs is required to assess whether the research results have application and commercial potential.

Table 43: GQM statement for the focus group study on risk management needs

A second aspect related to the first goal of the study was to assess what areas of risk management require improvement and what are the actions to be done with them. For this we used the same affinity grouping technique and the participants were asked to spend some time thinking about the answer to the following question:

- *What are the most important development areas in risk management in your company and what are the actions your company plans to take?*

These answers were written on number-coded, differently colored notes and participants were asked to give a priority to the actions they recognized. The participants were asked to use the results of the previous session if needed, but were also encouraged to think of other actions that may not have been mentioned in the first affinity grouping session. Results were posted on the same board where the initial risk management needs were posted. All responses were documented for the analysis of the results.

The second objective of the focus group study centered on discussing how important the participants considered the main characteristics of the Riskit method and the eRiskit application. This study objective was phrased as a GQM statement, as described in Table 44.

Analyze	Proposed Riskit-based product offering, specifically, • outsourced training, • risk management system development, • risk management database customization, , and • eRiskit main features.
In order to	Evaluate the perceived value and potential business success of the proposed products/services.
With respect to	Perceived benefits, barriers for use, willingness to implement or obtain, need for local adaptations, and pricing.
From perspective of	Corporate decision maker (risk mgmt process owner) and technology provider.
in the context of	Selected focus group representatives.
because	Understanding the industry demand for the features reflects their potential or perceived benefits in practice.

Table 44: GQM statement for the focus group study on Riskit and eRiskit characteristics

As Table 44 shows, this study objective combined the research perspective for this work and the commercial interest of the company involved in this study. The information about the Riskit and eRiskit characteristics or their underlying elements was presented to the participants in the form of product concepts. This required that we had to convert the Riskit or eRiskit characteristics into concrete, operational product descriptions that could be understood and evaluated by the participants. We felt that the industrial participants in the session would be better able to relate to such descriptions better than to abstract or more technical descriptions of the characteristics.

Each concept was evaluated using the same format. First, a predisposition presentation, lasting usually a few minutes, was given to present each product concept, highlighting its main features and giving examples of its use. Then, a semi-structured discussion took place and participants were asked to voice their opinions on the concept. Table 45 presents the discussion outlines used.

The sessions also included some specific topics that were included to cover some specific commercial issues related to the company cooperating in the study. They mainly focused on evaluating the potential for an additional product concept (not related to Riskit), pricing issues, and on their marketing strategy. These results are not reported in this work.

We held three focus group sessions, first one was pilot session with one industrial participant, and the session was primarily intended to practice the focus group process and evaluate the questions. The data from the pilot session was also included in this report as only minor changes in question phrasing were made.

Each session started with an overview of the objectives of the study and with a discussion on how participants should discuss and act during the session. Special emphasis was given to participants ensuring that the participants' opinions should represent the real situation and opinions from their organizational perspective and that the study organizers guaranteed the confidentiality and anonymity of the discussions. Participants were also anonymous to each other, i.e., they did not know what organization they came from. The sessions were audio and video recorded so that transcripts of the sessions could be made to document all points rose.

Riskit characteristic / product concept	Discussion outline
Outsourced training	What are the benefits of outsourcing risk management training? What are the impediments for outsourcing risk management training? To what degree does the training need to be company-specific?
Risk management system development	What would be the benefits of using an external consultant in such development? What would prevent the use of external consultants?
Risk management database customization	What are the benefits of customizing a risk management database, compared to using an existing, "as-is" solution? What might prevent the development and deployment of such a database? How simple or complex should the database application be? How much would you be willing to invest in developing such an application (in effort or money)?
eRiskit main features	How important are the following features? Which of them are "must" features? What is the priority of "must" features? What features are unnecessary? • Decentralized collection and tracking of risk information, i.e., risk information can be collected and tracked at the point of their use • Risk information is accessible centrally in consistent form • Stakeholders and goals are included and kept consistent throughout the risk management process • Accurate and formal documentation of risk information • Visualization of risk information, e.g., using formalisms like Riskit Analysis Graphs • Use of sound and reliable techniques in risk prioritization • Archiving of risk information for improvement purposes

Table 45: Discussion outlines used in the focus group study

5.1.2 Focus Group Selection

We used three main criteria in selecting the focus group organizations. First, we included companies that were involved in either software development or project based-business. We included the project-based business sector in order to benchmark the experiences in software development field. Second, we wanted to find organizations from two categories of companies: (i) large, established organizations whose business volumes and size pose challenges to risk management, and (ii) smaller organizations that operate in fast-growing or turbulent business areas, such as e-commerce, Internet or multimedia companies.

We used subjective, non-probability sampling (Judd et al. 1991; Ghauri et al. 1995) to select 19 companies that corresponded to the above criteria and contacted them personally to ask them to participate in the study. We attempted to find either a risk management process owner or specialist or a business decision maker to participate in the sessions. We were not able to reach the four of the companies (or a person to discuss the participation) but all the rest agreed to participate in the sessions. However, due to last minute cancellations, total

number of participants was 12. Table 46 lists and describes the companies that participated in the study.

Type	Size[21]	Risk mgmt process	Representative
Novo Group Oyj	Turnover: 320 M€ Personnel: ca. 2 000	Processes and principles defined. Templates, checklists, and calculation forms exist. Systematic planning and tracking in big projects, variation in smaller projects	Project manager, responsible for project methods
TietoEnator Oyj	Turnover: >1 000 M€ Personnel: ca. 10 000	A two-phase RM methods have been described, risks are continually tracked	Head of quality, develops and supports RM process
Kemira Engineering Oy	Turnover: >10 M€ Personnel: ca. 100	Procedures exist but not always applied systematically, project managers decide whether, when and how RM is done	Leading Expert, develops and supports RM process
Nokia Networks	Turnover:>5 000 M€ Personnel: >20 000	RM process and principles defined, RM activities frequently performed, based on guidelines and standards	RM process owner, develops and supports RM process
ABB	Turnover: 1 700 M€ Personnel: ca. 10 000	Standard templates, checklists and risk calculation forms exist, control and tracking practices are defined and applied	Business development manager, develops and supports RM process
Satama Interactive Oyj	Turnover: >30 M€ Personnel: 400	Business risks are evaluated during the strategy planning process, no defined processes or methods for projects	Business development manager, develops and supports RM process
Kesko Oyj	Turnover: >15 M€ Personnel: ca. 200	Business risks are evaluated during the strategy planning process, controlling actions are tracked	Head of quality, develops RM process
Radiolinja Oy	Turnover: >500 M€ Personnel: ca. 1 000	RM actions are done on case by case basis	Business development manager, develops and supports RM process
Vaisala Oyj	Turnover: >150 M€ Personnel: >1 100	RM part of business planning RM included in project planning and tracking	Head of R&D division, develops and supports RM process
Datatie Oy	Turnover: 110 M€ Personnel: 270	Business risks are evaluated during the strategy planning process, RM included in project approval and tracking	Business unit director, develops RM process
SSH Com-munications Security Oy	Turnover: >15 M€ Personnel: ca. 200	Projects perform risk identification, analysis and track risks	Head of quality, supports RM process
Akumiitti Telematics Oy	Turnover: 1.5 M€ Personnel: 30	Included in R&D process, based on company practices	Head of R&D, supports RM process

RM = risk management

Table 46: Description of companies participating in the focus group study

[21] Financial and personnel information based on 2000 annual reports.

5.1.3 Control of Validity

We used three techniques to ensure that the research construct in this study was valid and in line with our research goals. First, we kept the content and format of the sessions and the presentations in them the same between the sessions. Second, instrumentation errors were reduced by using the audio and video recordings we described earlier. Third, we reduced the potential bias in interpreting the results by having another person review all interpretations made during the analysis.

5.1.4 Focus Group Analysis and Results

The focus group session results were documented in the notes used during the first part of the session and in the video and audio recordings used during the sessions. Both were transcribed into a document for analysis. The discussion transcript was issue-based, i.e., each issue or point raised was documented verbatim, but the transcript did not include clarification discussions, jokes, or other non-related communications in the meeting. In total, the focus group session transcript included 455 individual points that were raised and recorded, in addition to priority tables and rankings pooled from the participants. Each unique point was numbered for traceability.

The analysis method used was based on pattern-matching the findings against the theoretical propositions we had made (Yin 1994). Except for the affinity grouping session, we used the product characteristics as the propositions we compared the comments against.

5.1.4.1 Risk Management Needs

We used two methods for synthesizing the results of the affinity group -based sessions on risk management needs. We will first present the bottom-up affinity grouping results, and then present the proposition based analysis results.

The affinity group -based analysis produced a structured list of risk management needs, as shown in Figure 35. We will discuss the findings in the following in more detail.

Several participants raised issues related to motivation and competence required to perform risk management. As shown in Figure 35, we categorized these points into four groups: attitudes, awareness, motivation, and competence. Regarding attitudes, participants mentioned that organizations are prone to unjustified optimisms and downplaying of potential problems. It was also pointed out that talking about risks can be considered pessimism, and such negative views are discouraged in teams. Participants also mentioned that organizations easily work in reactive mode and risk management is not a required activity: individualism and lack of process-oriented personnel may prevent systematic risk management to take place. They also felt that risk management is perceived as a formality or an overhead that does not actually contribute to critical tasks.

Lack of awareness about risks and risk management was another theme that surfaced. Participants mentioned that the purpose of risk management at project level is not understood, making it difficult to communicate about risk management and motivate people to perform it. Participants' different backgrounds pose additional challenges to developing a common view on risk management. Difficulties in identifying risk and lack of information about risks were also listed as problems making risk management difficult.

Category		Issues
Motivation and compe-tence	Attitudes	Unjustified optimism: need for risk management not acknowledged Proactive management not emphasized in the management culture Individualism valued more than systematic process approach Risk management perceived as a non-value adding activity
	Awareness	People do not understand the purpose of risk management People are do not know the possibilities of risk management
	Motivation	Objectives for risk management are not clear Fear of difficulties prevents discovery and analysis of risks
	Competence	Lack of knowledge about risk management Lack of skills for risk analysis
Manage-ment	Management practices	Lack of risk management practice Risk management not linked to management model or system Lack of tracking and controlling of risks Lack of measurement of risks No link between strategy and operational level in risk management Change of personnel causes discontinuities
	Performance Pressures	Tight goals and increase in workload reduce time available for proactive actions Other supporting processes compete for time for risk management Perceived high overhead of risk management
Communication		Communication between functions difficult due to lack of interaction Difficulties in communicating about risks due to different backgrounds of people and due to different concepts and approaches
Environment	Competitive Situation	Several factors increase the risk levels for a company but not necessarily the competence for risk management: ▪ Competitors' actions ▪ New technologies and markets ▪ Complexity ▪ Large number of partners
	Customer	Customer may not be used to discussing risks openly Customer commitments may be given too easily, increasing risks Contracts take a static view on risks Uncertainty about customers expectations
Risk management system	Methods	Risk management methods are not cost-effectiveness Risk analysis is not objectivity and soundness Lack of working methods Domain specific support is not available
	Integration into the process	Risk management is done too late in projects
	Common approach	There is no common framework Methods are not used systematically
	Feedback for improvement	Procedures for learning from risk management experience are missing

Figure 35: Affinity grouping results of risk management needs

It was seen that management may fail to express the need and objectives for risk management, without them it is difficult to empower project to perform risk management. It was also mentioned that fear of risks may prevent active analysis of risks.

Lack of knowledge about risk management was mentioned as the main problem related to competence. This could result in wrong assumptions or business calculations. It was also mentioned that the customers and vendors do not know how to deal with risks in their interactions.

Two main issue groups were identified under the topic of management. First, current management practices may not relate to risk management at all, or if they do, they may not link the strategic and operational level risks to each other. If the management does not communicate what kind or level of risks should be taken, it is difficult to discuss and manage risks. Linking organization's measurement system to risks was also considered a challenge, as well as discontinuities due to change of personnel.

Second, performance pressures were listed as a hurdle that prevents practitioners from performing proactive risk management. Projects have tight schedules, personnel are working under increasing workload, and there may be pressures to announce delivery deadlines prematurely. At the same time there are other, competing support processes that burden limited resources and risk management may be perceived as bureaucratic, non-productive activity that slows down the organization and its ability to react.

Communication was mentioned as the third main problem group. This mainly referred to communication problems between functions, e.g. marketing and R&D, about potential risks, their impact to business and how they could be controlled.

We grouped competitive situation and customer under the category "environment". Competitive situation may influence risk management in many ways. First, complexity of competitive situation, new markets or technologies, large number of partners, or fast pace of business may make it difficult to identify or model risks to a concrete level. Fears about losing competitive advantages or economic trends may prevent management from discussing scenarios that are perceived as too bleak.

Second, customer influences risks in many ways. Commitments made, often very early and with little real information, may become objectives that cannot be questioned. It is often also not clear if and how to discuss risks with a customer. Different customers also have different expectations and needs on the relationship and communication. Contractual issues are also difficult to link into the risk management context.

Most comments in the risk management need session related to risk management system. We grouped these comments into four categories, as shown in Figure 35. First, lack of practical, working methods or conventions was considered a problem. Participants called for systematic and sound approaches that would help in resolving different opinions on risks objectively. Common and clear framework or terminology was seen as a potential solution to this. Lack of domain specific risk categories was considered to be a problem that limited recognition of specific risks, on the other hand, high volumes of identified risks was mentioned as a problem by one participant. Risk prioritization and prediction in general were considered problems, as well as understanding the deeper causal relationships between potential events.

Second, participants listed integration of risk management into organization's processes as a problem. This may result in risk management entering the picture too late, or in too limited view in evaluating risks.

Third, participants called for a common, systematic approach to risk management. A common framework and practice would help communication, improve efficiency of risk management, and reduce the friction caused by different local conventions and terms. Such a framework would also make projects and their risks more comparable.

Finally, the importance of utilizing feedback from risk management for improvement was also mentioned as one problem that prevented more effective risk management.

The second analysis was based on the suppositions mentioned in chapter 1.1 and the method requirements listed in Table 1 to synthesize a set of theoretical propositions, shown in Table 47. We grouped the affinity group findings initially by the groups as they were identified in the sessions. We then regrouped them into categories corresponding the propositions in Table 47. For each proposition we synthesized the affinity grouped results and concluded to what degree the results supported the proposition or not, or whether alternative propositions could be formulated to correspond to the affinity grouping results.

Code	Proposition	Origin
P-1	Organizations want to have more cost-effective risk management methods.	S.1 (and R-9)
P-2	Organizations want to have systematic and explicit methods for risk management.	S.2
P-3	Organizations want to use methods with low overhead.	S.3
P-4	Organizations want to be able to demonstrate the benefits of risk management.	S.4
P-5	Organizations want to have methods that are not sensitive to individual users different backgrounds, skills, or attitudes.	R-1
P-6	Organizations want to have methods that are easy to learn and use.	R-2
P-7	Organizations want to have risk management methods can be applied to different situations and projects.	R-3
P-8	Organizations want to have risk management methods that are consistently applied in practice	R-4
P-9	Organizations want to have methods that comprehensively cover risk management activities.	R-5
P-10	Organizations want to have methods that accurately describe their real-world risks and decision-making problems.	R-6
P-11	Organizations want to have risk management methods that produce credible results.	R-7
P-12	Organizations want to have risk management methods that support communication about risks.	R-8

Table 47: Theoretical propositions used in the analysis of the risk management needs

The proposition P-1 in Table 47 seemed to be supported by several arguments. On one hand, participants mentioned as a problem that risk management is perceived as a non-value-adding activity, and, on the other hand, the increased competitive pressures require higher effectiveness from risk management activities.

The need for systematic and explicit risk management methods (proposition P-2) was supported by three points: lack of past practice in risk management was considered a problem, common practices were called for, and lack of objectivity in risk analysis was considered a problem.

The need to have low-overhead methods (proposition P-3) was supported by several observations, mainly warning about too bureaucratic procedures, increased workload, and limited time available for risk management.

The need to demonstrate benefits of risk management (P-4) was only indirectly supported. Single statements were made to improve measurement related to risk management and to establish feedback process for risk management.

The consistency risk management results (P-5) were supported by concerns about people's different backgrounds, their competence levels in risk management, their attitudes, and about their awareness about risks.

The ease of use for methods (P-6) was supported by several comments, e.g., situation complexity, large number of partners and lack of knowledge make risk management difficult and participants lacked working, practical methods.

The need to apply risk management in different situations (P-7) received only marginal support from comments: participants recognized the need to react to different customer needs and need to deal with new technologies and situations.

The consistency of applying risk management (P-8) was emphasized in several comments. Several causes for not performing risk management were mentioned (individualism, culture, lack of management enforcement, time pressures), yet systematic and consistent risk management practice was called for. Also, consistent practices across different projects was mentioned as a goal.

The comprehensive coverage of risk management (P-9) indirectly supported by two comments relating to lack of risk management in early phases of projects, and in dealing about risks with customers.

The need for accurate and realistic risk management (P-10) was supported by calls for more concrete analysis approaches, resolving conflicts, and the need to be able to model the real complexities in project environment.

The credibility of results (P-11) was supported by comments that related to risk analysis. Prioritization, analysis, and understanding the causal relationships were considered necessary, while lack of competencies was seen as a hurdle.

Finally, communication (P-12) was mentioned in the context of cross-functional communication about risks and about communicating about identified risks.

In summary, the analysis of affinity grouping results produced direct support for propositions P-1, P-2, P-3, P-6, P-8, P-10, and P-11. The other propositions received only indirect or marginal support from the comments. It is interesting to point out that while there were several points that could not be matched to the propositions we used, none of the comments in this part of the analysis (78 in total) was contradictory to any of our propositions.

While the number of participants and nature of comments, as well as the inevitable interpretation problems limit the generalizability of these findings, the support for all propositions and, especially, lack of opposing points seem to support the notion that the

suppositions in our research (chapter 1.1) and the requirements for our method (Table 1) correspond to industry needs.

5.1.4.2 Risk Management Improvement Focus Areas

Participants in the sessions were asked to list and prioritize actions or improvement areas in their organizations in risk management. We present the main findings from this analysis in the following.

All but one participant listed improving awareness about risk management and risk management methods as the action to be implemented. The awareness training was mainly targeted to top and middle management, whereas tool and method support was targeted to project level. Several participants mentioned the need to change attitudes and culture in order to show results. Part of the awareness training, in many comments, was the introduction of a common framework and concepts to support better communication about risks.

Another common theme in improvement actions was the integration of risk management into organization's processes and management practices. It was suggested that risk management should be an integral part of management, not an isolated activity. At the same time, continuous application and use of methods and techniques was required and expected as the result of awareness training.

Several participants also emphasized the need to introduce systematic, yet simple methods and tools to support risk management.

In general, all participants described a fairly comprehensive and ambitious list of activities. Naturally, such lists were presented without too much concern about available resources and therefore they probably do not represent an accurate estimate of what actions will happen in these organizations in the near future. However, they represent the relative importance of the identified actions reasonably well. As the discussion above shows, training to support awareness and specific methods, integration into other processes, and simple but systematic methods seem to be the most important improvements for these organizations.

5.1.4.3 Outsourced Training

The first product concept that was evaluated was outsourced training. It consisted of a service to customize professional quality training material, making it available to personnel, and the provision of the training to personnel.

The main benefits of such training included being able to introduce industry best practices in to the organization, more effective use of in-house resources, and speed of deployment. However, several participants expressed concerns whether an outsider can understand the domain specific and unique nature of a company's risks and their risk management approach.

Sensitivity of risk related information might be a hurdle that prevents closer cooperation with an outside partner; there is an "inevitable fear that confidential information leaks out". It was also pointed out that any company's essential core competence is its ability to deal with risks in its business area. Using an outside partner gives an impression that the company cannot master its core competence.

Most participants felt that basic terminology, concepts and industry best practices are common and relevant to all companies. All other aspects of the risk management training would benefit if they were customized to the organization.

In summary, given the high importance of training identified earlier, organizations seem to be able to use some external training, but due to sensitive nature of the topic, they also need to develop internal competencies to internally deploy and support risk management training and practice.

5.1.4.4 Risk Management System Development

The risk management system development concept was based on the idea of offering organizations a consulting service to develop and deploy a risk management system. The term risk management system in this context means processes, methods, and tools that are adapted to organization's needs, integrated to its processes and management system, and deployed into practice.

The main benefits of using an external partner was considered to be the fast introduction of best practices into a company. An important aspect of this is the potential of finding out how the "best of breed" in industry are managing risks. External partners can also dedicate their effort better to risk management and therefore deliver results faster than in-house personnel that are burdened with many other duties.

Main challenge in using an external partners are the difficulties in learning enough about the specific aspects of customers business in order to add value. Many practical aspects of risk management are company-specific and it may not be cost-effective to teach an outsider these basics.

It seems that the development of a risk management system require also strong in-house commitment and team, even when competent, external support is available.

5.1.4.5 Risk Management Database Customization

The risk management database customization concept included the adaptation of an existing database product, such as Lotus Notes or a Microsoft Access, for risk management purposes by developing a simple application that contains company specific terms and approach for entering and tracking risk information.

The importance of customizing the terminology of such an application was emphasized by several participants. In addition, it was agreed that such a system should not be an isolated system, it should be linked to other applications in the company.

Problems associated to a customized application included maintenance problems, and defining a manageable scope for the system. It also seemed that companies are willing to spend much money for developing such an application. There is a risk that a marginal investment will not produce and application that would have sufficient functionality.

It seems that a stand-alone, custom-built application is not a feasible solution. Instead, there is more potential in developing an application that is integrated with existing business applications or business frameworks (such as SAP) or developing a packaged software application that has sufficient functionality and interface capability, as well as competitive pricing, are strategies that are more appropriate.

5.1.4.6 eRiskit Main Features

The participants discussed and ranked the main features of eRiskit application. We selected seven main features and their benefits and trade-offs were discussed in the sessions and

participants gave individually feature rankings at the end of the discussion. The features included in the evaluation are listed in Table 48.

Feature	Ranking of the feature	# of "not needed" votes
Decentralized collection and tracking of risk information, i.e., risk information can be collected and tracked at the point of their use	39	2
Risk information is accessible centrally in consistent form	38	1
Stakeholders and goals are included and kept consistent throughout the risk management process	31	0
Use of sound and reliable techniques in risk prioritization	28	2
Archiving of risk information for improvement purposes	20	1
Visualization of risk information, e.g., using formalisms like Riskit Analysis Graphs	20	2
Accurate and formal documentation of risk information	5	5

Table 48: eRiskit features evaluated in the focus group

The distributed risk management data collection and tracking was considered an important feature. Most discussants assumed that an intranet-based solution would be preferable and practical, perhaps being a basic competitive requirement for such a product. Especially companies that have several geographical locations would benefit from such a feature.

A centralized access to risk management information was also considered an important feature, "without it the data is useless". The main thing is that data that is collected must be utilized in business decision-making. However, several discussants expressed concern over the confidentiality and security of such information.

Use of stakeholders and goals in risk analysis received mixed comments from participants. Some considered it an essential feature that allowed the linking of risks to business objectives; others did not see that they would bring added value over the increased complexity these concepts would bring along.

Reliable prioritization of risks was considered a less important feature. In normal business decisions the accuracy of information related to risk management decisions does not justify the time and precision to be spent in using advanced methods for risk prioritization. Crude approximations are sufficient.

The capturing of risk management information for process improvement purposes considered a valuable feature but not necessarily a near-term priority for the participants.

The graphical visualization of risk information was considered to be an essential feature. It helps communication and understanding of risks. However, there are several commercial tools that can be used to provide such graphical drawing functionality, this application should add value compared to using such tools.

The accurate and formal description of risk information was not considered an important feature. Most participants were concerned that such details would make the system unusable. However, ideally the system would allow more details to be entered and used when needed, yet allow simple cases to be handled with simple user interface and limited data.

The results of the feature prioritization are given in the second column of Table 48. We used a simple ranking algorithm to consolidate ranking information: the maximum number of

points to be given was determined by the widest ranking scale used by the participants, in this case it was six. We assigned the maximum number of points to highest ranks and each subsequent rank received one point less. The right-most column in Table 48 indicates how many participants considered the feature unnecessary.

5.1.5 Study Conclusions

The focus group study was a qualitative study to explore current risk management needs in industry. We used our theoretical suppositions, method requirements, and the eRiskit application requirements as a framework to analyze the study data.

The industry risk management needs seem to be in line with the underlying themes identified earlier in this work. Especially, the need to have more effective, systematic, cost effective, and easy-to-use methods was a common requirement in the focus group. The classification of risk management needs provides another perspective to industry needs and it can be seen as a model that complements improvement frameworks, such as Hall's maturity model (Hall 1995): our model highlights underlying issues that help improve the prerequisites of a good risk management system.

The product concept evaluations provided several important insights into what type of risk management solutions are likely to be deployable in industry. The theme of simplicity was observed in many discussions. Another important finding was that organizations will need to invest in their own risk management competence – risk management is not a service that can be "outsourced" completely, in-house insight and capability will allow more effective deployment of risk management practices.

The focus group study did not confirm some other important principles that we have used in this work. The need to accurate and more formal description of risks was not seen as an important characteristic. Nevertheless, we are emphasizing its importance for two reasons. First, more accurate definition of risks will eventually allow more focused discussion about risks, making the analysis and communication about risks more effective. Second, improved risk documentation will improve the risk management experience capture, leading to faster improvements in risk management technology. We believe that these benefits will become more important to organizations as they become more advanced in the use of risk management technology.

5.2 Study 2: Exploratory Case Study at NASA

In this chapter, we present the results of the first, exploratory empirical study of the Riskit method in industrial context. The study was carried out at the Software Engineering Laboratory (NASA 2001). The SEL is a partnership organization that was established in 1976 at NASA Goddard Space Flight Center (GSFC) by its Flight Dynamics Division (FDD), Computer Sciences Corporation (CSC), and the department of Computer Science at University of Maryland. The SEL was established for understanding and improving the software products and development process in the FDD.

The SEL supports the software development within the FDD. Software developed by the FDD is mainly scientific applications that process data received from earth orbiting satellites in the areas of orbit, attitude, and mission analysis. The total FDD software development staff, including contractor support, is approximately 250-275, and about half of this is allocated to software maintenance. Typical project involves between 5 to 25 staff members and results in system size of 100-300 KSLOC. The SEL itself has a staff of 10-15 analysts (McGarry et al. 1994).

The project selected for study was a small utility that was part of the Flight Dynamics Support System (FDSS) developed by the FDD in support of the Tropical Rainfall Measuring Mission (TRMM). The utility, known as the Maneuver Command Utility (MCU), produces spacecraft maneuver command sheet for use by mission operators. The project had been estimated to be approximately five person months in effort and was scheduled to take place between October 1995 and January 1996, including independent system testing. Two people had been assigned to the project along with the project manager. The project organization in our study used a systematic risk management approach (hereafter called the comparison method) that was supported by a spreadsheet-based tool. The project manager that participated in our case study had been using the comparison risk management method for about three years and had used it in close to ten projects. The method currently used by the organization is referred to as the comparison method.

5.2.1 Objectives, Design and Practical arrangements

The objectives of the case study were to characterize the feasibility and cost-effectiveness of the Riskit method in an industrial project and compare the Riskit method with the method currently used by the project. Furthermore, the case study was used to provide practical feedback on the use of the method. The GQM goal statement for the study is presented in Table 49.

We considered the Riskit method feasible, which was our hypothesis, if (i) it produces intended results, (ii) it can be applied within reasonable time and effort, and (iii) the users of the method give a positive opinion of its feasibility.

We arranged our case study so that we were able to apply two risk management methods during the project. The purpose of this arrangement was to provide a better basis for our qualitative analysis of the Riskit method and to characterize the comparison method and the current risk management practice in the organization. It is important to point out that our goal was not to compare the methods per se or to assess which one is “better”. We recognized that a single case study would not justify such conclusions.

Analyze	The Riskit process.
In order to	*Describe* and *understand* it.
With respect to	Feasibility and cost-effectiveness.
From perspective of	Risk management method developer.
In the context of	NASA Software Engineering laboratory and Riskit developer.
Because	Feedback on industrial feasibility and cost-effectiveness is important in order to assess if and how the method should be developed further.

Analyze	The Riskit process
In order to	*Compare* it
With respect to	Effort, granularity, coverage, accuracy, and effectiveness.
From perspective of	Risk management method developer.
In the context of	NASA Software Engineering laboratory and Riskit developer.
Because	Having an industrial benchmark method helps assessing benefits and disadvantages of the method.

Table 49: GQM statements for the NASA study

As Figure 36 shows, the case study started by a joint session where project goals were reviewed and risks identified. Using the list of risks produced the project manager used the comparison method to carry out risk analysis the way he normally does it. After this, the risk analysis using the Riskit method was carried out. After both analyses, the project manager decided on which risk controlling actions he should actually take.

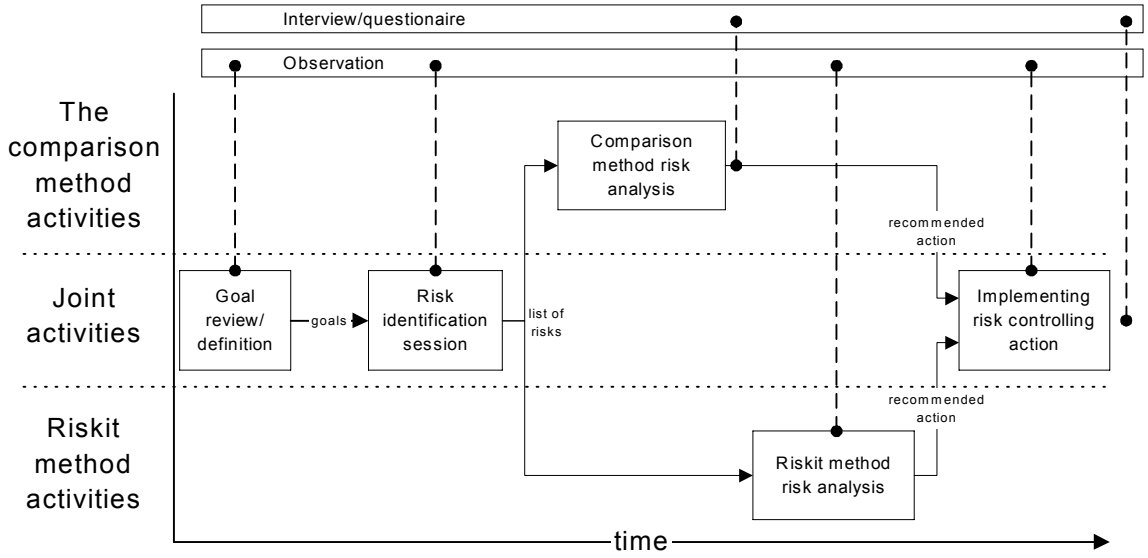


Figure 36: The timeline of case study activities

The project manager performed the first risk analysis on his own and documented the results of his analysis, including the risk controlling action he was planning to take. The Riskit method was applied in a session where the method expert (i.e., the method author, J. Kontio) facilitated the session. Figure 36 also shows where and how we collected the case study data. A dashed line to the vertical line from a case study activity indicates whether we

used observation or interviews and questionnaire to obtain relevant data. A connector appearing after an activity box indicates that the information was obtained after the activity was completed.

As we had only a single project in the study we were forced to apply the methods in sequence and this may have lead to some maturation effects (Campbell & Stanley 1963; Judd et al. 1991), i.e., the accumulated time spent on risk management may have increased participant's awareness and knowledge about risks. We tried to minimize this effect by taking two specific actions. First, even though the dedicated risk identification session is a characteristic of the Riskit method and not of the comparison method, we decided to conduct a joint risk identification session for both methods. We reasoned that risk identification would be especially vulnerable to maturation effect and could seriously bias the results. As risk identification was not a main aspect of the Riskit method, we did not consider this a serious compromise in the method comparison. Second, we avoided analyzing risks in the identification session. We simply listed candidate risks and tried not to analyze or discuss them in any detail.

The sequential application of methods may also have caused a multiple treatment effect: the latter, Riskit method application may have been influenced by earlier analysis done using the comparison method. We tried to control this threat by carrying out the latter risk analysis as independently from the comparison method analysis as possible: we asked the project manager not to think about the results of the comparison method, we used the original list of risks as a starting point, and we facilitated the Riskit risk analysis session according to the Riskit method. Two observations lead us to believe that multiple treatment effect did not occur or was minimal: the risks selected for analysis were different and the method user clearly indicated that the analysis processes were so different that he himself did not observe any effect, the Riskit method seemed to have immersed the user so that the previous analysis did not influence the second analysis.

As the Riskit method sessions were observed and the session notes reviewed shortly after each session, the observations were not affected by this threat. It should be noted that many of the original metrics and questions turned out not to be applicable in the study or produced no responses from the method user. In retrospect, these questions and metrics seemed to have been the result of our attempts to "over-measure" the study.

The fact that we facilitated the Riskit risk analysis session may have caused a different kind of bias in the results, i.e., a construct validity threat similar to the Hawthorne effect (Cook & Campbell 1979). It is plausible that the facilitator may have contributed to the analysis or that the mere presence of a facilitator and a scribe may have improved the performance of the project manager. We tried to minimize these effects by maintaining a strictly facilitating role in the analysis (we refrained from actually making any judgments or conclusions) and by strictly following the Riskit method. However, we cannot rule out the possibility that either our participation or unconscious contributions might have affected the analysis.

As the method developer was involved in the execution of the study and in the analysis of the results the experimenter expectancies may have influenced the results. We tried to control this threat by involving an experimenter whose sole research interest was in the experimental design and by documenting the case study results and outputs in detail in this report. This way outside, objective readers can evaluate possible bias independently.

Overall, we believe that our study design and arrangements prevented any significant validity threats to our results. The two most important validity threats relate to constructs used: the Riskit method changed two important parameters in risk analysis: the amount of effort spent and number of people participating. With the Riskit method, more time was spent on risk analysis and risk control planning than with the comparison method. With the Riskit method there also was a member of the technical staff present in the analysis session present. While these factors quite likely had an effect on the results, they are also characteristics of the Riskit method. In other words, they were part of the control variable that we wanted to study.

5.2.2 The Comparison Method

The case study organization provided most managers with training on risk management, primarily focusing on the risk management tool that was used (Kontio et al. 1996). Risk management was a required activity in all projects and risks are discussed with the management and customer frequently and risk estimates were normally updated monthly. The risk management approach is supported by a spreadsheet-based tool that guides risk analysis and helps in quantifying and ranking the risks. This internally developed tool had been in use since 1992 and it had been updated and improved during its usage. This risk management tool seems to have been the driver of the risk management process in projects.

The comparison risk management tool collected basic information about each risk, i.e., risk title, risk description, risk source, risk impact, importance to the customer, current status, and probability of occurrence. The tool also collects information about the impact of risk if no mitigation action is taken, estimating the impact on quality, schedule impact, and cost impact. Once each risk has been identified, information about risk mitigation plans is entered into the tool, i.e., a description of the risk mitigation approach, the trigger that is used to initiate the risk mitigation, quality impact of the risk mitigation, schedule impact of risk mitigation, cost impact of risk mitigation, and probability estimate of risk mitigation success.

The above information is used to calculate the risk analysis results using three scenarios (i) risk does not occur and no mitigation is done, (ii) risk occurs and mitigation is done but fails, and (iii) risk occurs, mitigation is done and it succeeds. The decision of the appropriate risk mitigation action is left to decision makers evaluating the risk analysis data. The main benefit of the method is that it forces projects to think about risks frequently, every month. The approach also gives a quantitative indication of whether risk mitigation should be done. The results are often used in the decision making with management. Among the problems associated with the tool, it was mentioned that probability values are difficult to obtain and there is little support for estimating them, yet they play a critical role in the risk analysis process. (Kontio et al. 1996)

5.2.3 Analysis and Results

We have reported the study results in detail in a separate report (Kontio et al. 1996) and present here the main findings of the study.

The risk management process that was enacted resulted in the identification of two stakeholders, six goals, 19 risks, and 12 potential controlling actions. In total, 20 hours were spent on risk management, representing about a third of management time and about 3% of the project's total effort (Kontio et al. 1996). Main findings are summarized below.

We used questionnaires and interviews to inquire the method user's experiences and opinions about the two methods. The user characterization of the method, compared to the comparison method, can be summarized as follows:

(f1.1) Riskit has a well-defined process and is easy to use.

(f1.2) Riskit uses a sound and systematic approach in risk identification and prioritization.

(f1.3) The graphical output of Riskit is complex, but summarizes risk information well.

(f1.4) The credibility of Riskit results seems to be higher, i.e., the user trusted the Riskit results more than those of the comparison method.

(f1.5) Riskit is a complete and thorough method, suitable for longer, high-risk projects.

(f1.6) The comparison method is easier to use, it has a well-defined input format, and it presents summary of risks better.

(f1.7) Riskit consumes more resources than the comparison method.

We compared the methods' granularity, coverage, and accuracy by defining a set of specific metrics for risks and controlling actions that were produced. We realized that a mere counting of risk or controlling actions fails to account for the granularity and coverage of respective items. Thus, we use the following additional metrics to characterize the methods:

- Number of *same risks/actions* produced by the method, i.e., risks/actions that are judged to be same or very similar to a risk described by the other method.
- Number of *unique risks/actions* produced by the method, i.e., risks/actions that have not been identified by the other method and which do not overlap or are subsumed by other method's risks/actions.
- Number of *subsumed risks/actions*, i.e., risks/actions that are subsets of risks/actions identified by the other method.
- Number of *containing risks/actions*, i.e., risks/actions that include one or more of the risks/actions identified by the other method.
- Number of *overlapping risks/actions*, i.e., risks/actions that have some similarities but do not belong to any of the previous categories.

We used the above definitions to classify the risks selected for risk control planning and the controlling actions that were produced. Riskit method analyzed more risks than the comparison method. However, the most significant difference between the methods is the number of unique risks produced by the methods: Riskit analyzed five unique risks compared to one of the comparison method's. Given the data about the analyzed risks in the case study, the risk management methods seem to differ in their coverage. If we assume that the union of analyzed risks represents the "real" risks in the situation and count same, subsumed, containing, and overlapping risks as one instance each, the *risk coverage ratio* was 38% for the comparison method and 88% for Riskit, leading us to suggest that

(f1.8) Riskit results in effective covering a wide range of risks.

We repeated a similar process for risk controlling actions that were produced. The Riskit method proposed more controlling actions than the comparison method. It also produced a higher number of unique controlling actions. Using the same principle as above, the coverage ratio for risk controlling actions was 44% for the comparison method and 75% for Riskit. These figures suggest that the coverage of actions proposed by the Riskit method is higher, i.e.,

(f1.9) Riskit proposed a wider range of actions to be considered for implementation.

We assess the accuracy of the methods indirectly through the risk controlling actions that were actually taken in the project vs. the actions that were planned. The rationale for this metric is that we assume that the project manager, as a rational decision maker, will take the necessary cost efficient action in the project as further information about the project becomes available. Any action that was planned but not implemented indicates that (i) risk situation changed after the action was planned, (ii) the action did not address a big enough risk to justify it, or (iii) the action was not considered effective enough to justify its costs.

According to the project manager, there were no recognizable changes in the risk situation after the risk control planning and taking the action. Thus, we are using the ratio "implemented actions / planned actions" as an indicator of the accuracy of the results produced. The comparison method's ratio was 44% and Riskit's 83%. These figures lead us to suggest that

(f1.10) the Riskit method is more effective in proposing accurate risk controlling actions, i.e., it proposed actions that were considered worth implementing in the project.

Our first goal was to investigate the feasibility of the Riskit method in industrial context. The criteria we defined for determining feasibility were met. First, the method produced intended results (identified risks, ranked them and proposed controlling action). Second, the overall effort spent on the use of the method was 12 + 8 hours. This is about 3% of the total effort in the project, i.e., within the effort range proposed by Ropponen's survey (Ropponen 1993). Third, as we reported earlier in this chapter, the method user gave a positive assessment of the method with respect to its thoroughness, indicated a higher level of confidence in its results, and considered its risk ranking approach more sound. Thus, this study indicates that

(f1.11) Riskit is a feasible risk management method in industrial context.

We defined two derived metrics to characterize efficiency. The first one, *risk coverage efficiency index*, utilized *the risk coverage ratio*, and the effort used for risk management using the method. The rationale for this metric is that the risk coverage ratio represents the best available information of the coverage of all relevant risks in a situation. Dividing this by the effort expended to reach that coverage gives an indication of a method's efficiency in risk analysis. Comparison method's *risk coverage efficiency index* was 13% and Riskit's 7%, indicating that the comparison method was more efficient in risk management. In other words,

(f1.12) Riskit was less efficient than the comparison method in analyzing risks.

The second metric, *risk controlling action efficiency index*, utilizes the concept *risk controlling action accuracy ratio*, and effort for the method. The rationale for this method is that the total of implemented actions represent the best available information about the correct action to take in a situation. As the *risk controlling action accuracy ratio* numerically describes how well the method was able to produce the ideal set of actions, normalizing the *risk controlling action accuracy ratio* by effort expended gives an indication of risk controlling action efficiency. Comparison method's *risk controlling action efficiency index* was 15% and Riskit's 7%, again implying that

(f1.13) the comparison method was more efficient in identifying risk controlling actions.

5.2.4 Study Conclusions

Even though this study used methods and compared them through various metrics, the purpose of the study was not to assess which method is "better" based on the study data. In fact, the study essentially produced only one data point per method and as such, no conclusions could be drawn to that effect. However, the use of two methods was very useful in terms of benchmarking and providing a basis for reflecting findings between the methods.

The main finding of the study was that the Riskit method is a feasible approach for risk management in industrial context (f1.10). While it had higher overhead, its benefits seemed to outweigh the added time spent with the method. The Riskit method also seemed to be effective in covering potential risk scenarios (f1.8), as well as proposing risk controlling actions (f1.9). A higher proportion of the risk controlling actions proposed by the Riskit method also were implemented, indicating that Riskit produced more appropriate actions (f1.13).

This study suggested some potential strengths and weaknesses for the Riskit method. As strengths, it seemed to increase user's confidence in results (f1.4), it is well defined (f1.1), it seemed to produce detailed risk description (f1.2, f1.3, and f1.5), and its graphical orientation seemed to communicate some aspects of risks well (f1.3). However, it was perceived as complex in some situations (f1.3 and f1.6), it consumed more resources (f1.7), and it seemed to be less effective in producing output for the user (f1.12).

The results of the study were used as input when the Riskit method version 1.0 was developed and released (Kontio 1997). In particular, the Riskit Analysis Graph formalism was redefined, the Riskit Pareto ranking approach was modified, and several adjustments were made to the Riskit process and templates.

5.3 Study 3: Characterizing Case Study at Hughes

The second empirical study included in this work was carried out with Hughes Information Technology Corporation working under a contract for the U.S. National Aeronautics and Space Administration (NASA). The study took place in the Mission to Planet Earth (MTPE) program established by the National Aeronautics and Space Administration (NASA) to study Earth as an integrated and coupled system consisting of the atmosphere, oceans, and continents interacting through exchange of energy, mass, momentum on a wide range of spatial and temporal scales (Baker 1990). The commitment to make data and information resulting from MTPE easily available to users is critical to the success of the project. NASA has started to meet this commitment through incremental and evolutionary development of the Earth Observing System Data Information System (EOSDIS) with significant user involvement in all of its phases.

The pilot project selected for the case study is an example of an anticipated new type of software development projects within the EOS program. It is a project where several partners jointly develop a service, instead of single partner being responsible for all development work. The development work is also more user and feedback driven than projects that have been developed in the past. The experiences from this pilot project will be used to define the new development paradigm for other, future projects in the program.

5.3.1 Objectives, Design and Practical Arrangements

The original plan for the study was to use Riskit as a risk management process in one of the projects in the program and to obtain characterizing information about Riskit in industrial context. However, the program was halted and restructured due to U.S. congress decision to realign NASA's programs. Hence, our study was interrupted and we only obtained empirical data about the first steps of the Riskit process, i.e., of goal review and risk identification. Nevertheless, this data was useful in developing these aspects of the method further.

The relevant, "surviving" research objectives of the study were to analyze the goal review and risk identification steps of the Riskit process to characterize them. This goal is expressed in more detail in Table 54.

Analyze	The goal review and risk identification steps of the Riskit process
In order to	*Describe* and *understand* them
With respect to	Number and type of goals and stakeholders identified, Feasibility of goals analysis and documentation, Benefits and disadvantages of checklists.
From perspective of	Risk management method developer.
In the context of	NASA EOSDIS program.
Because	Practical feedback on the initial steps of Riskit will be used in the further development and in the documentation of the method

Table 50: GQM statement for the Hughes study

We wrote a detailed plan and schedule for the study, including training, definition of process steps, and listing of methods and techniques used in each phase. The sessions that were held were facilitated by J. Kontio and we used a scribe to support the documentation of

the sessions, in addition to the artifacts that were produced during the meeting. The session transcript included timestamps on all main events or issues.

5.3.2 Analysis and Results

The goal review produced five stakeholders and 17 different goals. Goals included typical items, such as schedule, functionality and user satisfaction, but also several less typical goals were listed, such as setting a technology showcase, leverage and deepen university industry cooperation, and evaluate the new development process. The EOSDIS program and the project in question have a very broad scope and affect several stakeholders, possibly explaining the large number of goals.

The goal review itself lasted approximately an hour, although some of the goals were added or refined during the risk identification session. In fact, risk identification seemed to characterize and clarify goals that otherwise might have remained abstract.

We made two main recommendations for tool development from the goal review session:

(f2.1) Functionality goals should be grouped into a single set and linked to requirements specification in order to reduce the number of goals to be tracked to a manageable level.

(f2.2) Goal revision should be a natural part of risk identification (and analysis) as later steps in the process will provide further insight to them.

The risk identification session was based on the use of open brainstorming using post-it notes, supplemented by focused brainstorming. The focused brainstorming part was based on stakeholders and goals, i.e., participants were asked to think about risks that might affect the listed goals or stakeholders. The risk identification session lasted 70 minutes, producing a total of 48 risks. We were able to track the origin of each risk, both with respect to who proposed it and in what part of the process it was identified. About three quarters of the risks were identified in individual brainstorming, about a fifth in focused brainstorming, and three additional risks were identified during the discussion.

We took the risk list that was produced and mapped it to the SEI risk taxonomy (Carr et al. 1993) in order to find out how well the taxonomy matches to the identified risks. Most risks were mapped to taxonomy well, but six risks did not have a matching category in the SEI risk list. These risks were considered to be among the most relevant ones in the project.

We made the following findings from the risk identification session:

(f2.3) Individual brainstorming supported by focused brainstorming produces a good coverage of risks.

(f2.4) The number of additional risks identified in the discussion is low but the new identified this way were considered important and were less intuitive.

(f2.5) Checklists that are not specific to the domain in question may fail to support the identification of relevant, situation specific risks.

In addition to these findings, the practical, hands-on experience in using the Riskit method provided several suggestions to improve the methods usability and characteristics. In particular, the goal review step was improved, risk identification guidelines revised, and the training material improved.

5.3.3 Study Conclusions

The Hughes study provided limited data and insight, due to the environmental changes that took place during the study, effectively halting the study before its completion. Nevertheless, the study provided several contributions to the development of Riskit. First, it helped us focus on some key parts of the method and improved the goal review and risk identification steps. In particular, risk identification can contribute to more accurate definition of project goals (f2.2) and it is therefore beneficial to maintain explicit links to project goals throughout the risk management process; and the system requirements can be handled as one main goal set, decomposed if necessary (f2.1). We also synthesized an approach where focused brainstorming and checklists are used together to cover a broad range of risks in a limited time (f2.3, f2.4, and f2.5).

Second, the practical work done in the study gave us hands-on experience in performing and facilitating risk management in practice. This practical experience was essential in planning and conducting later studies. Third, the empirical study design and arrangements acted as models in the later studies, making them more focused and effective.

Finally, even though the participant feedback was not formally collected, the method was evaluated and used in a large, leading edge, industrial project. The informal feedback obtained indicated that the project management would have been willing to continue using the method and considered it practical and beneficial.

5.4 Study 4: Nokia and DaimlerChrysler Study

The third empirical study reported in this work was a study that used information from two organizations and projects, both based on the same empirical study design. The main findings of this study have been published in a separate report (Kontio et al. 1998) and we present the main findings and provide some additional insights from the study in the following.

Both of the participating organizations had existing, relatively informal risk management practices in place prior to this study. Their earlier practices were analyzed through ethnographic techniques, i.e., spending time at the organization and studying the current documentation of their risk management process. The highlights of these baselines are described in Table 51

The DaimlerChrysler project was a business process re-engineering project that produced a diagnosis support system that will be distributed worldwide. The development involved both in-house development and the use of consultants, as well as in-house and commercial components. The project size was about 200 person years and duration three years. The Nokia case developed an embedded telecommunications product, involving well over 100 person years in less than two years. This project was an in-house development involving advanced technologies and tools in a new organization, as well as including both software and hardware development.

5.4.1 Study Goals

Goals for the study included both industrial goals for the participating companies, as well as research goals that were defined by the participating researchers. The industrial goals were

- Establish a risk management process for the case study.
- Perform the risk management process:
  - identify risks;
  - analyze and rank risks; and
  - propose controlling actions.
- Monitor and record the risk management process and risks.

The research objectives in the study focused on three main areas. First, to characterize the Riskit method, the benefits it might bring along, what disadvantages does it have, what problems users face when using it, and how the method could be improved. Second, we wanted to evaluate the feasibility and usefulness of the method in an industrial context. Third, we wanted to improve our understanding of how to introduce risk management practices into a project. These research goals were documented in fourteen GQM statements, listed below:

Goal 1: Characterize Riskit
Goal 2: Evaluate feasibility of Riskit
Goal 3: Evaluate effectiveness of Riskit
Goal 4: Characterize applicability of Riskit components in different situations
Goal 5: Monitor risk factors and events
Goal 6: Characterize stakeholder and goal concepts
Goal 7: Characterize risk control planning activity of Riskit
Goal 8: Characterize risk element information attribute usefulness

Goal 9: Characterize risk controlling action impact tracking
Goal 10: Compare risk management infrastructure
Goal 11: Compare risk description sheets and Riskit Analysis Graphs
Goal 12: Compare the DaimlerChrysler risk ranking matrix and Riskit Pareto efficient ranking technique
Goal 13: Compare ad hoc and focused risk identification processes
Goal 14: Characterize experiences from the risk management database

The relationships between these goals are presented in Figure 37 and Appendix D contains their full definitions. As the goals show, the study goals can be grouped in the three main groups, descriptive goals, evaluation goals, and comparison goals.

Each goal was decomposed into characterizing questions and a set of metrics was defined for each such question. Some metrics were interview questions, in which case they were defined is a questionnaire, some metrics were items that were to be measured during the process or counted from the data or artifacts produced during the process. Appendix D contains a full list of metrics that were defined for each goal. Metrics were grouped into instrumentation sets, and a cross-checking table was created to ensure that all metrics were collected and assigned to appropriate sets (see Appendix D).

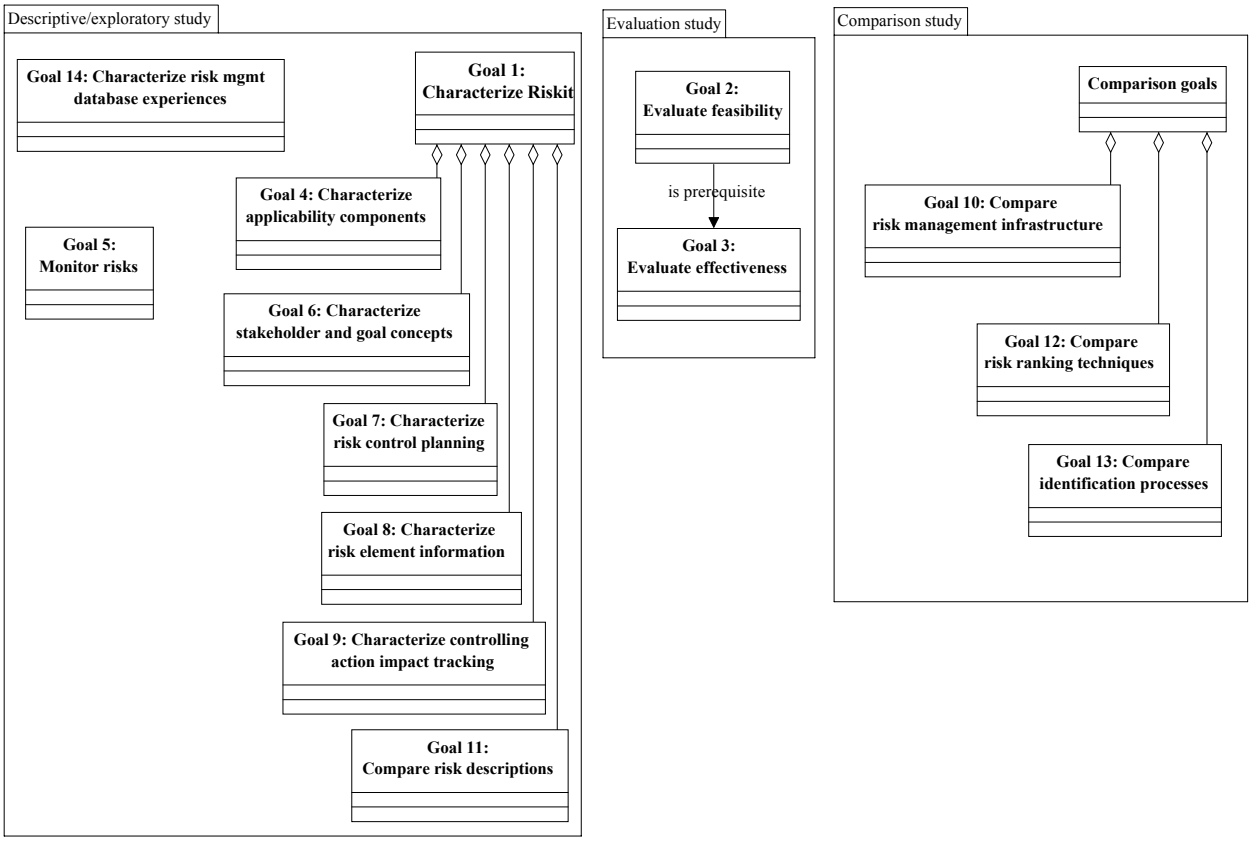


Figure 37: Structure of GQM goals in the Nokia DaimlerChrysler study

5.4.2 Study Design and Arrangements

The Riskit method was introduced to projects at an early phase of projects but not at their beginning. The Riskit method was used in slightly different way in the projects: at

DaimlerChrysler the method experts facilitated the sessions whereas at Nokia the project applied the method independently after an initial training and consulting period by the method developer.

Five forms of data collection were used in the case studies. First, the Riskit method itself produced extensive documentation about the risks and the risk management process that was followed. Our initial plan was to use the prototype eRiskit application to capture risk management information, but the maturity and reliability of the software did not allow its use in the highly constrained and critical application projects. Thus, the data collection was based on manual or electronic documentation.

Second, the risk management facilitators acted as observers in the risk management sessions and used this information as part of the analysis, taking notes and raising their observations in analysis sessions. This information was used to provide depth and context in the analysis of data, as well as to prompt observations in the sessions.

Third, a series of semi-structured interviews were performed to elicit participant feedback on the risk management process. The interview templates contained 83 open questions and they were used to structure the interview session and to provide consistent coverage in interviews (see Appendix D.3). In practice, interview sessions followed the interview template outlines, but additional information was often volunteered in various points in the interview. The interview data consisted of only three interviews: one at Nokia and two at DaimlerChrysler, and the two DaimlerChrysler interviewees were also the method facilitators who had been given the responsibility of conducting risk management in the project.

Fourth, DaimlerChrysler had written a lessons-learned report after the first risk management cycles in their projects, independently of the interview sessions held later. This report and its findings were included in the analysis.

Finally, in the Nokia case study we also used video recordings in the most critical sessions. This was done to avoid the potential observation bias by the method developer and to make sure that all relevant data was recorded. These recordings were analyzed to identify problems

	DaimlerChrysler	Nokia
Frequency	Risks listed weekly in every subproject	Risks were listed in monthly
Formality	Reporting at project meetings within status reports	Monthly reporting of top 5 risks required
Method and tools	Documentation only for project tracking	Risks listed in order of importance
Identification techniques	By team members without any specific methods or techniques	By program and project managers without any specific methods.
Analysis techniques	No specific analysis techniques	Ranking based on numerical estimate of probability and qualitative estimate of impact on schedule and quality
Controlling and tracking techniques	Part of normal project management	Part of normal project management
Training	No specific training for risk management	No specific training for risk management

Table 51: Previous risk management in the two organizations

in the communications and to provide more information on the notes taken during the meetings. Videoconferences were regularly used in this organization and some the risk management sessions did, in fact, take place between two continents.

Data from the case studies (participant observations, Riskit artifacts, interview notes and video recordings) were analyzed and relevant issues identified and highlighted. When an issue was highlighted, the experiences from the other case studies were compared to it and rationale and explanations were discussed.

Case studies are prone to many limitations, compared to situations where large amounts of data can be collected and analyzed (Simon 1969; Yin 1994). Studies in risk management, in particular, have even more serious constraints that limit the choice of experimental designs and available data points (Kontio & Basili 1997), as well as challenges in construct validity. In particular, low number of data points, their non-random selection, and variance in situational characteristics limit the external validity of the results obtained, i.e., their generalizability.

Our case studies tried to limit the internal validity threats associated with the descriptive part of our study by documenting and using raw data from the study and recording the interview data as objectively as possible. We tried to provide a better basis for controlling external validity threats by explicitly documenting the situational characteristics of the cases, as well as replicating the study in two different organizations. However, the replication benefits were limited due to low process fidelity (Feiler & Humphrey 1993), as both organizations made modifications to original method.

Despite these limitations, we believe that our study produced data that has reasonably high internal validity and there are no major threats to the external validity of the results.

	DaimlerChrysler	**Nokia**
Scope of applying the method	Riskit process steps followed, Riskit Analysis Graphs used for most complex risks, different ranking technique used	Riskit process steps followed, Riskit Analysis Graphs used for key risks, Riskit ranking approach used
Way of applying the method	Sessions facilitated by a Riskit expert	Independent use (initial sessions facilitated by a Riskit author)
Training given on risk management	1 hour for project management, 1 hour in each subproject	Self study Two hour private session to project manager One-hour session to project management team
Number of risks identified and *documented*	30 at project level, up to 130 at subproject level	150
Number of risks controlled	10-20 on project level, up to 20 (clustered) at subproject level	ca. 70

Table 52: Characteristics of risk management processes

5.4.3 Analysis and Results

5.4.3.1 Introducing Risk Management

The Riskit method was introduced and applied in slightly different ways in the two organizations, as shown Table 52. At Nokia, the Riskit method was introduced to a product development program when the program was already running at full speed. Therefore, only minimal additional training was possible on risk management. The program defined a formal risk management process and included it in the program management procedures. However, the program manager reported that there were problems with process fidelity in practice. The introduction of Riskit included making the Riskit documentation, drawing tools, and templates available. General training on risk management and on the Riskit method were given to the program management team in two single sessions. In addition, individual sessions were also given to key members of the management team.

At DaimlerChrysler the method was introduced and supported by two more experienced risk management experts that facilitated the risk management sessions in the project. They provided the training and defined project-specific conventions for risk management.

5.4.3.2 Risk Management Mandate

At Nokia, the risk management mandate was explicitly defined. The mandate provided better and unambiguous definition of the responsibilities and scope of risk management, compared to the situation before, and thus contributed to more explicit risk management practices in the project. The recognition of stakeholders clarified expectations and made the prioritization of goals easier, according to the program manager. However, the positions of some recognized stakeholders were not explicitly stated and this was a cause of concern to the project management.

At DaimlerChrysler there was no formal risk management mandate definition, although some aspects of the mandate were defined. In particular, the stakeholders were not defined. Project participants did not see any added value in spending time to analyze stakeholders that were already identified, although less formally. However, it was also observed that different project participants had different interpretations as to who are the relevant stakeholders and what their priority should be. We believe that part of the participants' resistance to stakeholder analysis is caused by the smaller amount of training given at DaimlerChrysler, as Table 52 indicates. Project participants may not have been aware of the rationale and benefits of stakeholder analysis.

These experiences indicated the following findings:

- *(f3.1) An explicit risk definition of risk management mandate seems to clarify the responsibilities and scope of risk management.*
- *(f3.2) In order for stakeholder recognition to take place, participants need to be trained and motivated.*
- *(f3.3) Without explicit stakeholder analysis, participants are likely to have different interpretations of project's stakeholders and their preferences.*
- *(f3.4) Stakeholder information helps to clarify and prioritize expectations and goals for the project.*

5.4.3.3 Goal review

At DaimlerChrysler the goal definition and goal review were based on project documentation, no specific goal analysis sessions were held with project personnel. The goals were used to analyze risk effects, i.e., used in risk prioritization. As with the stakeholder analysis, project participants were not interested in discussing or re-analyzing goals, although different interpretations of goals were observed.

Some project participants expressed a concern that some goals are not well suited for open and explicit discussion ("*We will get problems [if] we are discussing [the] goals and write them down*"). We believe this is partially a cultural issue related to how openly goals are generally communicated, and partially a natural tendency of individuals to avoid over-commitment.

At Nokia there was an explicit goal review phase. This resulted raised several questions on the priority of the goals with respect to different stakeholders. This lead to re-definition and re-prioritization of some goals by the executive management. According to program management, the goal review raised the general awareness of program goals and their priority and helped understand the importance of some key constraints of the program, giving program management more flexibility and better focus. The Riskit approach seemed to help focus discussions, clarify concepts and points of view.

People participating in the goal review sessions initially had some motivation problems, they were not sure why a goal review is necessary in the project. This was probably due to the limited amount of training and motivation given to participants, since the goal review resulted in major changes in goals. These experiences lead us to propose the following tentative conclusions:

(f3.5) An explicit goal review is provided useful input to project management in general.

(f3.6) Goal review requires motivation and training, as well as a right climate and attitude to achieve open and complete analysis of goals.

5.4.3.4 Risk Identification

Different techniques were used for risk identification in both organizations: interviews, brainstorming and checklists. At DaimlerChrysler the main technique was structured interviews. Riskit concepts were used to structure the interviews in which project members were asked about stakeholders, goals, risks, and risk scenarios. Some subprojects wished to identify risks in workshops to optimize time for identification and analysis of risks. The free-format risk identification was supplemented by interviews. Checklists (Carr et al. 1993) were used to guide the workshops and a questionnaire was used to verify brainstorming results more analytically. The information gained in interviews seemed to be more detailed and of better quality than the results of workshops, perhaps due to more confidential nature of interviews and the possibility to focus on specific topics. The additional yield of using checklists in risk identification was small, as the checklist did not match the domain of the project quite well.

According to DaimlerChrysler experiences, the disadvantages of interviews are the large amount of information (risks have to be clustered) and the added time needed to achieve an agreement of the whole group.

In both organizations, the raw risk data from initial identification sessions was clustered into groups based on some project-specific attributes. These clusters allowed better communication and filtering of risks for more detailed analysis. The clustering criteria varied between projects.

At Nokia, the risk identification had already been done previously in the program. A total of 60 risks had been identified and documented initially by individual sub-project managers, and during the program additional 90 risks were explicitly documented. The risk identification approach was an informal one and deviated from the Riskit process. The risk analysis in the program was done both at the project level by project managers and at the program level. The program level risk management was based on consolidating the project risks *and* evaluating risks from program perspective. The program manager expressed some concern about the coverage of the risk identification approach that was used: some risks that occurred were not identified in the identification process.

Participants reported that separate risk identification sessions helped them think about risks proactively (instead of recognizing problems that are already present), consider long-term risks, and consider risk information from various sources.

Based on these experiences we suggest the following tentative conclusions:

(f3.7) It is difficult to ensure adequate coverage of risk identification without explicit risk identification techniques.

(f3.8) Checklists do not seem to yield many additional risks when used after free-format brainstorming sessions.

(f3.9) Project personnel are under constant time pressure and without enforcing explicit risk identification sessions they may not spend sufficient time in risk identification after the initial risk management cycle.

We also noted that generic risk management checklists might bias the risk identification, unless they represent the domain and project characteristics accurately.

5.4.3.5 Risk Analysis

At Nokia, the Riskit key concepts were used informally at the subproject level but at the program level the main risks and risk scenarios were explicitly documented using the Riskit Analysis Graphs and ranked using the Pareto ranking table approach. The risk scenarios seemed to help analyze risks in more detail, but due to limited training given, they remained distant and theoretical for many participants.

At DaimlerChrysler the Riskit risk scenarios (documenting risk factors, risk events, and risk effects explicitly) seemed to result in deeper and unambiguous understanding of risks. However, our experience indicates that normally it is difficult to obtain the necessary detailed information for completing the risk scenarios. Risk scenarios were sometimes left incomplete or they were too abstract to be of practical value. The DaimlerChrysler experiences also indicated that developing risk scenarios requires more training and practice than was given in that case study. However, risk scenarios seemed to improve the transparency and understandability of risks, as well as increasing participants' confidence in the results. DaimlerChrysler used risk information sheets to document the main information about risks in the process and these sheets became a central communication mechanism for the participants.

DaimlerChrysler did not use a Riskit based prioritization approach. Instead, they used two sets of risk ranking grids that were based on two-dimensional tables that ranked risk scenarios using probability, impact, urgency, and level of uncertainty. Risk scenarios were developed for risks that had high levels of uncertainty. Although there are some potential theoretical limitations with this approach, participants were satisfied with the approach and it was used consistently in the project.

Based on these experiences we suggest the following findings:

(f3.10) Riskit scenarios are perceived complex, at least when a minimal amount of training has been given to practitioners.

(f3.11) Riskit scenarios require training and facilitation before they can be used independently by project personnel.

(f3.12) Practitioners are satisfied using simple, straightforward techniques in risk management, despite their potential, theoretical limitations.

5.4.3.6 General Observations

Overall the confidence of program participants on the risk management results increased with the Riskit-based risk management approach. There was a major shift in risk management thinking: earlier, risk identification was the main focus, now the risk controlling action received more focus and attention. This was not only supported by the risk management process but also by the templates that clearly guided the risk analysis towards risk controlling actions.

The systematic risk analysis also resulted in revised risk priorities. Based on the analysis of Nokia data, people's intuitive risk rankings were different from the rankings produced by the systematic risk ranking technique used in the Riskit process.

At Nokia, the use of Riskit seemed to increase the level of confidence in risk management results whereas at DaimlerChrysler changes in confidence levels were not reported. The Riskit method provided a conceptual framework that helped and structured discussions about risks. The more detailed documentation of risks has allowed the organization to accumulate risk management experience and localized checklists based on actual risk history data are currently being developed.

Thus, additional findings can be summed up as follows:

(f3.13) The Riskit method seemed to encourage proactive risk management attitudes in the projects overall.

(f3.14) Intuitive risk rankings seem to differ from rankings derived using the Riskit method.

(f3.15) The use of systematic risk analysis methods seemed to increase participants' confidence levels in the results of risk analysis.

5.4.4 Study Conclusions

This study was based on a detailed, a priori plan and measurement. In reality, as both cases were performed in time-critical projects, the rigor of data collection and number of people participating in explicit risk management both suffered. While the study has strengths in being performed in real, industrial projects in two organizations, the number of data points is very low and this limits the generalizeability of results.

The overall conclusion from the study is that the Riskit method is a feasible approach for risk management in industrial context. It can be applied with reasonable initial training and it helps in performing risk management in software projects. However, the following additional, tentative conclusions can also be drawn from the data of the study.

Both cases in the study highlighted a common difficulty in risk management: it is difficult to make sure that the project organization consistently performs risk management (f3.1, f3.6, f3.9, f3.10, f3.11). In particular, we observed a tendency to omit risk management towards the end of projects. We suggest that this trend can be avoided by improved training and support and by enforcing risk management consistently. Training for risk management should be given to all key project personnel so that they are fluent in risk management concepts and techniques. Our experience indicates that one or two hour training is not adequate but a half-day training with facilitated initial stage cycles may be sufficient.

Regardless of the risk management approach used, it seems to be important to start the risk management activities as early as possible (f3.2, f3.4, f3.5, f3.13). Early use of risk management should help control some risks better and the early introduction of the method is less likely to meet initial resistance by the project personnel.

It also seems that different risks require different documentation (f3.10). Some risks are clear and obvious when brief, informal description about them is given. Our case studies showed that it is not practical to document all risk scenarios with elaborate definitions and graphs. Sometimes it is not even possible to get all of the necessary information for Riskit Analysis Graphs. Riskit Analysis Graphs should be used when there is no consensus understanding on a risk or when there are significant uncertainties involved with risk. This will help clarify fuzzy areas and pinpoint the remaining uncertainties in a project.

Stakeholders and goals seem to play a critical role in risk management (f3.3, f3.4, f3.5). The importance of stakeholders and their expectations was clearly demonstrated in the case studies: different participants had different understanding of stakeholders, their expectations, and their priorities. Explicitly recognizing them will ensure that all relevant risk areas are better covered and the program can focus on essentials in their risk management.

We believe that a common risk management framework makes risk management efficient. Risk is a fuzzy concept term and it can mean different things to many people. The use of Riskit Analysis Graphs helped communications in some situations, but even when the risks were not documented graphically, the underlying concepts helped participants understand and communicate what aspect of risk was being discussed. This also allowed better delegation of risk management responsibilities and easier consolidation of such results.

The Riskit process seemed to increase confidence in risk analysis results (f3.14, f3.15). Based on our interviews, the explicit documentation of risks and the systematic risk ranking approach used provided participants full transparency to the risk analysis and its rationale, and they understood and trusted the analysis results better than they had done before.

We also observed that an intuitive risk management produces different results compared to systematic, explicit risk management process (f3.14). There were many instances where the initial, intuitive perceptions of risks were significantly changed during the risk management process. We believe that the additional time spent on risk management as well as the methods used result in better understanding of risks and more appropriate risk controlling actions.

Risk identification requires special attention and a different mindset from other project and risk management activities (f3.7). Risk identification requires an open mind and the ability to look beyond the obvious. While most of the other management tasks in a project may rely on analytical thinking, risk identification requires the ability to innovate. Therefore, risk identification sessions should be planned and supported to ensure adequate coverage of potential risks.

Additional benefit for the researchers from this study was that the metrics defined were used to validate the eRiskit database content, i.e., to ensure that the database contained the necessary fields to capture empirical data required by this study.

5.5 Study 5: Method Introduction Study with IESE and Tenovis

This chapter presents the results from a study that was performed by Fraunhofer IESE[22] to introduce the risk management process into Tenovis[23]. IESE was contracted to provide the risk management competence and technology to Tenovis, as Tenovis recognized that their new business situation and technology base required improved risk management for their projects. IESE and author of this work, in turn, cooperated in the adaptation and introduction of the Riskit method for this purpose.

The study was conducted in a project that developed a tool supporting the administration of Tenovis' existing product platforms. The project was started at the end of 1999 and it was planned to last approximately one year. The project introduced several new technologies into Tenovis, e.g., World-Wide Web technology was used in a client-server application, and object oriented technology for design and implementation. In addition, a new development processes and a new project organization was introduced, and the project involved teams in three different physical locations.

The study described in this chapter was conducted by IESE and Tenovis, my[24] involvement was limited to (i) providing and adapting the Riskit method and related risk management expertise to IESE and Tenovis, (ii) advising on the introduction and use of the method in the study, (iii) providing examples and experiences from earlier empirical studies to act as a basis for the empirical design IESE constructed, and (iv) contributing to the analysis of the study results. A joint paper was written and published based on the study (Freimut et al. 2001).

5.5.1 Objectives, Design and Practical arrangements

The IESE/Tenovis study had two main objectives. On one hand, the study aimed at characterizing the Riskit method, risk management process in general, and the technology transfer aspect of introducing risk management process into an organization. The second goal was to understand the cost and benefit aspects of the method, i.e., whether the investment and overhead in risk management pays off for a project. Both of these goals are presented in GQM format in Table 53.

IESE refined these goals into specific metrics using the GQM method (van Solingen & Berghout 1999). The metrics so defined included traditional, quantitative data, such as effort spent on risk management and number of risks identified, as well as qualitative metrics, such as perception interview questions from the participants. IESE defined a questionnaire, partially based on the questionnaire used in the DaimlerChrysler and Nokia study (Kontio et al. 1998), containing 33 questions. In addition, observation was used as an additional technique for capturing data during the process.

---

[22] IESE stands for "Institut Experimentelles Software Engineering" and it is a part of the Fraunhofer Gesellschaft, largest funding organization for applied research and technology transfer in Germany. Fraunhofer Gesellschaft has 9600 employees and IESE employs c. 100 people. http://www.iese.fhg.de/

[23] Tenovis is a former Private Network division of Bosch Telecom GmbH and is currently owned by a U.S. private equity firm. It has 7000 employees and is developing products for the telecommunication sector. http://www.tenovis.com/

[24] Note that in other parts of this document we have used the pronoun "we" to refer describe whatthe author has done. As this study was conducted by IESE, in this chapter we intentionally use the pronouns "they" when referring to IESE personnel, "we" when both the author of this dissertation and IESE personnel were involved, and "I" when the author of this dissertation performed something.

Analyze	The Riskit method, risk management in general, and the transfer of risk management process
In order to	*Understand* them
With respect to	Usefulness and adequacy, including advantages and drawbacks
From perspective of	Risk management participants and Riskit developers
In the context of	Tenovis project
Because	Further information on the usefulness and adequacy provides empirical validation and feedback for method development.

Analyze	The Riskit method
In order to	*Characterize* it
With respect to	Cost-effectiveness
From perspective of	Management and Riskit developers
In the context of	Tenovis project
Because	Cost-effectiveness is a major potential hurdle in the introduction and use of risk management methods.

Table 53: GQM statements for the IESE/Tenovis study

We identified several potential threats to the validity (Freimut et al. 2001; Judd et al. 1991; Wohlin et al. 1999) of the study and we briefly summarize the steps we took to control these threats.

The *reliability* of the data collection (i.e., its consistency and repeatability) was improved by documenting the interview questions and protocols in detail and applying them consistently, and by collecting the facilitator observations soon after each session.

We identified two main threats to internal validity: experimenter expectation bias and maturation. The potential *experimenter expectation bias* present in this study, i.e., technology providers' expectations or desire to see positive results in a study (Katzer et al. 1991), was reduced by carefully discussing and evaluating the facilitator observations and findings and emphasizing the Tenovis participant feedback on them.

Maturation effect threatens the conclusions of a study when subjects react differently as time passes. In this study this could have been possible as the participants were just going up their learning curve on risk management and thus became more fluent in their activities over time. However, data collection took place at the end of the project in a short period of time when the participants were quite mature in their risk management practices. Therefore, we believe that the maturation effect did not significantly affect our study.

The representativeness of the project and its participants relates to how well we can generalize the results, i.e., to external validity. The project itself was more risky and had perhaps higher expectation levels than normal projects in the company. We believe that this had two impacts: on one hand, this may have biased the participants to recognize the need for risk management more clearly, resulting in a generally positive attitude towards risk management. On the other hand, the pressures of aggressive goals may have also reduced the time available for risk management activities by simultaneously increasing the expectations from risk management results. This could have resulted in a more negative attitude towards the impact of risk management on the project. Regarding the representativeness of the project

participants, we have no specific reason or information to believe that the participants would be different from those of other projects. We did interview all participants that were involved in risk management.

5.5.2 Analysis and Results

IESE analyzed and structured the study data, i.e., interview answers and facilitator observations, into six groups to structure the data analysis. The groups were risk identification and representation; cost/benefit evaluation; documentation; controlling actions and monitoring; commitment to risk management; and participation to risk management. They reviewed the data and listed problems, as well as recommendations for improvement. Additionally, the IESE experts and I arranged a session where data was reviewed and discussed and potential explanations or recommendations drafted. IESE presented findings to project participants for review, and conclusions, representing the joint view of both industry participants and researchers, were documented. The main findings of the study are described in the following (Freimut et al. 2001).

The "full operational definition" of the Riskit method was perceived as a useful and helpful aspect of the method. It enabled systematic identification, analysis, and tracking of risks and prompted participants to perform the necessary risk management activities, leading us to suggest that

(f4.1) the Riskit process supports systematic and focused risk management practice in projects.

The risk identification approach of combining brainstorming and checklists was perceived as systematic and comprehensive. This was further supported by having people with different backgrounds and competence areas participate in risk identification, i.e.,

(f4.2) the Riskit risk identification approach is perceived as systematic and comprehensive.

The Riskit Analysis Graphs were perceived as helpful in understanding the risks and their consequences in their context. Project participants perceived the time spent on developing risk scenarios acceptable but the IESE facilitators were concerned about the fact that only a few scenarios per session were documented. It is interesting to note that the average time to complete a scenario in the study was 17 minutes. This does not appear as a long time to spend on understanding a scenario properly, we propose that due to practical meeting time constraints people perceive this time to be too long, but from the project perspective it does not appear as too labor-intensive task. Given this, we conclude that

(f4.3) the Riskit Analysis Graphs were perceived as helpful in documenting and analyzing risks, and

(f4.4) the Riskit Analysis Graphs are sufficiently fast to allow their use in meetings to model risks.

The Riskit Pareto ranking technique was perceived as beneficial and practical. Respondents especially appreciated that the ranking was possible without precise numerical data. Given this, we suggest that

(f4.5) the Riskit Pareto ranking technique is a practical and useful approach in risk prioritization.

The documentation of risks was done in forms as shown in Figure 38, i.e., the form contains several textual and numeric fields in addition to the Riskit Analysis Graph. The

form was designed to support both operational risk management in the project, as well as the learning and research objectives of this study. I had prepared Microsoft Word templates for the creation of such forms, including simple macro functionality to support the editing work.

Tenovis Risk Scenario Form

ID 1-1 poor quality code –review/tutoring	**Project:** Tool Harmonization	
Owner/Responsible:	**Date reviewed**: 2000-02-01	
Timeframe:	**Priority**: Controlled	**Probability**: 2
Stakeholder: Tenovis Mgmt	**Loss**: 3	
Stakeholder: Dept. Lead	**Loss**: 4	
Stakeholder: Project Leader	**Loss**: 4	

Event description	"poor qual." The missing experience of the development team with Java leads to poor quality code (i.e., buggy, not efficient)

Factors:
Not much Java experience

Selected scenario				
	X	**Reaction:** Code reviews & tutoring	**Effect set**	**Schedule**: Miss deadline a little **Cost**: go way up **Technology** Tech showcase not successful
		Reaction: (Deliver poor quality on time)	**Effect set**	Customers unhappy
		Reaction:	**Effect set**	

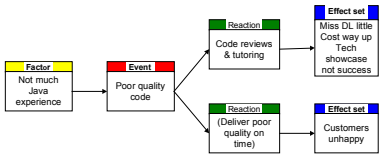
Potential risk controlling actions	Improve review practice, organize tutoring – log results, define metrics and thresholds, reduce time pressure and communicate importance of quality

Selected risk controlling actions

	Action	*Respons.*	*State*	*Finish*	*Check*
1	Introduce review process	Smith	done	end May	✓
6	Perform review process	Miller	ongoing	End Proj.	Oct.
2	Monitor the results of training	Miller	Ongoing	End Proj.	Sept.
3	Develop coding guidelines	Architects	Ongoing	End Proj.	Oct.
4	Evaluate Java code checkers	Doe	done	mid July	✓
5	Communicate importance of quality	Miller	done	end Sep	✓

Closing date:	**Closing Rationale**:

Graphical representation of scenario



Risk History:

31.3.00:	risk probably smaller, because people are aware of the importance of quality as a result of controlling action 5
17.5.00:	risk is still to be considered; there are new people within the project; and there will be new people coming within near future
29.5.00:	risk unchanged; controlling actions sufficient
13.6.00:	Re-ranking changed probability from 3 to 2, new utility loss for project leader assessed
30.6.00:	risk unchanged; controlling actions sufficient
25.8.00:	action 4 was stopped since an evaluation of a Code Checker was completed action 5 was stopped because this is an integral part of the tasks of the project leader and line managers

Controlling Action History

Controlling Action	Impact
1	29.5.00: introduced; impact: see follow-up controlling action 6
2	31.3.00: minor 29.5.00: reveals the effectiveness of training 30.6.00: currently no training 25.08.00 effect positive as people do build up know how; through the monitoring the need for additional training has been detected

Source: (Freimut et al. 2001)

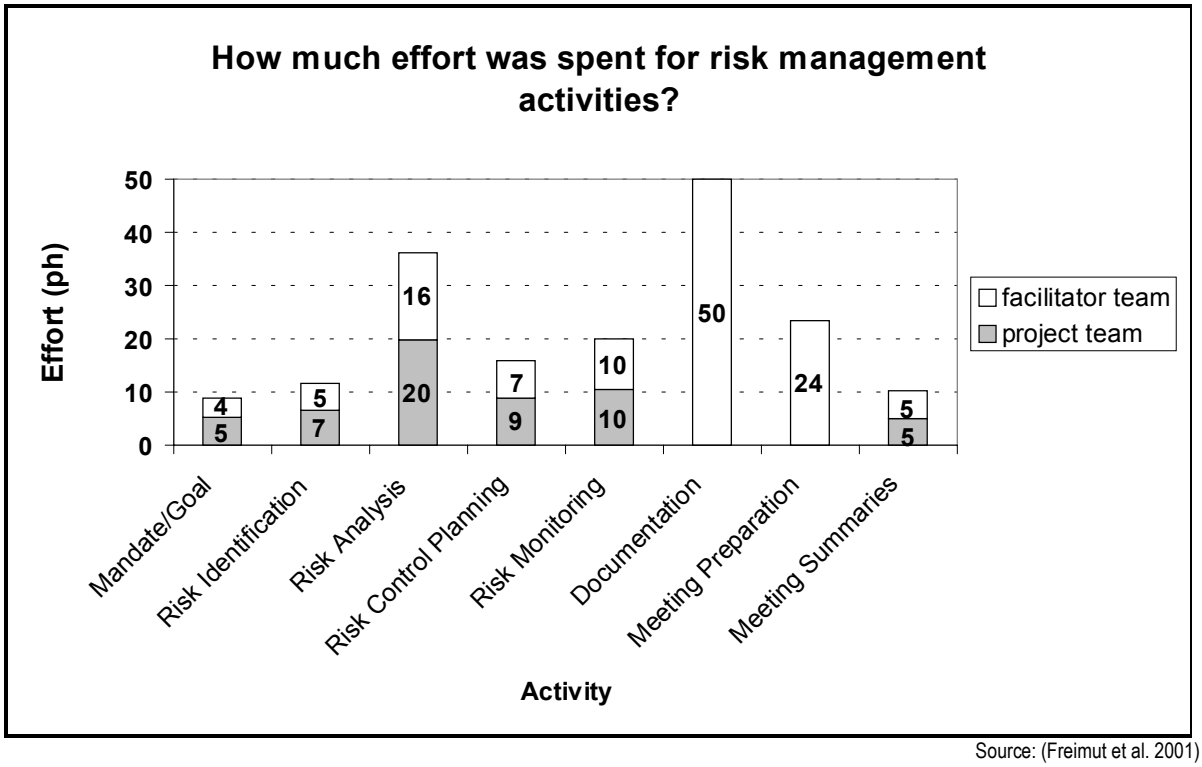
Figure 38: Risk Scenario Form for one risk event

Participants expressed three disadvantages related to forms. First, the forms contained too much detail for daily work. Second, the textual descriptions were kept too short, which made it difficult for other than originators of the form to understand what was really meant by the risk. Third, the effort spent on documentation was perceived to be too high: it represented 28% of the total effort for risk management and was the largest single task in risk management. Figure 39 presents the distribution of effort to different tasks.

The documentation of risk information was done manually in this study. The simple macros in the templates merely automated some basic editing tasks, the templates themselves were not based on an underlying database, and the administration of simple forms became difficult and time-consuming. We believe that appropriate software support could substantially reduce the amount of effort required. The effort savings would most likely result from the following main areas:

- Entry of redundant information is eliminated.
- Retrieval and management of data becomes faster.
- Amount of errors in data is reduced, reducing the time required to search and correct such errors.
- Links and dependencies between items can be more effectively kept up-to-date.

- Archiving and versioning of data can be automated.



Source: (Freimut et al. 2001)

Figure 39: Effort spent on risk management

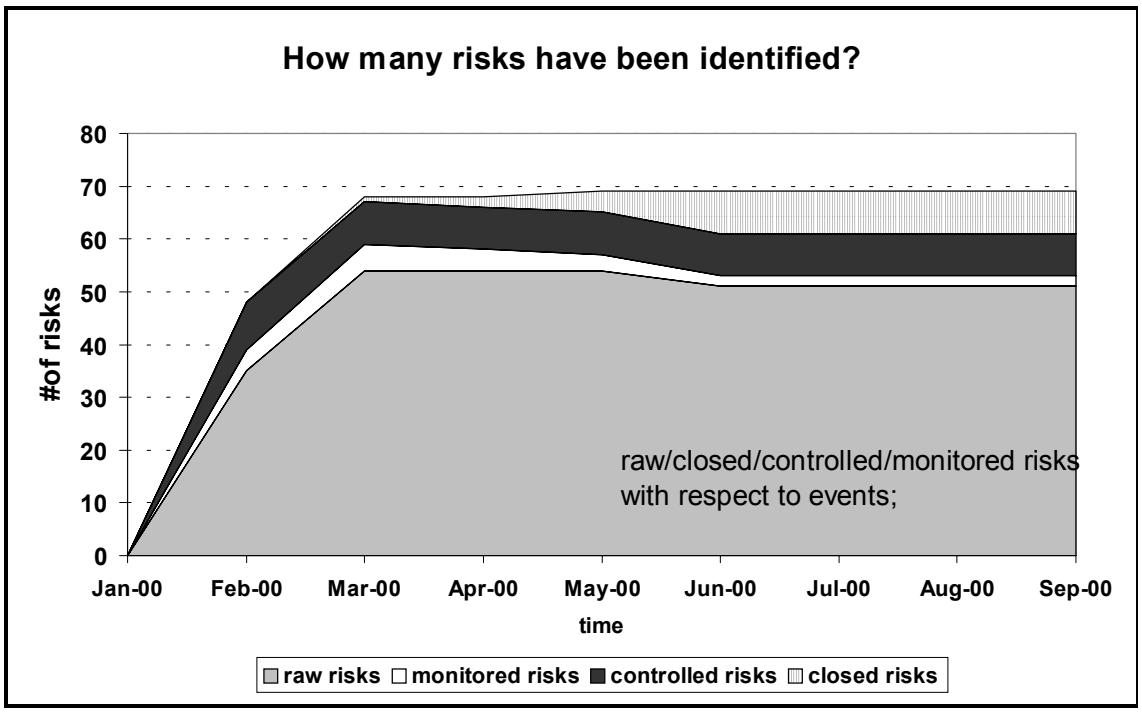
Naturally, software support for risk management offers other potential benefits as well, such as sharing of information, automated checking for consistency, retrieval, and analysis of data for research and improvement purposes, and providing process support and guidance to users.

Given these experiences, we propose the following findings on documenting risks:

(f4.6) Manual, form-based documentation is time-consuming, yet the information contained within them may be difficult to interpret unambiguously.

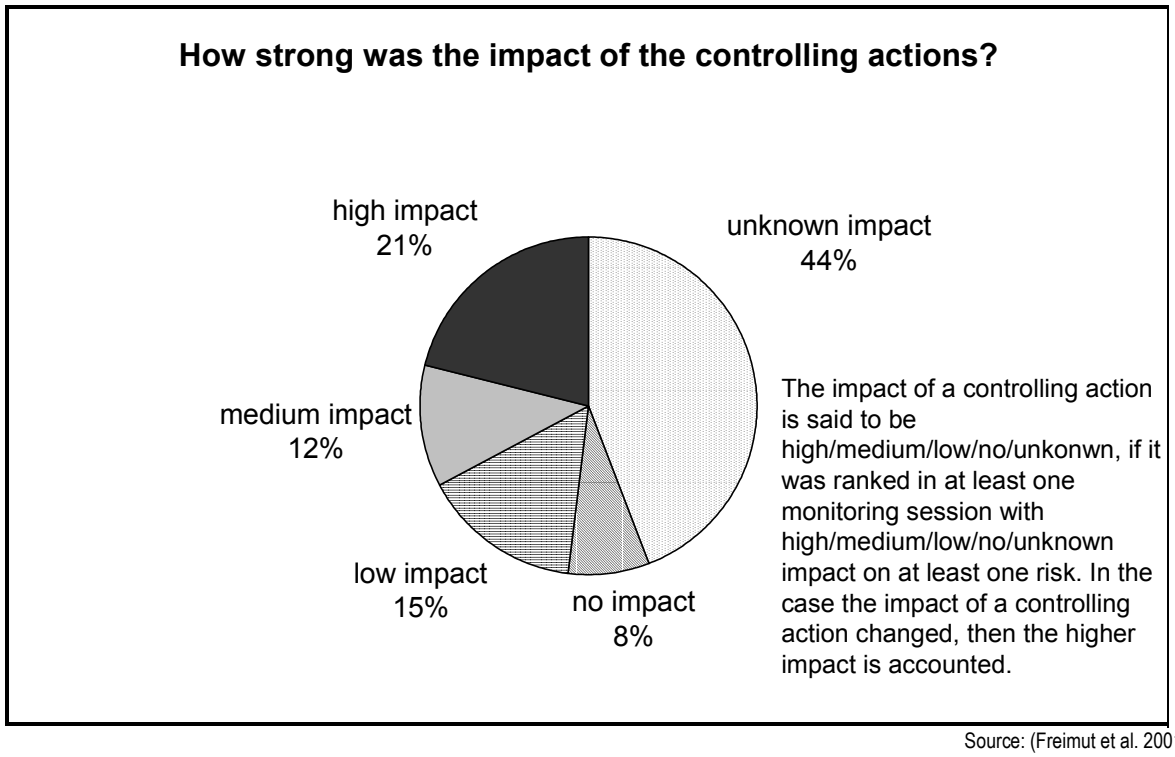
(f4.7) Software support for the risk documentation can result in significant effort savings and improved data accuracy.

The cost-efficiency of Riskit was also evaluated in the study. The cost of risk management, shown in detail in Figure 39, was 5% of the total effort for project management. The impact of risk management was evaluated by categorizing risks into raw, monitored, controlled, and closed risks. Raw risks were items that were identified in risk identification; some of them were categorized into monitored or controlled ones, based on risk analysis; and closed risks were ones that were no longer needed to be controlled or monitored, either due to effective risk controlling action or due to new risk analysis. Their volumes evolved over time as shown in Figure 40. The impact of risk management actions was assessed by asking participants to assess controlling actions' impact over the course of the study. Figure 41 shows the summary of this data.



Source: (Freimut et al. 2001)

Figure 40: Number of Risks



Source: (Freimut et al. 2001)

Figure 41: Impact of controlling actions

The data in Figure 41 has two interesting characteristics. First, about one fifth of the controlling actions were considered to have had a high impact on risks, i.e., these actions influenced projects risks and can be considered effective in reducing projects risks. Even if we were to include the "medium impact" controlling actions, only 33% of controlling actions had medium or high impact, as estimated by the participants. We suggest that this is potentially too low a figure, subjectively we feel that actions that are implemented should have a better success rate and bigger impact on project risks. Second, participants reported that the controlling actions were not performed as planned and, therefore, were not as effective as they could have been. This might explain the low impact figures. This may also be due to poor questioning of controlling actions during risk monitoring: the impact of 44% was unknown.

The participants also were asked their overall opinion on the impact of risk management in the project. They favored the sound and systematic approach that the risk management process established. The participants also considered the overall effort spent on risk management acceptable, but considered the impact of risk controlling actions too low. Overall, these points suggest the following conclusions:

(f4.8) The Riskit process consumed only a small share (5%) of the management time in the project.

(f4.9) The effective implementation of risk controlling actions is a critical success factor for risk management.

(f4.10) Riskit was perceived as a sound and systematic process.

Finally, the IESE/Tenovis study also produced some findings about introducing risk management in general, as listed below:

(f4.11) Risk management activities should be closely integrated with other project management activities.

(f4.12) Risk identification should be repeated over the course of the project as situations change and new information becomes available.

(f4.13) Monitoring of risks' status should be done frequently.

(f4.14) Sufficient emphasis should be placed on ensuring that risk controlling actions are implemented as planned.

(f4.15) Sufficient training should be given to people participating in risk management so that they can participate efficiently.

(f4.16) Key members of the project should commit to performing risk management. Especially project manager's commitment is essential.

These findings were considered general ones that are not necessarily tied to the Riskit method, i.e., they can be considered more generic guidelines for applying risk management in projects.

5.5.3 Study Conclusion

This study represents one of the most detailed empirical studies of risk management in an industrial context. It produced several concrete recommendations for the method and tool development, and helped characterize the process and problems associated with introducing risk management into software projects. Overall, the systematic approach of Riskit was valued (f4.1 and 4.10), the Riskit Analysis Graphs were found usable (f4.3 and f4.4), and the Riskit Pareto ranking technique was considered beneficial (f4.5). The overhead spent on

documentation (f4.6) and ineffective risk controlling actions (f4.7) were highlighted as the main problems. However, both of these problems are relatively easy to avoid.

The Riskit process was perceived efficient (f4.8) in terms of effort spent on risk management activities, and participants valued the soundness of the method (f4.10). The study also produced several more general findings about risk management (f4.11 – f4.16) that seem to be applicable to risk management in general.

The risk management in this study was strongly facilitated by IESE experts. It allowed efficient execution of the study and risk management in the project. However, it may have also changed the way project personnel might have preferred or been able to perform risk management on their own. Facilitated risk management is not a reasonable option for all projects, eventually; projects should be able to perform risk management independently.

5.6 Study 6: Risk Information Documentation Study with Students

This chapter presents the objectives and results of a study that was conducted with students at Helsinki University of Technology to evaluate and compare modeling of risks using the Riskit Analysis Graphs. The purpose of the study was to compare three different implementation techniques of using of Riskit Analysis Graphs (discussed in chapter 3.5), use of forms to capture risk information, and the SEI risk statements (presented in chapter 2.3 (Dorofee et al. 1996)).

5.6.1 Objectives, Design and Practical Arrangements

The motivation for the study was two-fold. First, we wanted to compare how the three main approaches for documenting risks – the Riskit Analysis Graphs, risk forms, and the SEI risk statements – compared to each other. Second, we also wanted to study which implementation technique for the Riskit Analysis Graphs – use of blank whiteboard, use of software, or use of laminated risk element symbols – is most effective.

Analyze	Five different methods for modeling risks in a risk analysis meeting, using three Riskit Analysis Graph –based techniques: (1) software with video projection, (2) blank whiteboard, and (3) laminated Riskit Analysis Graph frames; (4) using forms to capture risk information, and (5) using the SEI Risk statements.
In order to	evaluate their effectiveness
With respect to	time to use; information produced during discussion vs. captured; number of unclear issues; amount of time spent discussing graphs sheets vs. general discussion; amount of time per scenario; number of elements in scenarios; number of goals in effect sets; number of factors per scenario; number of revisions in the scenarios; amount of time spent on revising documentation; user perceptions on usability, ease of use, and effectiveness; recommendations for further use; and improvement suggestions.
From perspective of	Risk management method developer or process owner.
In the context of	University course "Tik-76.115 Software Project"
Because	Effectiveness and methods for risk documentation in a meeting influences how much information is captured and how well it will be stored for further analysis. It is also likely to affect participants' motivation to participate in risk management.

Table 54: GQM statement for risk documentation study

The use of blank flip charts was based on a manual use of the Riskit Analysis Graphs, i.e., session participants were expected to draw Riskit Analysis Graphs symbols, i.e., simple rectangles, using freehand, on blank flip chart sheets. The RiskitFrames are laminated Riskit Analysis Graph elements, which can be written upon with water-soluble pens and wiped clean. They have magnetic strips or adhesive sticky tape on the back so that they can be attached and reattached to most surfaces in meeting rooms. In this study, we used a blank, metallic whiteboard to "host" the risk scenarios. The software tool was a VISIO software

package that had Risk Analysis Graph symbols implemented in a template, allowing "drag-and-drop" editing of the risk elements in a scenario.

Metrics	Description	Measurement
Elapsed time to complete a scenario	The time it takes to create a Riskit scenario for each scenario	Measured with a stopwatch from videotape, elapsed minutes
General discussion time in a scenario	Time during scenario discussion when the discussion strayed away from the scenario at hand.	Time within elapsed time that was spent on general discussion
Time spent on method questions	Time spent on discussing "what should we do now"	Measured from videotape
Effective time	Elapsed time minus the time spent on method questions and general discussion	Measured from videotape
Number of elements in scenario	Number of Riskit Analysis Graph elements that are produced in each scenario	Counted from produced risk scenarios (not done for SEI risk statements)
Time per element	The average time spent per Riskit Analysis Graph element	Number of elements / Effective time (Not done for SEI risk statements)
Information captured / information produced	Average of the amount of the information that is captured on risk scenario divided by the risk information produced (what the risk means, its effects, etc.). The "amount" of produced information is calculated by counting the "points" made during the discussion including the points that were forgotten to add to scenarios and the points that do not have a placeholder in them, but not including the points that were deliberately left out. The "amount" of captured information is calculated by counting the "points" recorded in risk scenarios or statements.	Measured from videotape by comparing discussion and produced risk scenarios
Number of goals in effect sets	The number of goals in effect sets of the scenarios	Counted from produced risk scenarios
Number of unclear issues	The number of unclear issues that arise, when using different documentation methods Count two metrics: issues related to the use of the method and issues related to subject matter	Counted from the videotape by following the discussion
Usability	Open ended question, Ask respondent to rate the methods in order of usability	Questionnaire
Ease of use	Open ended question Ask respondent to rate the methods in order of ease of use	Questionnaire
Effectiveness	Open ended question Ask respondent to rate the methods in order of effectiveness	Questionnaire
Overhead	Open ended question Ask respondent to rate the methods in order of effectiveness	Questionnaire, open ended question
Recommendation for further use	Recommendations for further use from the participants	Questionnaire, open ended question
Improvement suggestions	Improvement suggestions from the participants	Questionnaire, open ended question

Table 55: Metrics used in the risk documentation study

The risk form that was almost identical to the one presented in Table 28, the only additional fields were fields defined for recording the history of changes to the form, closing date, and closing rationale.

Case Study Questionnaire

Group name: _____

1. Please, put the following five risk documentation methods in order of preference from the point of view of **usability**

   ___ Software
   ___ Drawing Riskit analysis graphs on whiteboard
   ___ RiskitFrames
   ___ Risk sheets
   ___ SEI risk statements

2. Please, put the following five risk documentation methods in order of preference from the point of view of **ease of use**

   ___ Software
   ___ Drawing Riskit analysis graphs on whiteboard
   ___ RiskitFrames
   ___ Risk sheets
   ___ SEI risk statements

3. Please, put the following five risk documentation methods in order of preference from the point of view of **effectiveness**. The definition of effectiveness = supports **all** essential aspects of risk documentation without overhead or unnecessary activities.

   ___ Software
   ___ Drawing Riskit analysis graphs on whiteboard
   ___ RiskitFrames
   ___ Risk sheets
   ___ SEI risk statements

4. Comments on the overhead required for different documentation methods.

5. Recommendations for further use.

6. Improvement suggestions.

Figure 42: The questionnaire used to capture student feedback in the risk documentation study

The study objectives were formulated into a GQM statement, stated in Table 54. The objectives in Table 54 were decomposed into a set of metrics that are presented in Table 55.

As Table 55 implies, we used video recordings, questionnaires, and postmortem analysis of artifacts to capture the required data from the study. The questionnaire used is presented in Figure 42. The video recording analysis was done shortly after the sessions. A stopwatch and agreed coding principles were used to log the data into tables.

The analysis of the data is based on simple raw data and calculating averages. Even though the study included a fairly large number of students (18), the relevant data granularity was collected at the team level, resulting in essentially four data points for each metric.

5.6.2 Participant Selection

Participants in the study were students who were taking the class "Tik-76.115 Software Project". The Software Project course is a two-term (ca. nine months), five-credit course, where students work through major software projects in groups of six to seven people (HUT 2001). Each project comprises all typical software design and implementation phases, such as requirements analysis, conceptual and detailed design, coding, testing, documentation, and delivery to the customer. Each student is expected to contribute five working weeks of effort to their projects, i.e., usually projects plan their work to contain approximately seven to ten person months. However, quite often these initial effort estimates are exceeded.

We asked volunteer project teams to take an additional course along with the software project course. This additional course was a risk management course and contained performing systematic risk management in the projects of the software project class. Each project team received training on the basic concepts of risk management and on the methods needed to perform risk management and to participate in this study. Each participating student received two credits for this course. Four teams and a total of 18 students signed up for the course and participated in the experiment.

Each project had prepared a risk management plan and documented its goals and risk management mandate before the risk identification and analysis session. We held separate sessions for each team and the sessions had similar structure. At the beginning of the session, the participants were given a short briefing on the five documentation methods to be used and on the agenda of the session. Risk identification session was held to identify potential new threats to their projects. Based on the identification results, the five risks to be documented were selected from these newly identified raw risks or from raw risks that have been identified on earlier risk management sessions. Selection was not random, it was based on what the team as a whole considered to be the most important risks to be analyzed in more detail.

	Group A	Group B	Group C	Group D
Scenario 1	SEI risk statements	SEI risk statements	RiskitFrames	Risk forms
Scenario 2	VISIO software	Drawing on blank flipcharts	Risk forms	VISIO software
Scenario 3	Drawing on blank flipcharts	RiskitFrames	VISIO software	Drawing on blank flipcharts
Scenario 4	RiskitFrames	Risk forms	Drawing on blank flipcharts	RiskitFrames
Scenario 5	Risk forms	VISIO software	SEI risk statements	SEI risk statements

Table 56: Order of using the documentation techniques in the group sessions

After this, the groups analyzed and discussed each of the five risks in turn. Discussion session, we randomly selected both the risk to be analyzed and documented, and the facilitator from the team to conduct the discussion on that risk. The groups used different methods in different order to avoid systematic learning bias in the study results, as shown in Table 56.

5.6.3 Analysis and Results

We used non-parametric statistical techniques in the analysis of the study data, in addition to qualitative, contextual analysis we performed. The limited number of student groups produced essentially only four data points for each "treatment". We used Wilcoxon's signed rank test (Conover 1999) to compare the techniques pairwise to test the hypothesis that techniques are not from the same sample. The data and analysis results are presented in the following.

The first study goal was to compare the time spent on each of the techniques. We measured elapsed time to complete the scenario, subtracted any time spent in general discussion, as well as time spent on discussing the method questions during the scenario development. The resulting data is shown in Table 57.

Metrics	Group A	Group B	Group C	Group D	Average	
Elapsed time/scenario – Software	7,5	9,5	10,75	7	8,7	min
Elapsed time/scenario – Drawing	11,75	13,9	6,75	8,2	10,2	min
Elapsed time/scenario – RiskitFrames	6	22,75	5,25	14,5	12,1	min
Elapsed time/scenario – Risk forms	5,75	6,75	3,25	6,25	5,5	min
Elapsed time/scenario – SEI risk statements	9,5	3,6	2	5,75	5,2	min
Total (average for average column)	*40,5*	*56,5*	*28*	*41,7*	*8,3*	min
General discussion time – Software	1	0,4	0,9	0,3	0,7	min
General discussion time – Drawing	0,1	0,2	0,5	0,5	0,3	min
General discussion time – RiskitFrames	0,1	0,9	0,1	1,4	0,6	min
General discussion time – Risk forms	0,8	0,1	0,1	0,2	0,3	min
General discussion time – SEI risk statements	1	0,1	0,1	0,3	0,4	min
Total (average for average column)	*3*	*1,7*	*1,7*	*2,7*	*0,5*	min
Time spent on method questions – Software	0	0	0,6	0,25	0,2	min
Time spent on method questions – Drawing	0	0,4	0,6	0,2	0,3	min
Time spent on method questions – RiskitFrames	0	0,6	1,5	0	0,5	min
Time spent on method questions – Risk forms	0	0,05	0,5	0	0,1	min
Time spent on method questions – SEI risk	0,6	0,4	0,05	0,2	0,3	min
Total (average for average column)	*0,6*	*1,45*	*3,25*	*0,65*	*0,3*	min
Effective time – Software	6,5	9,1	9,3	6,5	7,8	min
Effective time – Drawing	11,7	13,3	5,7	7,5	9,5	min
Effective time – RiskitFrames	5,9	21,3	3,7	13,1	11,0	min
Effective time – Risk forms	5,0	6,6	2,7	6,1	5,1	min
Effective time – SEI risk statements	7,9	3,1	1,9	5,3	4,5	min
Total (average for average column)	36,9	53,4	23,1	38,4	7,6	min

Table 57: Risk documentation study data on time usage

As Table 57 shows, the teams spent different amounts of time discussing their scenarios, team C being the fastest, and team B taking the most time. Note that the average time for modeling a risk scenario using the Riskit Analysis Graphs is about ten minutes. In the IESE/Tenovis study the average time for modeling a scenario was 17 minutes. We believe that this is mainly due to smaller projects and simpler project context for these student projects. This seems to confirm the finding presented earlier (f4.4) about Riskit Analysis Graphs being sufficiently fast to be used in risk analysis sessions.

We enumerated a set of hypotheses based on pairwise comparisons between methods. These hypotheses are listed Table 58. The table should be interpreted as follows: each cell documents the critical value of the Wilcoxon's test to test the null hypothesis that states that techniques are from the same distribution, i.e., hypothesis that stated that the technique on that line is faster than the technique mentioned in that column.

	Software	Drawing	RiskitFrames	Risk forms	SEI risk statements
Software		0.2269	0.2402		
Drawing			0.3416		
RiskitFrames					
Risk forms	**0.0675**	**0.0225**	**0.0841**		
SEI risk statements	0.1032	**0.0334**	0.1197	0.3519	

Table 58: Speed of using techniques: comparison cross-table and critical levels

We chose the risk level of 10% as a threshold for rejecting the null hypotheses. As Table 58 shows, risk forms can be considered faster than software, drawing on a blank flipchart, and RiskitFrames. In addition, the SEI statements are faster than drawing on a flip chart and their speed over software and RiskitFrames is close to the 10% threshold.

We decided not to use the data about the method related questions, due to the fact that method related questions only accounted for a small amount of time in sessions: the longest time spent on method questions was 3.25 minutes by group C, being about 12% of the total time. However, the average time spent on method questions was one and a half minutes (3.5% of total time) and the lowest time on method questions was 0.6 minutes (1.5% of total time) by group A. As these times were so low, we did not see that this data would reliably represent method related overhead of the techniques. In any case, the statistical analysis did not show any rejections on the hypotheses.

The analysis in Table 58 concerns only the effective time used in constructing the scenarios. It does not include any information about the effectiveness of the time, i.e., what was produced during this time. We have calculated the ratio between effective time and number of risk items produced during the analysis. This data is shown in Table 59 in rows titled "time per elements". Since the SEI statements only support the capture of risk factors and the risk event, they were all categorized as having two elements.

Metrics	Group A	Group B	Group C	Group D	Average	
Number of elements – Software	4	6	6	6	5,5	items
Number of elements – Drawing	15	17	4	8	11,0	items
Number of elements – RiskitFrames	7	16	4	8	8,8	items
Number of elements – Risk sheets	8	6	5	9	7,0	items
Number of elements – SEI risk sattements	2	2	2	2	2,0	items
Total	*74,5*	*101,5*	*47*	*72,7*	*73,9*	items
Time per elements – Software	1,6	1,5	1,5	1,1	1,4	min
Time per elements – Drawing	0,8	0,8	1,4	0,9	1,0	min
Time per elements – RiskitFrames	0,8	1,3	0,9	1,6	1,2	min
Time per elements – Risk sheets	0,7	0,4	0,7	0,8	0,6	min
Time per elements – SEI risk statements	2,5	3,3	1,3	3,0	2,5	min
Total	*6,4*	*7,3*	*5,9*	*7,4*	*6,8*	min
Number of goals in effect sets – Software	2	0	3	1	1,5	items
Number of goals in effect sets – Drawing	3	3	1	3	2,5	items
Number of goals in effect sets – RiskitFrames	3	4	1	3	2,8	items
Number of goals in effect sets – Risk sheets	2	3	1	3	2,3	items
Number of goals in result – SEI risk statements	1	1	1	3	1,5	items
Total	*11*	*11*	*7*	*13*	*10,5*	items

Table 59: Risk documentation study data on information documented

We used the "time per element" data in Table 59 and produced a similar set of hypothesis as earlier. This data is presented in Table 60. As the table shows, there are two main patterns that are obvious:

(f5.1) All techniques seem to be more efficient than the SEI statements.

(f5.2) Risk forms seem to be more efficient than any other technique.

This conclusion could be drawn even at 5% risk level. It is also interesting to point out that at a risk level of approximately 20% the blank flipchart technique and the RiskitFrames could be considered more effective than software.

	Software	Drawing	RiskitFrames	Risk forms	SEI risk statements
Software					**0.0363**
Drawing	0.2142				**0.0151**
RiskitFrames	0.2042	0.5000			**0.0199**
Risk forms	**0.0099**	**0.0254**	**0.0459**		**0.0161**
SEI risk statements					

Table 60: Effectiveness of techniques: comparison cross-table and critical levels

We also compared whether there are differences between the number of goals the techniques prompted participants to document. The rationale for this test was that goals – and stakeholders – represent significant information that characterizes risks. The more accurately

they are defined, the more accurate we assumed that the risk description was. These results are presented in Table 61.

	Software	Drawing	RiskitFrames	Risk forms	SEI risk statements
Software					0.5000
Drawing	0.2114			0.1955	**0.0908**
RiskitFrames	0.1955	0.1955		**0.0908**	**0.0971**
Risk forms	0.2736				0.1076
SEI risk statements					

Table 61: Number of goals listed by the techniques: comparison cross-table and critical levels

The Table 61 suggests two conclusions:

(f5.3) Drawing on a flipchart and RiskitFrames seem to document goals in more detail than the SEI risk statements.

To some degree this is an expected result, as the underlying Riskit Analysis Graph conceptual model explicitly recognizes effects and their link to goals. However, note that the forms themselves do not explicitly prompt to list goals, just the effects.

(f5.4) RiskitFrames seem to document goals in more detail than risk forms.

This is perhaps due to the graphical modelling paradigm that may lead participants to think about the links that are associated to each scenario. Note that the null hypothesis to risk forms describing goals in more detail than SEI risks statements was quite close to being rejected.

Table 62 provides data on how well the information that was raised during the discussion was actually recorded in the documentation of the scenarios. We performed two statistical analyses on this data. First, we wanted to compare whether techniques possibly prompted more information to be discussed during the session. This is reflected by the items "information produced". Second, we want to study how well different techniques succeed in capturing the information that was raised. This was analyzed using the "information captured / produced" ratio as the metric. The rationale for this metric was that we wanted to compare the methods with respect to how accurately they capture the content of the discussion in the documentation. Any information not documented at a session is likely to be ignored in future discussions about the risks. Therefore, methods that capture all or most of the discussion content were considered better methods. Table 63 presents the results of the analysis on information produced and Table 64 presents the results of how well the information was captured.

Metrics	Group A	Group B	Group C	Group D	Average	
Information produced – Software	9	6	7	8	7,5	points
Information produced – Drawing	13	20	6	10	12,3	points
Information produced – RiskitFrames	6	18	4	11	9,8	points
Information produced – Risk forms	8	10	4	10	8,0	points
Information produced – SEI risk statements	11	3	3	6	5,8	points
Total	47	57	24	45	43,25	points
Information captured – Software	7	6	7	7	6,8	points
Information captured – Drawing	13	18	4	10	11,3	points
Information captured – RiskitFrames	6	17	4	10	9,3	points
Information captured – Risk forms	7	9	4	10	7,5	points
Information captured – SEI risk statements	5	3	2	5	3,8	points
Total	38	53	21	42	38,50	points
Information captured / produced – Software	0,78	1,00	1,00	0,88	0,91	points
Information captured / produced – Drawing	1,00	0,90	0,67	1,00	0,89	points
Information captured / produced – RiskitFrames	1,00	0,94	1,00	0,91	0,96	points
Information captured / produced – Risk forms	0,88	0,90	1,00	1,00	0,94	points
Information captured / produced – SEI risk statements	0,45	1,00	0,67	0,83	0,74	points
Total	4,11	4,74	4,33	4,62	4,45	points

Table 62: Risk documentation study data on information produced vs. captured

	Software	Drawing	RiskitFrames	Risk forms	SEI risk statements
Software					0.1377
Drawing	0.1200		0.1144	**0.0728**	**0.0811**
RiskitFrames	0.2853			0.2399	0.2057
Risk forms	0.3844				0.1848
SEI risk statements	0.8623				

Table 63: Information produced: comparison cross-table and critical levels

The analysis of results shown in Table 63 indicates that

(f5.5) drawing on a blank sheet produces more information than risk forms and SEI risks statements.

We propose that there are two different reasons fro this. The flipchart is a free-format modeling approach and as such likely to lead to unconstrained discussion and potentially allow freer association of various aspects in the discussion. Compared to risks forms, predefined forms may limit participants' mental "search space" and instead they focus on what is presented in the form. SEI risk statements, on the other hand, contain little guidance what to include in them. They perhaps are too limiting in the sense that they limit the focus on the two parts of the statement but provide little cues for providing additional information.

	Software	Drawing	RiskitFrames	Risk forms	SEI risk statements
Software					**0.0695**
Drawing	0.5671				0.1788
RiskitFrames	0.2447	0.2485		0.3577	**0.0972**
Risk forms	0.2977	0.3125			**0.0853**
SEI risk statements					

Table 64: Information captured vs. produced: comparison cross-table and critical levels

The analysis of the results in Table 64 suggests that

(f5.6) software, RiskitFrames, and risk forms are more effective in capturing the content of the discussion than the SEI risk statements.

However, none of the other hypothesis could be confirmed. This is most likely explained by the more detailed modeling formalism that the Riskit based techniques support. This theory is also supported by the drawing based technique not being able to dominate the SEI statements: drawing requires participants to remember the method in order for it to support the capture of information.

Metrics	Group A	Group B	Group C	Group D	Average	Average-based ranking
Usability – Software	1	1	2	3	1,75	2
Usability – Drawing	4	4	3	4	3,75	4
Usability – RiskitFlaps	2	2	1	1	1,50	1
Usability – Risk sheets	3	3	4	1	2,75	3
Usability – SEI risk statements	4	5	5	5	4,75	5
Ease of use – Software	1	2	3	2	2,00	2
Ease of use – Drawing	3	4	2	5	3,50	4
Ease of use – RiskitFlaps	2	3	1	1	1,75	1
Ease of use – Risk sheets	5	5	5	3	4,50	5
Ease of use – SEI risk statements	4	1	4	4	3,25	3
Effectiveness – Software	1	1	1	3	1,50	1
Effectiveness – Drawing	4	4	3	4	3,75	4
Effectiveness – RiskitFlaps	2	3	2	1	2,00	2
Effectiveness – Risk sheets	3	2	4	2	2,75	3
Effectiveness – SEI risk statements	5	4	5	5	4,75	5

Table 65: Risk documentation study: participant perception rankings

Finally, Table 65 presents how the participants perceived the various five different methods w.r.t. usability, ease of use, and effectiveness. These three different terms were defined and clarified in the questionnaire and verbally before respondents completed the forms. Usability referred to how convenient and practical the method is to use, ease of use specifically attempted to clarify how easy and intuitive the method was perceived, and the

effectiveness aimed at clarifying whether participants considered the method to cover or support all essential aspects of the analysis without unnecessary overhead.

We performed similar analysis on this data as on the previous data. Results are shown in Table 66, Table 67, and Table 68.

	Software	Drawing	RiskitFrames	Risk forms	SEI risk statements
Software		**0.0328**		**0.0582**	**0.0339**
Drawing					**0.0512**
RiskitFrames	0.3527	**0.0317**		**0.0512**	**0.0339**
Risk forms		**0.0582**			**0.0339**
SEI risk statements					

Table 66: Usability of techniques: comparison cross-table and critical levels

The usability data indicated that

(f5.7) Software is perceived as more usable over drawing on flipcharts, risk forms, and SEI risk statements.

(f5.8) RiskitFrames were considered more usable over flipchart drawing, risk sheets, and the SEI risk statements.

(f5.9) Risk forms were considered more usable than drawings. All techniques were considered more usable than the SEI risk sheets.

(f5.10) The software-based technique and RiskitFrames seemed to have received most support in terms of usability.

	Software	Drawing	RiskitFrames	Risk forms	SEI risk statements
Software		**0.0582**		**0.0339**	**0.0582**
Drawing				0.1158	
RiskitFrames	0.3527	**0.0294**		**0.0328**	**0.0582**
Risk forms					
SEI risk statements		0.4270		**0.0582**	

Table 67: Ease of use of techniques: comparison cross-table and critical levels

Three conclusions can be drawn from the data presented in Table 67 about the perceived ease of use of the techniques:

(f5.11) The software-based modeling was considered easier than drawing on a blank flipchart, using risk forms, and the SEI risk statements.

(f5.12) RiskitFrames was also considered easier to use than the said three other techniques.

(f5.13) The SEI risk statements were considered easier to use than the risk forms.

	Software	Drawing	RiskitFrames	Risk forms	SEI risk statements
Software		**0.0339**	0.2887	**0.0582**	**0.0328**
Drawing					0.5120
RiskitFrames		**0.0339**		**0.0582**	**0.0328**
Risk forms		**0.0582**			**0.0339**
SEI risk statements					

Table 68: Effectiveness of techniques: comparison cross-table and critical levels

Table 68 presents the results regarding the perceived effectiveness of the methods. This data indicates the following conclusions:

(f5.14) The software-based technique and the RiskitFrames were considered more effective than drawing on a blank flipchart, risk forms, and the SEI risk statements.

(f5.15) Risk forms were considered more effective than the drawing and the SEI risk statements.

5.6.4 Study Conclusions

The main conclusions from the analysis presented in the previous chapter can be summarized as follows:

- The SEI risk statements were least efficient in modeling risks (f5.1).
- Risk forms seem to be the fastest technique to use in completing risk documentation, both in terms of time to complete a risk scenario, and also taking into account the amount of information produced (f5.2).
- Drawing on a blank flipchart and RiskitFrames seem to be most effective in capturing information about goals and how they relate to risks (f5.3).
- Drawing on a blank flipchart seems to produce more information in a session than risk forms or the SEI risk statements (f5.5).
- Software-based tool, RiskitFrames, and risk forms seem to be effective in capturing the information discussed during a session (f5.6).
- RiskitFrames and the software-based solution were perceived as more usable than other techniques (f5.7 and f5.8); and risk forms were perceived as easier to use than drawing on a blank flipchart (f5.9).
- RiskitFrames and the software based solution were perceived as easier to use than other techniques (f5.11 and f5.12), and the SEI risk statements were perceived as more usable than risk forms (f5.13).
- RiskitFrames and the software based solution were perceived as more effective than other techniques (f5.14), and risk forms were perceived as more effective more effective than drawing on a blank flipchart and the SEI risk statements (f5.15).

In short, the RiskitFrames and software-based use of the Riskit method were considered favorably by several of the key metrics we used. Correspondingly, the SEI risk statements were often inferior to some other methods in the analysis we performed.

The overall arrangements and design of this study was largely based on the underlying concepts of the Riskit method. The problems associated with the SEI risk statements may be partially due to this set up. However, some of the metrics were independent of the underlying method. For instance, the information produced versus captured data recorded all information that was presented in the meeting, regardless of whether it was Riskit or SEI "compatible". Another potential bias in the student group may have been that students may have received more exposure to Riskit method than to the SEI method. Assuming that this exposure would have resulted in favorable bias for the Riskit method, the student feedback on the perception question might have been affected by this. We did not observe any such impact but cannot rule such an effect out, either.

Overall, the study indicated that the Riskit-based techniques seem to work well in practice, and it characterized the advantages and disadvantages of various ways of constructing Riskit Analysis Graphs. The data presented in this chapter can be used to provide guidance on what method to use in a given situation. It is also good to note that methods can be combined in practice. For instance, the use of RiskitFrames or flipcharts can be augmented by one team member simultaneously documenting the results using a software tool, or someone filling in forms while others are discussing risks at a whiteboard.

The software-based solution, in particular, offers a potential high payoff, since students were only considering benefits from the point of view of simple drawing aids. The automatic capture of the risk information during the graphical editing function offers substantial additional elements for sharing information, tracking and updating information, and using the information for process improvement purposes.

6. Conclusions

In this work, we have made four main contributions, a method for risk management, an improvement framework for risk management, the eRiskit application, and findings from empirical studies. In the following, we discuss the conclusions related to each, and make some general conclusions at the end.

6.1 Riskit Method

In the following, we will discuss the findings and contributions of this work about the Riskit method, first by evaluating the method against the method development requirements we presented in 1.3.1 (Table 1), then by highlighting some general empirical findings, and then discussing the main characteristics of the method. We present some problems associated with the method at the end of the chapter.

The requirements for Riskit method development were partially derived from literature review but we also used our own insight and experience to synthesize them. The focus group study, reported in chapter 5.1, provided support for them, in particular for the importance of usability, feasibility of methods, validity of modeling approaches, and credibility of the method. The other requirements also received some support from the focus group study. In addition, the initial suppositions of our research (chapter 1.1) were supported by the focus group data. Thus, we believe that the requirements we presented in chapter 1.3.1 are a valid for risk management method development. In the following, we discuss the Riskit method in light of these requirements.

The first method requirement (R-1) called for consistency in applying the method, i.e., independent users should apply the method in similar way and get the same results in the same situation. The full process definition of Riskit and the practical templates and techniques embedded within it are the mechanisms that aim to satisfy this requirement. It is clear that given the subjective and vague underlying nature of risk, full consistency is an unachievable goal: individuals will always have some differences how they perceive risks, and, hence, the risk management results will not be entirely consistent. However, the feedback from empirical studies indicated that this characteristic is perceived as one of the strengths of Riskit. The NASA study (study 2, chapter 5.1), the study with DaimlerChrysler and Nokia (study 4, chapter 5.4), and the IESE study (study 5, chapter 5.5) all gave positive assessments of Riskit's systematic approach and indicated that it helped perform better risk management.

The second requirement (R-2) called for the method to be usable, i.e., it should be easy to learn and use. The Riskit method is supported by extensive training material and we have given several tutorials on it, including one at the International Conference in Software Engineering in 1999 (Kontio 1999). Our empirical studies indicate that with facilitation support, a training of half a day to a full day in duration gives sufficient basis to use Riskit in practice (IESE study, chapter 5.5). On the other hand, more limited training, as the two-hour training in the DaimlerChrysler and Nokia study (study 4, chapter 5.4), seemed to result in

problems in applying the method. These empirical studies were based on facilitation support in the use of Riskit and we cannot use these studies to draw conclusions on what level of training is sufficient for projects to use Riskit independently. However, after the initial, facilitated cycles of the Riskit, the project personnel reached the ability to use Riskit on their own. The study with university students (study 6, chapter 5.6) also indicated that with approximately a day's worth of training and practice, independent use of Riskit is feasible. The study with students also indicated that users ranked the ease of use and usability of Riskit high.

We are also aware of several other organizations that have adopted Riskit as their risk management approach entirely based on the public documentation of the method. This also suggests that the method is deployable and usable in practice.

The third requirement (R-3) called for adaptability, i.e., the method should lend itself to different situations and projects. The wide scope of organizations and domains of the empirical studies we reported in chapter 5 indicates that Riskit is adaptable to many situations. Also, the existence of other industrial users indicates that the method is adaptable to different situations.

Based on the studies presented in this work, the Riskit method satisfies the feasibility requirement (R-4) as well. The NASA study (chapter 5.2), Hughes study (chapter 5.3), the study with DaimlerChrysler and Nokia (chapter 5.4), the IESE study (chapter 5.5), and the study with university students (chapter 5.6) all indicate that the method feasible in practice. Especially the industrial projects, given their tight performance targets, confirm this conclusion: the industrial participants in these studies were primarily concerned with the success of their project and it is likely that they would have voiced their concerns if they had considered the method not feasible.

The Riskit method also seems to be a complete one (requirement R-5): projects were able to conduct all of their risk management activities within the Riskit process and additional steps or actions were not necessary. We also noted that we have not encountered a situation where a risk could not be effectively modeled by the Riskit Analysis Graphs, having used the Riskit Analysis Graphs to document several hundreds of different risk scenarios.

The requirement concerning validity (R-6) is difficult to assess as the "true values of risk are unknowable" (see chapter 5). However, the Riskit Analysis Graph has been a sufficient mechanism to document all risk scenarios we have encountered in the studies and this would seem to suggest that the representational power of the graph is sufficient to model most risks. As the Riskit Analysis Graph was considered a useful representational scheme in several studies (The NASA study, chapter 5.1, DaimlerChrysler and Nokia study, study 4, chapter 5.4, IESE study, chapter 5.5, university study, chapter 5.6), we suggest that the validity of Riskit is sufficient from practitioner perspective. The university study also indicated that the Riskit Analysis Graphs are effective in capturing the information surfacing during a discussion about risks.

We also listed credibility as one of the method development requirements (R-7), i.e., the method should increase confidence in the validity of risk analysis results. This was supported by the NASA study (chapter 5.1), the study with DaimlerChrysler and Nokia (chapter 5.4), and the IESE study (chapter 5.5). The university study (chapter 5.6) also seems to support this as the students ranked Riskit based techniques among the highest in terms of effectiveness.

The improved communications of risk information (requirement R-8) was mentioned as a benefit in the NASA (chapter 5.1) and IESE (chapter 5.5) studies. However, data from the DaimlerChrysler and Nokia study (chapter 5.4) indicated that the Riskit Analysis Graphs can also be perceived as complex and hard to understand. While this could be compensated by providing more training, it is likely that for an untrained eye the Riskit Analysis Graphs may be difficult to interpret. This is a potential disadvantage as it may be difficult and expensive to provide sufficient training to all people who may encounter Riskit Analysis Graphs. Ideally, these graphs should be intuitive enough on their own so that they can be understood sufficiently well without special training.

The cost-effectiveness of the Riskit method (requirement R-9) was also addressed in the studies. The absolute cost and overhead of the Riskit method seems to be higher than some of the other methods we compared (the NASA study, chapter 5.2) and it may not be as efficient in risk analysis as some other methods (the NASA study, chapter 5.2). However, the overall cost of applying Riskit was considered acceptable (NASA study, chapter 5.2; DaimlerChrysler Nokia study, chapter 5.4; IESE study, chapter 5.5), and Riskit Analysis Graphs are effective in supporting risk analysis and capturing risk information (university study, chapter 5.6).

The empirical studies also produced several more specific findings about the Riskit method. First and foremost, the Riskit method was shown to be a feasible approach for risk management in both small and large industrial software projects (studies 2, 3, and 4). More specifically, empirical studies indicated the following overall characteristics:

- The Riskit method is fully documented and easy to follow, contributing to systematic and consistent management of risks (studies 2 and 5).
- Riskit increased participants' confidence in risk management results (studies 2 and 4).
- While the Riskit method had higher overhead than some of the comparison methods, its users perceived the benefits provided by the method to more than compensate for the higher overhead (studies 2 and 5).

The previous discussion reflected on the method development requirements set for Riskit. It is also interesting to discuss the conclusions we can make on the specific characteristics of Riskit, i.e., the stakeholder and goal concepts, Riskit Analysis Graphs, and the Riskit Pareto ranking approach.

The stakeholder and goal review aspects of the Riskit method were considered beneficial. According to our studies, Stakeholder analysis improves project management's understanding of project priorities and objectives (studies 2 and 4). The goal review step, on the other hand, contributes both to improved project focus (studies 2 and 4) and improved risk management activity (study 2).

The Riskit Analysis Graphs were perceived useful by most of the empirical data. They were considered easy to use, effective, and produced detailed descriptions or risk scenarios, capturing well the various aspects that were raised during risk analysis (study 6). In addition, when used selectively for more complex risk scenarios and when sufficient training or facilitation has been provided, they visualize risk information well, they can be developed efficiently, and participants can see both details and a bigger picture with them (studies 4, 5 and 6). However, as we pointed out earlier, some users were concerned about the complextity of the diagrams in some risk scenarios.

The Riskit Pareto ranking technique was considered practical and understandable method to rank risks (study 5). Given that the Riskit risk prioritization approach is well justified by

mainstream theories in economics, management science, and psychology, we believe that it is a substantially more reliable approach to rank risks than the currently widely used methods in software engineering.

Based on the previous discussion, we conclude that the Riskit satisfies the development requirements defined earlier in this thesis for a risk management method. However, we have also identified several problems that should be resolved in order to improve the method. We will discuss these in the following.

As the eRiskit software is not a mature product, Riskit is currently mainly a manual method for an industrial practitioner. It carries a considerable overhead in documentation, and large volumes of risk may become a high overhead for a project. Software support for this should be developed and made available to Riskit users.

We should find a way to make the Riskit Analysis Graph formalism more intuitive without sacrificing the precision and representational power. This might be done by supporting simpler versions of the graph and providing easy mechanisms to refine the information at a later stage.

The implementation of risk controlling actions in the Riskit method is assumed to be taken care of by projects' management structures. However, the IESE study indicated that problems in implementation undermine the effectiveness of the whole risk management process. The Riskit method should, perhaps, provide more support in making sure that controlling actions are implemented.

6.2 Improvement Framework

The risk management improvement framework we have presented in this work is based on Basili's Experience Factory concept. The risk management improvement framework is an adaptation and specialization of the Experience Factory. The framework presented here contains the process, documents key roles in the process, and identifies the main information types needed in the risk management improvement framework. The framework itself has been implemented as a proof-of-concept model and it has not been formally subjected to empirical validation. However, the empirical studies reported in this thesis were instances of such a framework in practice, providing a concrete demonstration of the framework in operation.

As with the Riskit method, we have defined as set of requirements for the risk management improvement framework in the beginning of our work (see chapter 1.3.2, Table 2). We will now discuss how the proposed framework satisfies these requirements.

The first requirement called for a continuous learning cycle (I-1). As the framework is based on the QIP cycle (see chapter 4.2), it implements a well-established continuous improvement process. The empirical studies, as well as the analysis steps documented in this thesis, are examples of this process in practice.

The second requirement (I-2) required a complete improvement process. Our improvement framework covers the complete improvement cycle and, to some degree, extends to QIP cycle by incorporating the external input and influences explicitly into the process. The Appendix A contains a detailed definition of the improvement process.

A key characteristic of our improvement framework is the use of empirical data and experience to support learning and improvement (requirement I-3). The empirical studies

reported here also demonstrate how the use of this principle in practice, and the empirical study designs can be used as examples of study packages to repeat such studies.

The improvement framework also provides some support for capturing risk management data and raw experience (requirement I-4). The Riskit templates and proposed Experience Base content both support this capture but perhaps the most significant contribution in this area is the eRiskit application: as it supports the whole Riskit process and captures all risk management information in that process, it is a powerful tool to support experience capture. While we have not evaluated the application in industrial context, the application is operational and contains actual project data from a university development project.

The fifth requirement for the improvement framework related to the clarification of roles and responsibilities (requirement I-5). We have adopted the concepts used in business process management and defined roles for key roles in risk management and risk management process improvement (see chapters 3.6 and 4.6). All of these roles have been taken by people participating in our empirical studies and we thus have first-hand experience of them in practice. However, this experience is anecdotal and subjective and we only use this experience to demonstrate the proof of concept for these roles. Alternative definitions for roles are also possible.

The framework proposed here has defined possible repositories for risk management knowledge in the form of the risk management Experience Base (requirement I-6).

Finally, traceability of experience and knowledge was listed as the seventh requirement for the improvement framework (I-7). The framework itself does not provide direct support for this requirement, except in the form of the eRiskit application. However, the Experience Factory paradigm itself is based on explicit documentation of the analysis process and original empirical data. An orthodox implementation of Experience Factory and our improvement framework should provide basic elements of traceability.

We believe that the adaptation of the EF and QIP cycle for risk management also contributes to the general development of the Experience Factory concept.

An additional contribution to the improvement framework is the documented empirical study designs presented in this thesis. They can be reused to replicate these studies or as templates for other studies, leading to improved knowledge in the risk management practice in software engineering.

The proposed improvement framework is a conceptual model with a sample implementation through our empirical studies and this research work. As such, it has not been empirically evaluated and we cannot make claims or conclusions on its characteristics. On the other hand, validating such high-level concepts is quite difficult as such concepts have several potential different instantiations, each one being influenced by several uncontrollable environmental factors. Even though it is difficult or impossible to obtain empirical validation for such a framework, such frameworks can be used as concrete planning models in practice. We believe that since the framework is based on a well-established model and the work presented here provides an initial proof of concept; our improvement framework can be used as reference model for implementing such frameworks in practice.

6.3 eRiskit Application

The eRiskit application has been implemented, containing functionality to support all steps of the Riskit method. The application has also been populated with real risk management data from a project that implemented the application. However, the application has not been used in industrial projects. While all major errors have been corrected in the application, it contains several minor errors and inconvenient features.

The main function of the eRiskit application from the perspective of this research has been to act as a proof of concept for implementing software support for the Riskit method. As such, it has been a successful implementation, demonstrating the key concepts of the Riskit method and the underlying conceptual model can be formalized into form of software. Given that the empirical studies highlighted the need to provide software support for Riskit, the application has potential to benefit practitioners.

Several of the empirical studies highlighted the potential benefits from supporting risk management activities by functionality similar to those in eRiskit (studies 4 and 5). Among the benefits identified are more effective management of risk information, better sharing of information, more accurate tracking of risks, avoiding errors in data, effort and cost savings, and accumulating experience for improvement. However, our focus group study (study 1) indicated that such an application would need to be well integrated with other operational information system in a company or, alternatively, be very cost-efficient to obtain and use.

The most important features of the eRiskit application are the distributed collection of and wide and consistent access to risk information. Maintaining links to stakeholders and goals, as well as using reliable methods in risk prioritization were also considered valuable features. However, focus group participants did not consider the visualization of risk information and the collection of risk information for improvement purposes as valuable; and the accuracy of risk information was not considered a necessary feature.

6.4 Conclusions about Risk Management in General

Our studies also produced several more general findings about risk management, i.e., useful guidelines that are not Riskit specific.

First of all, the software and project industries are looking for systematic, sound, but easy to use and low-overhead approaches for risk management (study 1). We believe that most of the currently available methods do not satisfy this need, given their theoretical limitations. On the other hand, Riskit is one potential approach that can effectively satisfy this need.

Second, risk management process needs to be supported and enforced to ensure sufficient and continuous management of risks (study 4). This can also be improved by providing sufficient training and motivation to project personnel, and clarifying the roles and responsibilities for risk management participants (study 4). Most organizations currently have minimal requirements for performing risk management and, as a result, risks are rarely effectively managed. Our experience also indicates that the earlier the risk management process is introduced to the project, the easier and more effective its use is, and its impact is likely to be higher in reducing project risks (study 4).

Third, several of our studies suggested that effective risk management requires that people have been given sufficient training for it (studies 4, 5 and 6). Our practical experience

indicates that half a day to a full day of training gives a sufficient basis for conducting systematic risk management.

Fourth, it seems that checklists may introduce biases into the risk identification process if they do not match well to the domain and project characteristics (study 3). Therefore, we recommend that checklists be customized for each domain and organization to provide most value. Such customization can be supported by systematic capture of risk management information, as suggested by our risk management process improvement framework. We also found that structured brainstorming seems to be effective in finding most relevant risks quickly. Checklists can be used most effectively to cover the potential omissions (studies 3 and 5).

Our fifth general conclusion is that practitioners seem to favor simple and straightforward techniques over more thorough and complex ones (study 4), especially if they are not aware of the theoretical and practical limitations of these simpler approaches. However, intuitive risk prioritization may yield different, and possibly biased, results, compared to more systematic prioritization approaches (study 4). We believe that practitioners should be given sufficient and practical information about these limitations so that they can make more conscious and educated decisions about the methods they use.

Sixth, effective implementation of risk controlling actions is a critical link in risk management. The implementation of risk controlling actions needs to be tracked to ensure their impact (study 5). All risk management activities are meaningless if the controlling actions are not effectively implemented. Our studies indicated that current problems can take priority over proactive measures, easily leading to a fire-fighting mode in projects.

Risks are often documented in forms. An interesting detail that was discovered in study 6 is that risk forms seem to be fast and easy to use, but not very effective in capturing different aspects of risk information. We recommend that methods like Riskit Analysis Graphs are used to augment and support form based risk documentation approaches to enrich and deepen the analysis.

Seventh, we believe that a common risk management approach and framework makes communication about risks more effective between different stakeholders and projects (study 4). Each organization should develop and deploy such a framework to support more consistent and effective risk management.

Finally, we believe that risk management activities should be closely integrated with the project management activities. Even though we have presented the risk management process as a separate process in this work, the practical implementation of risk management should allow project management to conduct risk management activities as an integral part of project management, including frequent identification, analysis, control, and monitoring of risks.

6.5 Future Work

This research and the results achieved have deepened our understanding of the issues and challenges in software engineering risk management. In this chapter, we briefly outline some important and promising research issues that should be addressed to improve risk management practice in industry.

The most obvious missing link in the results presented in this research is the piloting of the eRiskit application in a large scale. This would allow evaluating it in from two perspectives: to assess how well it can support practical risk management in a project, and to evaluate

usefulness of the risk management data that will be accumulated into the risk management database. Both evaluation goals would bring valuable information on the eRiskit application itself, as well as the overall concept of risk management experience capture. In particular, it would be interesting to study how information about past risks can be used to support current risk analysis and how the risk management experience can be generalized.

The full evaluation of the risk management improvement framework in a single organization, including the establishment of a risk management Experience Base, would also be a valuable contribution in evaluating its feasibility and value for practical purposes. An important potential contribution in the risk management Experience Base are the risk patterns: while we have identified some tentative patterns already, it would be most valuable to identify additional ones, validate them, and understand how generic or domain specific they can be. The potential power of Riskit –based risk patterns is that instead of identifying and validating "flat" risk items, the Riskit Analysis Graphs can support the identification of complex and powerful risk patterns, i.e., a set of risk factors, risk events, reaction, and effect sets, as well as effective risk controlling action associated to such patterns. However, while the notion of risk patterns is very attractive, it remains unclear whether practitioners would actually benefit from them – it is plausible that the complexity of searching, understanding and reusing such patterns creates too much overhead and practitioners may prefer building up risk scenarios independently.

The studies reported in this research provided an initial assessment of the characteristics of the Riskit method. We strongly recommend that these studies are replicated so that the findings can be confirmed or revised, and additional insights gained. There are also several specific questions and issues that should be studied further, such as:

- What is the right level of abstraction for defining goals for risk management purposes?
- What are the cost benefits and relative coverage of checklist –based risk identification vs. brainstorming?
- How effective and appropriate are different focused brainstorming approaches, such as goal review, assumption analysis, critical path analysis, and stakeholder analysis?
- How serious are the proposed biases listed in chapter 2.2 in software project risk management and what are the most effective ways to control them?
- What are the most effective and reliable methods for ranking a large number of risk scenarios when only ordinal scale information is available about loss and probability?

It is also important to study how effective risk management influences business strategies and decision making. Our hypothesis is that effective risk management allows companies to take bigger risks that have higher business benefits. In order for this to happen, executives will need to know how effective and reliable their risk managent system is. It would be important to study and understand (i) how to validate the credibility of a risk management system, (ii) how to estimate the increased risk taking capability, and (iii) how executives as decision makers can intergrate risk information in their decision making process.

We also believe that small, start-up companies operating in high-growth areas obviously would benefit from good risk management. It would be valuable to understand what kind of risk management strategies such companies have used so far and how the Riskit method would need to be adopted to support such business environment.

We have applied the Riskit method in software engineering context. However, we believe that the underlying theory and concepts of Riskit are applicable to any goal –oriented undertaing or business. Thus, it would be valuable to evaluate this hypothesis by testing the

Riskit method in different domains, such as project business, construction, civil engineering, strategy planning, and public policy decision making.

Finally, this research has demonstrated that there are several challenges that make empirical studies in risk management difficult. We would like to call for researchers to innovate and adapt different research approaches and designs to study this area better. Without sound empirical basis the risk management research will remain abstract and be largely based on opinions. Industry needs more credible answers from researchers.

6.6 General Conclusions

This doctoral thesis reported on research that had a relatively large scope; a risk management method was developed, supported by an improvement framework and an application, and several empirical studies were conducted to evaluate the method. In addition, risk management is a broad and challenging topic, as it is a truly multidisciplinary problem and it is a difficult topic to study empirically. Due to this large scope, it has not been possible to provide equal coverage, formality, and empirical validation to all aspects of the constructs we have developed. Instead, we focused on evaluating the Riskit method through a series of case studies and one experiment.

Evaluating comprehensive methods and frameworks is difficult and software engineering risk management domain presents particular difficulties for such evaluation. We do not claim to have completed definitive empirical evaluation of our work. However, we believe that the several studies that we have conducted characterize the methods and provide insights to their strengths and weaknesses. To our knowledge, these studies represent one of the most comprehensive series of empirical studies in software risk management and they have been conducted using as sound scientific principles that the practical, industrial constraints have allowed.

Most of the empirical studies in this research have been conducted in industrial settings. While this has given us valuable experience, the practical constraints and difficulties have often forced us to make compromises from the ideal empirical study arrangements. However, as we see that the challenges in risk management are practical ones, not theoretical ones, we propose that such practical, industry-based work is the correct way to pursue risk management research.

One common theme prevails over all industrial studies we have performed. Despite our attempts – and sometimes even too rigorous planning approach, empirical studies performed in industrial context are susceptible to various changes. Obviously, some of our studies suffered from such changes. Based on our experiences, we offer two guidelines to support better empirical studies in such situations. First, researchers should perform risk management on their research project and empirical study to be better able to deal with such changes. Second, empirical studies should be planned and executed as fast as possible so that valuable data is obtained before changes render the empirical design and data obsolete. Predefined empirical study "packages" can help in faster execution, but mainly this is a challenge to researchers to design and execute empirical studies more quickly.

Even though we have reported several empirical studies in this work, the number and type of studies limit the generalizeability of the results. These findings should be confirmed by

several additional studies so that we understand the methods and their characteristics better and we can have more confidence in the conclusions that result.

Finally, we believe that this research has potential to make a strong impact on industrial software risk management. The software engineering risk management practice dearly needs improvements and the contributions reported in this work can address many of the burning issues in software risk management.

Software is becoming the predominant technology that supports all aspects of life and society. We hope that the contributions made in this research make the development of software more successful so that people, organizations, and the society as a whole will be able to enjoy the benefits and opportunities software can deliver.

7. References

Anon. 1913, Webster's Revised Unabridged Dictionary.

Anon. 1983, "Risk Assessment Techniques," in Defense Systems Management College Handbook, Defense Systems Management College, pp. iv-1--25, F-1--13.

Anon. 1988, Software Risk Abatement, Department of the Airforce, Adrews Air Force Base, DC, 800-45.

Anon. 1989, Risk Management -- Concepts and Guidance, Defense System Management College, Fort Belvoir, VA, U.S.A., MDA 903-87-C-0781.

Anon. 1992, The American Heritage Dictionary of the English Language, 3 edn, Microsoft Bookshelf/Houghton Mifflin Company, U.S.A.

Anon. 1995a, Merriam-Webster's Collegiate Dictionary, 10 edn, Merriam-Webster, Springfield, MA.

Anon. 1995b, Risk management. Guide to Risk analysis of technological systems, IEC, IEC 60300-3-9:1995.

Anon. 1998a, Emerging Digital Economy, U.S. Department of Commerce.

Anon. 1998b, European Information Technology Observatory 98, European Information Technology Observatory, ISSN 0947 4862.

Anon. 1998c, SPICE: The Theory and Practice of Software Process Improvement and Capability Determination, IEEE Computer Society, Washington, DC.

Anon. 2000a, Managing risk for corporate governance PD 6668:2000.

Anon. 2000b, Project management. Guide to the management of business related project risk, BS 6079-3:2000 edn, BSI.

Abdel-Hamid, T. K. & Madnick, S. E. 1991, Software Project Dynamics, An Integrated Approach, Prentice-Hall, Englewood Cliffs, New Jersey 07632.

Accorsi, R., Apostolakis, G., & Zio, E. 1999, "Prioritizing stakeholder concerns in environmental risk management", Journal of Risk Research, vol. 2, no. 1, pp. 11-29.

Adrion, W. R. 1993, "Research Methodology in Software Engineering, Summary of Dagstuhl Workshop on Future Directions in Software Engineering", Software Engineering Notes, vol. 18, no. 1, pp. 36-37.

Albrecht, A. J. "Measuring Application Development Productivity", in Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium, Oct. 1979 pp. 83-92.

Alho, K., Lassenius, C., & Sulonen, R. 1996, "Process Enactment Support in a Distributed Environment", Computers in Industry, vol. 29, no. 1-2, pp. 5-13.

Allais, M. 1953, "Le comportement de l'homme rationnel devant le risque, critique des postulats et axiomes de l'école Américaine", Econometrica, vol. 21, pp. 503-546.

Alter, S. & Ginzberg, M. 1978, "Managing Uncertainty in MIS Implementation", Sloan Management Review, vol. 20, no. 1, pp. 23-31.

Armenise, P., Ghezzi, C., & Morzenti, A. 1993, "A Survey and Assessment of Software Process Representation Formalisms", International Journal of Software Engineering and Knowledge Engineering, vol. 3, no. 3, pp. 401-426.

Armitage, J. W., Briand, L., Kellner, M. I., Over, J. W., & Phillips, R. W. Software Process Definition Guide: Content of Enactable Software Process Representations. 1995.

Awad, M., Kuusela, J., & Ziegler, J. 1996, Object-Oriented Technology for Real-Time Systems, Prentice Hall, Upper Saddle River.

Babbie, E. R. 1973, Survey Research Methods, Wadsworth Publishing Company, Belmont, CA.

Baker, J. D. 1990, Planet Earth, The View from Space, Harvard University Press, Cambridge, Massachusetts.

Baker, S. L. 1991, "Improving Business Services through the Use of Focus Groups", Reference Quarterly, vol. 30, no. Spring, pp. 377-385.

Bandinelli, S. C., Fuggetta, A., & Ghezzi, C. 1991, "Software process as real-time systems: a case study using high-level Petri nets," in Proceedings of the 1st European Workshop on Software Process Modeling, Milan, Italy, May 1991.

Barghouti, N. S. 1992, "Supporting Cooperation in the **M**ARVEL Process-Centered SDE", ACM SIGSOFT Software Engineering Notes, vol. 17, no. 5, pp. 21-31.

Barki, H., Rivard, S., & Talbot, J. 1993, "Toward an Assessment of Software Development Risk", Journal of Management Information Systems, vol. 10, no. 2, pp. 203-225.

Basili, V. R. 1985, "Quantitative Evaluation of Software Engineering Methodology", in Proceedings of the First Pan Pacific Computer Conference.

Basili, V. R. "Software Development: A Paradigm for the Future", in Proceedings of the 13th Annual Computer Software and Applications Conference (COMPSAC) IEEE Computer Society Press, Washington, DC, pp. 471-485.

Basili, V. R. 1992, Software Modeling and Measurement: The Goal/Question/Metric Paradigm, University of Maryland, College Park, MD, CS-TR-2956.

Basili, V. R. 1993, "The Experience Factory and its Relationship to Other Improvement Paradigms," in Proceedings of the 4th European Software Engineering Conference, Springer-Verlag.

Basili, V. R. 1996, "The Role of Experimentation in Software Engineering: Past, Current, and Future", in Proceedings of the 18th International Conference on Software Engineering IEEE Computer Society, Washington, DC, pp. 442-449.

Basili, V. R. & Caldiera, G. "Methodological and Architectural Issues in the Experience Factory", in Proceedings of the 16th Annual Software Engineering Workshop NASA, Greenbelt, MD, pp. 17-28.

Basili, V. R., Caldiera, G., & Cantone, G. 1991, A Reference Architecture for the Component Factory, University of Maryland, College Park, MD, UMIACS-TR-91-24.

Basili, V. R., Caldiera, G., & Cantone, G. 1992a, "A Reference Architecture for the Component Factory", ACM Transactions on Software Engineering and Methodology, vol. 1, no. 1, pp. 53-80.

Basili, V. R., Caldiera, G., McGarry, F., Pajerski, R., Page, G., & Waligora, S. "The Software Engineering Laboratory - an Operational Software Experience Factory", in Proceedings of the International Conference on Software Engineering, May 1992 IEEE Computer Society Press, Washington, DC., pp. 370-381.

Basili, V. R., Caldiera, G., & Rombach, H. D. 1994a, "Goal Question Metric Paradigm," in Encyclopedia of Software Engineering, vol. 1 J. J. Marciniak, ed., John Wiley & Sons, New York, pp. 528-532.

Basili, V. R., Caldiera, G., & Rombach, H. D. 1994b, "The Experience Factory," in Encyclopedia of Software Engineering, vol. 1 J. J. Marciniak, ed., John Wiley & Sons, New York, pp. 470-476.

Basili, V. R. & Green, S. 1994, "Software Process Evolution at the SEL", IEEE Software, vol. 11, no. 4, pp. 58-66.

Basili, V. R. & Rombach, H. D. "Tailoring the Software Process to Project Goals and Environments", in Proceedings of the 9th International Conference on Software Engineering IEEE Computer Society Press, pp. 345-357.

Basili, V. R. & Rombach, H. D. 1988, "The TAME Project: Towards Improvement-Oriented Software Environments", IEEE Transactions on Software Engineering, vol. 14, no. 6, pp. 753-778.

Basili, V. R. & Rombach, H. D. 1991, "Support for comprehensive reuse", Software Engineering Journal, vol. 6, no. September, pp. 303-316.

Basili, V. R., Selby, R. W., & Hutchens, D. H. 1986, "Experimentation in Software Engineering", IEEE Transactions on Software Engineering, vol. 12, no. 7, pp. 758-773.

Basili, V. R. & Weiss, D. M. 1984, "A Methodology for Collecting Valid Software Engineering Data", IEEE Transactions on Software Engineering, vol. 10, no. 6, pp. 728-738.

Basili, V. R., Zelkowitz, M. V., McGarry, F., Page, J., Waligora, S., & Pajerski, R. 1995, "SEL's Software Process Improvement Program", IEEE Software, vol. 12, no. November, pp. 83-87.

Bean, T. J. & Gros, J. G. 1992, "R&D Benchmarking at AT&T", Research & Technology Management, vol. 35, pp. 32-37.

Benbasat, I. 1999, "Empirical Research in Information Systems: The Practice of Relevance", MIS Quarterly, vol. 23, no. 1, pp. 3-16.

Benbasat, I. & Nault, B. 1990, "An evaluation of empirical research in managerial support systems", Decision Support Systems, vol. 6, no. 2, pp. 203-226.

Bendell, T., Boulter, L., & Gatfort, G. 1997, The Benchmarking Workout, Pitman Pub Ltd.

Bender, M. J., Swanson, S., & Robinson, R. "On the role of fuzzy decision support for risk communication among stakeholders", in IEEE International Conference on Systems, Man, and Cybernetics pp. 317-322.

Bernoulli, D. Specimen Theoriae Novae de Mensura Sortis. 1738.

Bernoulli, D. 1954, "Exposition of New Theory on the Measurement of Risk", Econometrica, vol. 22, pp. 23-36.

Bernstein, P. L. 1996, Against the Gods, John Wiley & Sons, New York.

Berny, J. & Townsend, P. R. F. 1993, "Macrosimulation of project risks -- a practical way forward", International Journal of Project Management, vol. 11, no. 4, pp. 201-208.

Bezirkan, A. & Mulazzani, M. "Experiences with Risk Management in a Large Multi-Site Project", in Proceedings of the Third SEI Conference on Software Risk Management SEI, Pittsburgh, PA.

Bhandari, I., Halliday, M., Tarver, E., Brown, D., Chaar, J., & Chillarege, R. 1993, "A Case Study of Software Process Improvement During Development", IEEE Transactions on Software Engineering, vol. 19, no. 12, pp. 1157-1170.

Boehm, B. W. 1981, Software Engineering Economics, Prentice Hall, Englewood Cliffs, N.J.

Boehm, B. W. 1987, "Industrial Software Metrics Top 10 List", IEEE Software, vol. 4, no. September, pp. 84-85.

Boehm, B. W. 1988, "A Spiral Model of Software Development and Enhancement", IEEE Computer, vol. 21, no. 5, pp. 61-72.

Boehm, B. W. 1989, Tutorial: Software Risk Management, IEEE Computer Society Press.

Boehm, B. W. 1991, "Software Risk Management: Principles and Practices", IEEE Software, vol. 8, no. 1, pp. 32-41.

Boehm, B. W. 1992, "Risk Control", American Programmer, vol. 5, no. September, pp. 36-43.

Boehm, B. W. & Bose P. "A Collaborative Spiral Software Process Model Based on Theory W", in Proceedings of the 3rd International Conference on the Software Process IEEE Computer Society, Washington, DC.

Boehm, B. W., Clark, B., Horowitz, E., Westland, C., Madachy, R., & Selby, R. W. 1995, "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0," in Ann. Softw. Eng., vol. 1 pp. 57-94.

Boehm, B. W. & Ross R 1989, "Theory W Software Project Management: Principles and Examples", IEEE Transactions on Software Engineering no. July, pp. 902-916.

Bollinger, T. B. & McGowan, C. 1991, "A Critical Look at Software Cabability Evaluations", IEEE Software, vol. 8, no. July, pp. 25-41.

Bowers, J. A. 1994, "Data for project risk analyses", International Journal of Project Management, vol. 12, no. 1, pp. 9-16.

Brandl, D. 1991, "Modeling and Describing Really Complex Processes", Texas Instruments Technical Journal, vol. May-June, pp. 21-27.

Brassard, M. & Ritter, D. 1994, The Memory Jogger, GOAL/QPC.

Briand, L. C., Basili, V. R., & Hetmanski, C. J. 1993a, "Developing Interpretable Models with Optimized Set Reduction for Identifying High-Risk Software Components", IEEE Transactions on Software Engineering , vol. 19, no. 11, pp. 1028-1044.

Briand, L. C., Thomas, W. M., & Hetmanski, C. J. 1993b, "Modeling and Managing Risk Early in Software Development," in Proceedings of the 15th International Conference on Software Engineering, IEEE Computer Society Press, Los Alamitos, CA, pp. 55-65.

Brooks jr., F. P. 1987, "No Silver Bullet: Essence and Accident of software Engineering", IEEE Computer pp. 10-19.

Brown, P. G. 1995, "QFD: Echoing the Voice of the Customer", AT&T Technical Journal no. March/April, pp. 18-32.

Bröckers, A. "Process-Based Software Risk Assessment", in Proceedings of the 4th European Workshop on Software Process Technology W. Schäfer, ed., Springer-Verlag, pp. 9-29.

Campbell, D. T. & Stanley, J. C. 1963, Experimental and Quasi-Experimental Designs for Research, Houghton Mifflin Co., Boston.

Campbell, J. P., Daft, R. L., & Hulin, C. L. 1982, "What to Study: Generating and Developing Research Questions," in Studying Organizations: Innovations in Methodology, Sage Publications, London.

Caplan, M. A. "Risk Management in Practice", in Proceedings of the Third SEI Conference on Software Risk Management SEI, Pittsburgh, PA.

Card, D. 1991, "Understanding Process Improvement", IEEE Software, vol. 8, no. 4, pp. 102-103.

Carr, M. J., Konda, S. L., Monarch, I. A., Ulrich, F. C., & Walker, C. F. 1993, Taxonomy-Based Risk Identification, SEI Technical Report SEI-93-TR-006, Software Engineering Institute, Pittsburgh, PA.

Catron, B. A. & Ray, S. R. 1991, "ALPS: A Language for Processing Specification", International Journal of Computer Integrated Manufacturing, vol. 4, no. 2, pp. 105-113.

Cavaye, A. L. M. 1996, "Case study research: a multi-faceted research approach for IS", Information Systems Journal, vol. 6, pp. 227-242.

Chapman, L. J. & Chapman, J. P. 1969, "Illusory correlation as an obstacle to the use of psychodiagnostic observations", Journal of Abnormal Psychology no. 74, pp. 271-280.

Charette, R. N. 1989, Software Engineering Risk Analysis and Management, McGraw-Hill, New York.

Charette, R. N. 1990, Applications Strategies for Risk Analysis, McGraw-Hill, New York.

Charette, R. N. 1992, "Building Bridges over Intelligent Rivers", American Programmer, vol. 5, no. September, pp. 2-9.

Charette, R. N. 1999, "The Competitive Edge of Risk Entrepreneurs", IT Pro no. July/August, pp. 69-73.

Chen, J.-Y. & Tu, C.-M. 1994, "CSPL: a process-centred environment", Information and Software Technology, vol. 36, no. 1, pp. 3-11.

Cheon, M. J., Grover, V., & Sabherwal, R. 1993, "The evolution of empirical research in IS", Information and Management, vol. 24, no. 5, pp. 107-119.

Chittister, C. & Haimes, Y. Y. 1993, "Risk Associated with Software Development: A Holistic Framework for Assessment and Management", IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no. 3, pp. 710-723.

Chittister, C., Kirkpatrick, R. J., & Van Scoy, R. L. 1992, "Risk Management in Practice", American Programmer, vol. 5, no. September, pp. 30-35.

Christie, A. M. 1993, "A graphical process definition language and its application to a maintenance project", Information and Software Technology, vol. 35, no. 6/7, pp. 364-374.

Christie, A. M. 1994, A Practical Guide to the Technology and Adoptation of Software Process Automation, Software Engineering Institute, Pittsburgh, CMU/SEI-94-TR-007.

Clapp, J. 1993, "Getting Started on Software Metrics", IEEE Software, vol. 10, no. 1, pp. 108-117.

Cohen, J., Chesnick, E. I., & Haran, D. A. 1972, "A confirmation of the inertial-Ψ effect in sequential choice and decision", British Journal of Psychology no. 43, pp. 129-144.

Conover, W. J. 1999, Practical Nonparametric Statistics, 3 edn, John Wiley and Sons.

Conradi, R., Osjord, E., Westby, P. H., & Liu, C. 1991, "Initial software process management in EPOS", Software Engineering Journal, vol. 6, no. September, pp. 275-284.

Conrow, E. H. 2000, Effective Risk Management: Some Keys to Success, Amer Inst of Aeronautics.

Conrow, E. H. & Shishido, P. S. 1997, "Implementing Risk Management on Software Intensive Projects", IEEE Software, vol. 14, no. 3, pp. 83-89.

Converse, J. M. & Presser, S. 1996, Survey Questions, Handcrafting the Standardized Questionnaire, Sage, London.

Cook, T. D. & Campbell, D. T. 1979, Quasi-Experimentation: Design & Analysis Issues for Field Settings, Rand McNally College Pub. Co., Chicago.

Covello, V. T. 1984, "Actual and Perceived Risk: A Review of Literature," in Technological Risk Assessment, P. F. Ricci, L. A. Sagan, & C. G. Whipple, eds., Martinus Nijhoff Publishers, The Hague, pp. 225-245.

Covello, V. T. & Mumpower, J. 1998, "Risk Analysis and Risk Management: A historical Perspective," in Risk Evaluation and Management, V. T. Covello, J. Menkes, & J. Mumpower, eds., Plenum Press, New York, pp. 519-540.

Creswell, J. W. 1994, Research Design: Qualitative & Quantitative Approaches, Sage, Thousand Oaks, CA.

Crouhy, M., Galai, D., & Mark, R. 2001, Risk Management, McGraw-Hill.

Crowley, J. & Silverthorn, M. 1991, "Software Artifacts: Recorded Information in STEP", Texas Instruments Technical Journal, vol. May-June, pp. 38-47.

Culver-Lozo, K. & Gelman, S. 1993, "A Process Definition Methodology for Software Development Organizations," in Proceedings of the seventh International Software Process Workshop, Yountville, California 1991, IEEE Computer Society Press, Los Alamitos, pp. 54-56.

Currie, W. 2000, The Global Information Society, Wiley, London.

Curtis, B. 1980, "Measurement and Experimentation in Software Engineering", Proceedings of the IEEE, vol. 68, no. 9, pp. 1144-1157.

Curtis, B., Kellner, M. I., & Over, J. 1992, "Process Modeling", Communications of the ACM, vol. 35, no. 9, pp. 75-90.

Curtis, B. & Paulk, M. C. 1993, "Creating a software process improvement program", Information and Software Technology, vol. 35, no. 6/7, pp. 381-386.

Daskalantonakis, M. K. 1992, "A Practical View of Software Measurement and Implementation Experiences Within Motorola", IEEE Transactions on Software Engineering, vol. 18, no. 11, pp. 998-1010.

David, F. N. 1978, "Dicing and Gaming (a note on the history of probability)," in Studies in the History of Statistics and Probability, 2 edn, E. S. Pearson & M. Kendall, eds., Charles Griffin & Company Ltd, London, pp. 1-15.

Davis, G. B. 1982, "Strategies for Information Requirements Determination", IBM Systems Journal, vol. 21, no. 1.

Debou, C., Fuchs, N., & Saria, H. 1993, "Selling Believable Technology", IEEE Software, vol. 10, no. November, pp. 11-27.

DeMarco, T. & Lister, T. R. 1987, Peopleware: Productive Projects & Teams, Dorset House Publishing Company.

Derniame, J.-C. & Gruhn, V. 1994, "Development of Process-Centered IPSEs in the ALF Project", Journal of Systems Integration, vol. 4, pp. 127-150.

Deutsch, M. S. 1991, "An Exploratory Analysis Relating the Software Project Management Process to Project Success", IEEE Transactions on Engineering Management, vol. 38, no. 4, pp. 365-375.

Diekman, J. E. 1992, "Risk analysis: lessons learned from artificial intelligence", International Journal of Project Management , vol. 10, no. 2, pp. 75-80.

Dion, R. 1992, "Elements of a Process-Improvement Program", IEEE Software, vol. 9, no. July, pp. 83-85.

DoD 1988, Military Standard, Defense System Software Development, DoD-STD-2167A, Department of Defense, U.S.A., Washington, D.C.

Donaldson, T. & Preston, L. E. 1995, "The stakeholder theory of corporation: concepts, evidence, and implications", Academy of Management Review, vol. 20, no. 1, pp. 65-91.

Dorling, A. 1993, "SPICE: Software Process Improvement and Capability dEtermination", Information and Software Technology, vol. 35, no. 6/7, pp. 404-406.

Dorofee, A. J., Walker, J. A., Alberts, C. J., Higuera, R. P., Murray, T. J., & Williams, R. C. 1996, Continuous Risk Management Guidebook, Software Engineering Institute, Pittsburgh, PA.

Dowson, M. 1987, "Integrated Project Support with ISTAR", IEEE Software, vol. 4, no. November, pp. 6-15.

Draper, N. R. & Lawrence, W. E. 1970, Probability: An Introductory Course, Markham Publishing, Chicago.

Dutton, J. E. 1993, "Commonsense Approach to Process Modeling", IEEE Software, vol. 10, no. July, pp. 56-64.

Edgar, J. D. 1989, "Controlling Murphy: How to Budget for Program Risk (originally presented in Concepts, summer 1982, pages 60-73)," in Tutorial: Software Risk Management, B. W. Boehm, ed., IEEE Computer Society Press, Washington, D.C., pp. 282-291.

Edmunds, H. 1991, The Focus Group Research Handbook, Ntc Business Books.

Einhorn, H. J. & Hogarth, R. M. 1985, "Ambiguity and uncertainty in probabilistic inference", Psychological Review, vol. 92, no. 4, pp. 433-461.

Eisenhardt, K. M. 1989, "Building Theories from Case Study Research", Academy of Management Review, vol. 14, no. 4, pp. 532-550.

Eldred, E. W. & McGrath, M. E. 1997a, "Commercializing New Technology--I", Research & Technology Management, vol. 40, no. 1, pp. 41-47.

Eldred, E. W. & McGrath, M. E. 1997b, "Commercializing New Technology--II", Research & Technology Management, vol. 40, no. 2, pp. 29-33.

Englund, H. 1997, A Case Study to Explore Risk Management Methods, Masters thesis, Kunglika Tekniska Högskolan, Stockholm, Sweden.

Erikkson, I. & McFadden, F. 1993, "Quality function deployment: a tool to improve software quality", Information and Software Technology, vol. 35, no. 9, pp. 491-498.

ESA 1991, ESA Software Engineering Standards, ESA PSS-05-0 Issue 2, 2 edn, European Space Agency, Paris.

Eslinger, S., Ellis, C. M., Hoting, S. K., & Walden, G. F. "PACE System Risk Analysis: An Application", in Proceedings of the Second SEI Conference on Software Risk Management SEI, Pittsburgh, PA.

Fairley, R. E. 1989, "Risk Management: The Key to Successful Software Projects," in Proceedings of the 3rd IFAC/IFIP Workshop, F. J. Mowle & P. F. Elzer, eds., Pergamon, Oxford, pp. 45-50.

Fairley, R. 1994, "Risk Management for Software Projects", IEEE Software, vol. 11, no. May, pp. 57-67.

Feig, B. 1989, "How to Run a Focus Group", American Demographics, vol. 11, no. December, pp. 36-37.

Feiler, P. H. & Humphrey, W. S. 1993, "Software Process Development and Enactment: Concepts and Definitions", in Proceedings of the 2nd International Conference on the Software Process, Berlin 1993 IEEE Computer Society Press, Los Alamitos, CA, pp. 28-40.

Fenton, N. E. 1991, Software Metrics A Rigorous Approach, Chapman & Hall, London.

Fenton, N. E., Pfleeger, S. L., & Glass, R. A. 1994, "Science and Substance: A Challenge to Software Engineers", IEEE Software, vol. 11, no. 4, pp. 86-95.

Fernström, C. 1993, "PROCESS WEAVER: Adding Process Support to UNIX," in Proceedings of the 2nd International Conference on the Software Process, Berlin 1993, IEEE Computer Society Press, Los Alamitos, pp. 12-26.

Festinger, L. 1957, A Theory of Cognitive Dissonance, Stanford University Press, Stanford, CA.

Finkelstein, A., Kramer, J., & Nuseibeh, B. 1994, Software Process Modeling and Technology, John Weiley & Sons.

Fisz, M. 1967, Probability Theory and Mathematical Statistics, 3 edn, John Wiley & Sons, New York.

Fitzgerald, G. 1991, "Validating new information systems techniques: a retrospective analysis," in The Information Systems Research Arena of the 90s, Challenges, Perceptions and Alternative Approaches, H.-E. Nissen, ed., North Holland, Amsterdam, pp. 657-672.

Flowers, S. 1996, Software Failure : Management Failure : Amazing Stories and Cautionary Tales, John Wiley & Son Ltd.

Foo, S.-W. & Muruganantham, A. "Software risk assessment model", in Proceedings of the 2000 IEEE International Conference on Management of Innovation and Technology IEEE, pp. 536-544.

Fortune, J. & Peters, G. 1995, Learning from Failure: The Systems Approach, John Wiley.

Fowler, F. J. jr. & Mangione, T. W. 1990, Standardizing Survey Interviewing. Minimizing Interview-related Error, Sage, Newburry Park.

Frailey, D. J., Bate, R. R., Crowley, J., & Hills, S. 1991, "Modeling Information in a Software Process," in Proceedings of the First International Conference on the Software Process, Redondo Beach, California, October 1991, M. Dowson, ed., IEEE Computer Society Press, Los Alamitos, pp. 60-67.

Freeman, R. E. 1984, Strategic Management: A Stakeholder Approach, Ballinger Publishing, Cambridge, Mass.

Freimut, B., Hartkopf, S., Kaiser, P., Kontio, J., & Kobitzsch, W. "An Industrial Case Study of Implementing Software Risk Management", in Proceedings of the European Software Engineering Conference.

French, S. 1986, Decision Theory: An Introduction to the Mathematics of Rationality, Ellis Horwood, Chichester.

French, S. 1989, Readings in Decision Analysis, Chapman and Hall, London.

Friedman, G. J. "Risk Management", in Proceedings of the Second SEI Conference on Software Risk Management SEI, Pittsburgh, PA.

Friedman, M. & Savage, L. J. 1948, "The Utility Analysis of Choices Involving Risk", Journal of Political Economy, vol. 56, pp. 279-304.

Galliers, R. D. 1985, "In search of a paradigm for information systems research," in Research Methods in Information Systems, North-Holland Publishers, New York, pp. 281-297.

Galliers, R. D. 1991, "Choosing appropriate information systems research approaches: A revised taxonomy," in Information Systems Research: Contemporary Approaches and Emerging Traditions, H.-E. Nissen, H. K. Klein, & R. Hirschheim, eds., Elsevier Science Publishers, pp. 327-345.

Galliers, R. D. 1992, Information Systems Research: Issues, Methods and Practical Guidelines, Blackwell.

Garrabrants, W. M., Ellis III, A. W., Hoffman, L. J., & Kamel, M. 1990, "**CERTS**: A Comparative Evaluation Method for Risk Management Methods and Tools," in Proceedings of the Sixth Annual Computer Security Applications Conference, IEEE Computer Society Press, Los Alamitos, pp. 251-257.

Garrick, B. J. & Gekler, W. C. 1991, The Analysis, Communication, and Perception of Risk, Plenum Press, New York.

Garvey, P. R., Phair, D. J., & Wilson, J. A. 1997, "An Information Architecture for Risk Assessment and Management", IEEE Software, vol. 14, no. 3.

Gemmer, A. 1997, "Risk Management: Moving Beyond Process", IEEE Computer no. May, pp. 33-43.

Gemmer, A. & Koch, P. "Rockwell Case Studies in Risk Management", in Proceedings of the Third SEI Conference on Software Risk Management SEI, Pittsburgh, PA.

Getto, G. & Landes, D. "Risk Management in Complex Project Organizations: A Godfather-driven Approach", in Proceedings of the Project Management Institute (PMI) Conference 99.

Getto, G. & Landes, D. "Systematic Risk Management as a Key Factor in the Management of Project Quality", in Proceedings of the 6. European Conference on Software Quality (ECSQ).

Ghauri, P., Grønhaug, K., & Kristianslund, I. 1995, Research Methods in Business Studies, Prentice Hall, New York.

Glass, R. A. 1995a, "A Structure-Based Critique of Contemporary Computing Research", Journal of Systems and Software, vol. 28, no. 1, pp. 3-7.

Glass, R. A. 1995b, "The Software-Research Crisis", IEEE Software, vol. 12, no. November, pp. 42-47.

Glass, R. A. 1997, The Software Runaways, Prentice Hall.

Glass, R. A. 1998, "Is There Really a Software Crisis?", IEEE Software, vol. 15, no. 1, pp. 104-105.

Gluch, D. P. 1994, A Construct for Describing Software Development Risks, Software Engineering Institute, CMU/SEI/-94-TR-14.

Grady, R. B. 1992, Practical Software Metrics for Project Management and Process Improvement, Prentice-Hall, Englewood Cliffs.

Greer, D., Bustard, D. W., & Sunazuka, T. "Prioritization of System Changes using Cost-Benefit and Risk Assessment", in Prroceedings of the Requirements Engineering Conference pp. 180-187.

Groth, J. C. 1992, "Common-sense Risk Assessment", Management Decision, vol. 30, no. 5, pp. 10-16.

Guba, E. G. & Lincoln, Y. S. 1981, Effective Evaluation: Improving the Usefulness of Evaluation Results through Responsive and Naturalistic Approaches, Jossey-Bass, San Francisco.

Haase, V., Messnarz, R., Koch, G. R., Kugler, H. J., & Decrinis, P. 1994, "Bootstrap: Fine-Tuning Process Assessment", IEEE Software, vol. 11, no. 4, pp. 25-35.

Hakel, M. 1982, Making It Happen: Designing Research with Implementation in Mind, Sage Publications, Beverly Hills.

Hall, E. M. "Evolution of Essential Risk Management Technology", in Proceedings of the Third SEI Conference on Software Risk Management SEI, Pittsburgh, PA.

Hall, E. M. 1995, Proactive Risk Management Methods for Software Engineering Excellence, Florida Institute of Technology.

Hall, E. M. Email correspondence. Kontio, Jyrki. 1996. 4-9-1996.

Hall, E. M. 1998, Managing Risk: Methods for Software Systems Development, Addison-Wesley Pub Co., Reading.

Hall, T. & Fenton, N. E. 1997, "Implementing Effective Software Metrics Programs", IEEE Software, vol. 14, no. 2, pp. 55-65.

Hammer, M. 1996, Beyond Reengineering, HarperCollins Business, London.

Hammer, M. & Champy, J. 1993, Reengineering the Corporation - A Manifesto for Business Revolution, HarperBusiness, New York.

Harel, D. 1987, "Statecharts: a Visual Formalism for Complex Systems", Science of Computer Programming, vol. 8, pp. 231-274.

Harmon-Jones, E. & Mills, J. 1999, "An Introduction to Cognitive Dissonance Theory and an Overview of Current Perspectives on the Theory," in Cognitive Dissonance: Progress on a Pivotal Theory in Social Psychology, E. Harmon-Jones & J. Mills, eds..

Harrington, H. J. 1991, Business Process Improvement. The Breakthrough Strategy or Total Quality, Productivity and Competitiveness, McGraw-Hill.

Haytin, D. L. 1988, The Validity of Case Studies. Deviance and Self-destruction, Peter Lang, New York.

Hefner, R. "Experience with Applying SEI's Risk Taxonomy", in Proceedings of the Third SEI Conference on Software Risk Management SEI, Pittsburgh, PA.

Hogarth, R. M. 1987, Judgment and Choice, John Wiley & Sons, New York.

Honkonen, M. 1999, The Use of Risk Management Methods in Software Projects, Masters Masters Thesis, University of Jyväskylä.

Huckvale, T. & Ould, M. A. Process Modelling - Why, What and How. Software Assistance for Business Re-Engineering . 1993.

Humphrey, W. S. & Kellner, M. I. 1989, "Software Process Modeling: Principles of Entity Process Models," in Proceedings of the 11th International Conference on Software Engineering, ACM, pp. 331-342.

Humphrey, W. S. & Sweet, W. L. 1987, A Method for Assessing the Software Engineering Capability of Contractors, Technical Report CMU/SEI-87-TR-23, ESD/TR-87-186, Software Engineering Institute, Carnegie Mellon University, Pittsburgh.

HUT, Tik-76.115 Software Project, http://mordor.cs.hut.fi/tik-76.115/98-99/projects.htm, 2001

IEEE 1987, IEEE Standard for Software Project Management Plans, Std 1058.1-1987, IEEE, New York.

IEEE 1992, IEEE Standard for a Software Quality Metrics Methodology, IEEE Std 1061 - 1992, IEEE, New York.

IEEE. Managing Risk. IEEE Software 14[3]. 1997.

ISO 1987, ISO 9001, Quality systems -- Model for quality assurance in design/development, production, installation and servicing, International Standards Organization.

ISO 1991a, Information Technology Software Life Cycle Process ISO/IEC(JTC1)-SC7.

ISO 1991b, ISO 9000-3, Guidelines for the application of ISO 9001 to the development, supply and maintenance of software, ISO 9000-3:1991(E), International Standards Organization.

ISO. SPICE: Baseline Practices Guide, an unfinished draft of a standard being developed for ISO, version 1.00. 1994.

ISO 1996, Information technology. Guidelines for the management of IT security. Concepts and models for IT Security, ISO/IEC TR 13335-1:1996 edn.

ISO 1998a, Information technology. Guidelines for the management of IT security. Techniques for the management of IT security, ISO/IEC TR 13335-3:1996 edn.

ISO 1998b, Information technology. Software process assessment. A reference model for processes and process capability, ISO/IEC TR 15504-2:1998 edn, International Standards Organization.

ISO 1998c, Information technology. Software process assessment. Concepts and introductory guide, ISO/IEC, ISO/IEC 15504-1:1998.

ISO 1999, Information security management. Specification for information security management systems, ISO, BS 7799-2:1999.

ISO. RA Software Tool. 2000.

Jeffery, D. R. & Votta, L. G. 1999, "Empirical Software Engineering", IEEE Transactions on Software Engineering, vol. 25, no. 4, pp. 435-437.

Jick, T. D. 1979, "Mixing Qualitative and Quantitative Methods: Triangulation in Action", Administrative Science Quarterly, vol. 24, no. December, pp. 602-611.

Jones, C. 1994, Assessment and Control of Software Risks, Yourdon Press, Englewood Cliffs.

Judd, C. M., Smith, E. R., & Kidder, L. H. 1991, Research Methods in Social Relations, 6 edn, Harcourt Brace Jovanovich College Publishers, Fort Worth.

Juristo, N. & Moreno, A. M. 2001, Basics of Software Engineering Experimentation, Kluwer Academic Publishers.

Järvenpää, S. L. 1988, "The Importance of Laboratory Experimentation in IS Research", Communications of the ACM, vol. 31, no. 12, pp. 1502-1504.

Järvenpää, S. L., Dickson, G. W., & DeSanctis, G. 1985, "Methodological Issues in Experimental IS Research", MIS Quarterly, vol. 9, no. 2, pp. 141-156.

Kahneman, D., Slovic, P., & Tversky, A. 1982, Judgment Under Uncertainty: Heuristics and Biases, Cambridge University Press, New York.

Kahneman, D. & Tversky, A. 1973, "On the psychology of prediction", Psych.Rev. no. 80, pp. 237-251.

Kaiser, G. E., Barghouti, N. S., & Sokolsky, M. "Preliminary Experience with Process Modeling in the Marvel SDE Kernel", in Proceedings IEEE 23rd Hawaii International Conference on System Sciences IEEE Comput. Soc. Press, Los Alamitos, CA, pp. 131-140.

Kalliomäki, S. & Känsälä, K. VTT Risk Tool 1.1, User's Instructions. 1993.

Kaltio, T. 2001, Software Process Asset Management and Deployment in a Multi-Site Organization, Finnish Academies of Technology.

Karolak, D. W. 1996, Software Engineering Risk Management, IEEE, Washington, DC.

Katzer, J., Cook, K. H., & Crouch, W. W. 1991, Evaluating Information. A Guide for Users of Social Science Research, McGraw-Hill, New York.

Keen, P. G. W. 1991, "Relevance and Rigor in Information Systems Reearch: Improving Quality, Confidence, Cohesion and Impact," H.-E. Nissen, H. K. Klein, & R. Hirschheim, eds., Elsevier Science Publishers, Amsterdam, pp. 27-49.

Keeney, R. L. & Raiffa, H. 1976, Decision with Multiple Objectives: Preferences and Value Tradeoffs, John Wiley & Sons, New York.

Kellner, M. I. 1989, "Representation Formalisms for Software Process Modeling", ACM SIGSOFT Software Engineering Notes, vol. 14, no. 4, pp. 93-96.

Kellner, M. I. 1996, "A Method for Designing, Defining, and Evolving Software Processes", in Proceedings of the 1996 SEPG Conference, Atlantic City SEI, Pittsburgh, PA.

Kellner, M. I. & Rombach, H. D. 1991, "Session Summary: Comparison of Software Process Descriptions," in Proceedings of the 6th International Software Process Workshop, Hakodate, Japan, 1990, IEEE Computer Society, Los Alamitos, CA, pp. 7-18.

Kemerer, C. F. 1993, "Reliability of Function Points Measurement", Communications of the ACM, vol. 36, no. 2, pp. 85-97.

Keynes, J. M. 1921, Treatise on Probability, Macmillan, London.

Kirkpatrick, R. J., Walker, J. A., & Firth, R. 1994, "Software Development and Risk Management: An SEI Appraisal," in Annual Technical Review '92, SEI -92-TechReview.

Kitchenham, B. & Linkman, S. 1997, "Estimates, Uncertainty, and Risk", IEEE Software, vol. 14, no. 3, pp. 69-74.

Kitchenham, B., Pickard, L., & Pfleeger, S. L. 1995, "Case Studies for Method and Tool Evaluation", IEEE Software, vol. 12, no. July, pp. 52-62.

Koch, G. R. 1993, "Process Assessment: The 'BOOTSTRAP' Approach", Information and Software Technology, vol. 35, no. 6/7, pp. 387-403.

Kontio, J. 1994a, Software Engineering Risk Management: A Technology Review Report, Nokia Research Center, Helsinki, Finland, PI_4.1.

Kontio, J. Software Process Modeling: A Technology Review and Notation Analysis, Nokia Research Center Report, Process Improvement Project deliverable PI_1.6. 1994b.

Kontio, Jyrki, IWSED-95 Web pages, http://www.cs.umd.edu/projects/SoftEng/ESEG/iwsed/iwsed95/, 1995a

Kontio, J. 1995b, Promises: A Framework for Utilizing Process Models in Process Asset Management, Licentiate in Technology, Helsinki University of Technology.

Kontio, J. Riskit: An Analytical Model for Risk Management. 1995c. 26-10-1995c.

Kontio, J. Definition and Validation of a Risk Management Method, a Ph.D. proposal. 1996a.

Kontio, J. 1996b, Definition of the Riskit Method, Version 0.10, University of Maryland, College Park.

Kontio, J. 1997, The Riskit Method for Software Risk Management, version 1.00, University of Maryland, College Park, MD, CS-TR-3782 / UMIACS-TR-97-38.

Kontio, J. 1998, "A Process Engineering Framework," in Advances in Computers, vol. 45 M. V. Zelkowitz, ed., Academic Press, pp. 36-108.

Kontio, J. "Risk Management in Software Development: A Technology Overview and the Riskit Method, Tutorial", in Proceedings of the ICSE 99 Conference.

Kontio, J. & Basili, V. R. "Risk Knowledge Capture in the Riskit Method", in Proceedings of the 21st Software Engineering Workshop NASA, Greenbelt, Maryland.

Kontio, J. & Basili, V. R. "Empirical Evaluation of a Risk Management Method", in Proceedings of the SEI Conference on Risk Management Software Engineering Institute, Pittsburgh, PA.

Kontio, J., Englund, H., & Basili, V. R. 1996, Experiences from an Exploratory Case Study with a Software Risk Management Method, University of Maryland, College Park, Maryland, CS-TR-3705.

Kontio, J., Getto, G., & Landes, D. "Experiences in improving risk management processes using the concepts of the Riskit method", in Proceedings of the Sixth International Symposium on the Foundations of Software Engineering (FSE-6) pp. 163-174.

Kuvaja, P., Bicego, A., Cachia, R. M., Haase, V., Koch, G. R., Maiocchi, M., Messnarz, R., Saukkonen, S., Schynoll, W., & Similä, J. 1993, "Bootstrap -- A European Assessment Methodology", IEEE Software, vol. 10, no. April.

Känsälä, K. An Introduction to RiskMethod. 1993.

Känsälä, K. 1997, "Integrating Risk Assessment with Cost Estimation", IEEE Software, vol. 14, no. 3, pp. 61-67.

Lacity, M. C. & Hirschheim, R. 1995, "Benchmarking as a strategy for managing conflicting stakeholder perceptions of information systems", Journal of Strategic Information Systems, vol. 4, no. 2, pp. 165-185.

Lai, R. C. T. Process Definition and Process Modeling Methods, research report SPC-91084-N by Software Productivity Consortium. 1991.

Laitinen, L., Kalliomäki, S., & Känsälä, K. 1993, Ohjelmistoprojektien Riskitekijät, Tutkimusselostus N:o L-4, VTT, Tietojenkäsittelytekniikan Laboratorio, Helsinki.

Langer, E. J. 1975, "The Illusion of Control", Journal of Personality and Social Psychology, vol. 32, pp. 311-328.

Lee, A. S. 1989, "A scientific methodology for MIS case studies", MIS Quarterly, vol. 13, no. 1, pp. 33-50.

Lehman, M. M. 1986, "Approach to a Disciplined Development Process - The ISTAR Integrated Project Support Environment", ACM SIGSOFT Software Engineering Notes, vol. 11, no. 4, pp. 28-33.

Lyytinen, K. 1988, "Stakeholders, IS failures and soft systems methodology: an assessment", Journal of Applied Systems Analysis , vol. 15, pp. 61-81.

Lyytinen, K. & Hirschheim, R. 1987, "Information systems Failures - a Survey and Classification of the Empirical Literature", Oxford Surveys in Information Technology pp. 257-309.

Lyytinen, K., Mathiassen, L., & Ropponen, J. 1993, Software Risk Management as Satisficing Behavior: an Environmental Model, University of Jyväskylä, Department of Computer Science and Information Systems, Jyväskylä.

Lyytinen, K., Mathiassen, L., & Ropponen, J. 1996, "A Framework for Software Risk Management", Scandinavian Journal of Information Systems, vol. 8, no. 1, pp. 53-68.

Lyytinen, K., Mathiassen, L., & Ropponen, J. 1998, "Attention Shaping and Software Risk - A Categorical Analysis of Four Classical Approaches", Information Systems Research, vol. 9, no. 3, pp. 233-255.

Machina, M. 1987, "Choice under uncertainty: Problems solved and unsolved", The Journal of Perspectives, vol. 1, pp. 121-154.

Madachy, R. J. 1997, "Heuristic Risk Assessment Using Cost Factors", IEEE Software, vol. 14, no. 3, pp. 51-59.

Madhavji, N. H., Höltje, D., Hong, W., & Bruckhaus, T. F. "Elicit: A Method for Eliciting Process Models", in Proceedings of the Third International Conference on the Software Process IEEE Computer Society Press, Washington, pp. 111-122.

March, S. T. & Smith, G. F. 1995, "Design and natural science research on information technology", Decision Support Systems, vol. 15, pp. 251-266.

Mas-Colell, A., Whinston, M. D., & Green, J. R. 1995, Microeconomic theory, Oxford University Press, New York.

Mason, R. O. 1989, "MIS Experiments: A Pragmatic Perspective," in The Information Systems Research Challenge: Experimental Research Methods, vol. 2 I. Benbasat, ed., Harvard Business School, Boston, pp. 3-20.

McCaugherty, D. "Criticality Analysis and Risk Assessment (CARA)", in Proceedings of the Third Annual Conference on Software Acquisition Management Technology Training Corporation, Washington, DC, pp. 306-340.

McFarlan, F. W. 1974, "Portfolio approach to information systems", Harvard Business Review no. January/February, pp. 142-150.

McFarlan, F. W. 1984, The Information Systems Research Challenge, Harvard Business School Press.

McGarry, F., Pajerski, R., Page, G., Waligora, S., Basili, V. R., & Zelkowitz, M. V. 1994, Software Process Improvement in the NASA Software Engineering Laboratory, Software Engineering Institute, Pittsburgh, PA, CMU/SEI-94-TR-22.

McGrath, J. E. & Martin, J. 1982, Judgment Calls in Research, Sage Publications, Beverly Hills.

Merkhofer, M. W. 1987, "Quantifying Judgemental Uncertainty: Methodology, Experiences, and Insights", IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-17, no. 5, pp. 741-752.

Meyers, D. J. & Trbovich, D. R. "One Project's Approach to Software Risk Management", in Proceedings of the Second SEI Conference on Software Risk Management SEI, Pittsburgh, PA.

Michaels, J. V. 1996, Technical Risk Management, Prentice Hall, Upper Saddle River, NJ.

Microsoft. VISIO. Technical[2000]. 2000. Microsoft Corp.

Mills, G. E. 1999, Action Research: A Guide for the Teacher Researcher, Prentice Hall.

Minkowitz, C. 1993, "Formal process modeling", Information and Software Technology, vol. 35, no. 11/12, pp. 659-667.

Monarch, I. A., Konda, S. L., & Carr, M. J. "Software Engineering Risk Repository", in Proceedings of the 1996 SEPG Conference Software Engineering Institute, PIttsburgh, PA.

Morin, J.-M. "Risk Driven Project Management: A Practical Approach", in Proceedings of the Second SEI Conference on Software Risk Management SEI, Pittsburgh.

Moynihan, T. 1997, "How Experienced Project Managers Assess Risk", IEEE Software, vol. 14, no. 3, pp. 35-41.

NASA, Software Engineering Laboratory World Wide Web home page: http://fdd.gsfc.nasa.gov/seltext.html, 2001

Neter, J. & Waksberg, J. 1964, "A Study of Response Errors in Expenditure Data from Household Interviews", Journal of the American Statistical Association, vol. 59, pp. 18-55.

Newland, K., Mays, M., Chapman, C., Gerdes, R., Hillson, D., Rawlings, P., Norris, C., & Vose, D. 1997, Project Risk Analysis and Management Guide, The Association for Project Management, Norwich Norfolk, U.K.

Nissen, H.-E., Klein, H. K., & Hirschheim, R. 1991, Information Systems Research: Contemporary Approaches and Emerging Traditions, Elsevier Science Publishers, Amsterdam.

Nolan, R. 1973, "Managing the Computer Resource: A Stage Hypothesis", Communications of the ACM, vol. 16, no. 7, pp. 399-405.

Nolan, R. 1979, "Managing the Crises in data processing", Harvard Business Review no. March/April, pp. 115-126.

Nukari, J. & Forsell, M. 1999, Suomen ohjelmistoteollisuuden kasvun strategiat ja haasteet, Teknologian kehittämiskeskus, Teknologiakatsaus 67/99.

Nunamaker jr., J. F., Chen, M., & Purdin, T. D. M. 1991, "System Development in Information Systems Research", Journal of Management Information Systems, vol. 7, no. 3, pp. 89-106.

Offen, R. J. & Jeffery, D. R. 1997, "Establishing Software Measurement Programs", IEEE Software, vol. 14, no. 2, pp. 45-53.

Oivo, M. & Basili, V. R. 1992, "Representing Software Engineering Models: The TAME Goal Oriented Approach", IEEE Transactions on Software Engineering, vol. 18, no. 10, pp. 886-898.

Ould, M. A. Process Modeling with RADS. 1992. Praxis Technology/TRMC.

Ould, M. A. 1995, Business Processes, John Wiley and Sons.

Pandelios, G. "Software Risk Evaluation and Team Risk Management", in Tutorial Presentations at the 1996 SEPG Conference Software Engineering Institute, Pittsburgh, PA.

Pandelios, G., Rumsey, T. P., & Dorofee, A. J. 1996, "Using Risk Management for Software Process Improvement", in Proceedings of the 1996 SEPG Conference SEI, Pittsburgh.

Papazafeiropoulou, A., Pouloudi, A., & Currie, W. L. "Applying the stakeholder concept to electronic commerce: extending previous research to guide government policy makers", in Proceedings of the 34th Annual Hawaii International Conference on System Sciences IEEE, pp. 1719-1728.

Patton, M. Q. 1990, Qualitative Evaluation and Research Methods, 2 edn, SAGE Publications.

Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. 1993a, Capability Maturity Model for Software, Version 1.1, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, SEI-93-TR-024.

Paulk, M. C., Weber, C. V., Garcia, S. M., Chrissis, M. B., & Bush, M. 1993b, Key Practices of the Capability Maturity Model, Software Engineering Institute, Pittsburgh, PA, SEI-93-TR-025.

Petroski, H. 1985, To Engineer is Human: The Role of Failure in Successful Design, Macmillan.

Pfleeger, S. L. 1993, "Lessons Learned in Building a Corporate Metrics Program", IEEE Software, vol. 10, no. May, pp. 67-74.

Pfleeger, S. L. 1995, "Maturity, models and goals: How to build a metrics plan", Journal of Systems and Software, vol. 31, pp. 143-155.

Pfleeger, S. L. 1997, "Experimentation in Software Engineering," in Advances in Computers, vol. 44 M. V. Zelkowitz, ed., Academic Press, London.

Pfleeger, S. L. & McGowan, C. 1990, "Software Metrics in the Process Maturity Framework", Journal of Systems and Software, vol. 12, pp. 255-261.

Porter, M. E. 1985, Competitive Advantage, Free Press, New York.

Potts, C. 1993, "Software Engineering Research Revisited", IEEE Software no. Sept, pp. 19-28.

Pouloudi, A. "Aspects of the Stakeholder Concept and their Implications for Information Systems Development", in Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences IEEE.

Pralahad, C. K. & Hamel, G. 1990, "The Core Competencies of the Corporation", Harvard Business Review no. May-June, pp. 79-91.

Pressman, R. S. 2000, Software Engineering -- A Practitioner's Approach, 5 edn, McGraw-Hill, London.

QPR. ProcessGuide. [5]. 2001. QPR Software Plc.

Quattrone, G. A. & Tversky, A. 1984, "Causal versus Diagnostic Contingencies: On Self Deception and on the Voter's Illusion", Journal of Personality and Social Psychology, vol. 46, no. 2, pp. 237-248.

R & D-Ware Oy, Riskit web pages, http://www.rdware.com/Riskit, 2001

Radice, R. A., Roth, N. K., O'Hara, A. C., & Ciarfella, W. A. 1985, "A Programming Process Architecture", IBM Systems Journal, vol. 24, no. 2, pp. 79-90.

Rescher, N. 1983, Risk: A Philosophical Introduction to the Theory of Risk Evaluation and Management, University Press of America, Lanham, MD.

Ricci, P. F., Sagan, L. A., & Whipple, C. G. 1981, Technological Risk Assessment, Martinus Nijhoff Publishers.

Ritchie, B. & Marshall, D. 1993, Business Risk Management, Chapman & Hall, London.

Rombach, H. D. "An Experimental Process Modeling Language", in Proceedings of the Conference on Software Maintenance pp. 95-96.

Rombach, H. D. "Practical Benefits of Goal Oriented Measurement", in Proceedings of the 7th Software Reliability and Metrics Conference pp. 217-235.

Rook, P. & Cowderoy, A. "Software Risk Management Practice in Industry and Support for Risk Engineering in the GOAL Toolset", in Proceedings of the Second SEI Conference on Software Risk Management SEI, Pittsburgh.

Ropponen, J. 1993, Risk Management in Information System Development, University of Jyväskylä, Department of Computer Science and Information Systems, Jyväskylä, TR-3.

Ropponen, J. 1999, Software Risk Management - Foundations, Principles and Empirical Findings, University of Jyväskylä.

Ropponen, J. & Lyytinen, K. 2000, "Components of software development risk: how to address them? A project manager survey", IEEE Transactions on Software Engineering, vol. 26, no. 2, pp. 98-112.

Rothfeder, J. 1988, "It's Late, Costly Incompetent -- But Try Firing a Computer System", Business Week, vol. November 7, pp. 164-165.

Rowe, W. D. 1977, An Anatomy of Risk, John Wiley & Sons, New York.

Roy, G. G. & Woodings, T. L. "A framework for risk analysis in software engineering", in Proceedings of the Seventh Asia-Pacific Software Engineering Conference IEEE, pp. 441-445.

Rudestam, K. E. & Newton, R. R. 1992, Surviving Your Dissertation. A Comprehensive Guide to Content and Process, Sage Publications, Newburry Park.

Rumbaugh, J., Blaha, M., Premerlani, W., & Lorensen, W. 1991, Object-Oriented Modeling and Design, Prentice Hall, Englewood Cliffs, New Jersey.

Saarinen, T. 1993, Success of Information Systems, Helsinki School of Economics and Business Administration.

Saaty, T. L. 1982, Decision Making for Leaders, Lifetime Learning Publications, Belmont, California.

Saaty, T. L. 1990, The Analytic Hierarchy Process, McGraw-Hill, New York.

Saaty, T. L. 1992, "Analytic Hierarchy," in Encyclopedia of Science & Technology, McGraw-Hill, pp. 559-563.

Sambursky, S. 1956, "On the Possible and Probably in Ancient Greece", Osiris, vol. 12, pp. 35-48.

Scholtes, P. R., Joiner, B. L., & Streiber, B. J. 1996, The Team Handbook, 2 edn, Joiner Associates, Madison, WI.

Schuman, H. & Presser, S. 1981, Questions and Answers in Attitude Surveys: Experiments on Question Form, Wording, and Context, Academic Press, New York.

Seaman, C. & Basili, V. R. "OPT: An approach to organizational and process improvement".

SEI 1993, Proceedings of the Second SEI Conference on Software Risk Management, Software Engineering Institute, Pittsburgh, PA.

SEI 1994, Proceedings of the Third SEI Conference on Software Risk Management, Software Engineering Institute, Pittsburgh.

SEI 1995, Proceedings of the Fourth SEI Conference on Software Risk Management, Software Engineering Institute, Pittsburgh, PA.

SEI 1997, Proceedings of the Fifth SEI Conference on Software Risk Management, Software Engineering Institute, Pittsburgh, PA.

Selig, B. J. 1993, "Technology's Impact on Risk", Risk Management no. November, pp. 58-63.

Shafir, E. & Tversky, A. 1992, "Thinking Through Uncertainty: Nonconsequential Reasoning and Choice", Cognitive Psychology, vol. 24, pp. 449-474.

Shapiro, C. & Varian, H. R. 1998, Information Rules : A Strategic Guide to the Network Economy, Harvard Business School Press.

Shefrin, H. & Thaler, R. H. 1988, "The Behavioral Life-Cycle Hypothesis", Economic Inquiry, vol. 26, no. October, pp. 609-643.

Shiller, R. J. 1999, "Human Behavior and the Efficiency of the Financial System," in Handbook of Macroeconomics, Elsevier Science Ltd.

Simister, S. J. 1994, "Usage and benefits of project risk analysis and management", International Journal of Project Management , vol. 12, no. 1, pp. 5-8.

Simon, H. A. 1979, "Rational Decision Making in Business Organizations", The American Economic Review, vol. 69, no. 4, pp. 493-513.

Simon, J. L. 1969, Basic Research Methods in Social Science, Random House, New York.

Singh, B. & Rein, G. L. 1992, Role Interaction Nets (RINs): A Process Description Formalism, MCC, Austin, Texas, CT-083-92.

Singh, R. Information Technology Software Life-Cycle Process, ISO/IEC (JTC1)-SC7. 1991.

Singpurwalla, N. D. & Wilson, S. P. 1999, Statistical Methods in Software Engineering: Reliability and Risk, Pringer.

Sisti, F. J. & Joseph, S. 1994, Software Risk Evaluation Method Version 1.0, Software Engineering Institute, Pittsburgh, CMU/SEI-94-TR-19.

Skinner, B. F. 1948, "Superstition in the Pigeon", Journal of Experimental Psychology, vol. 38, pp. 168-172.

Sommerville, I. 1995, Software Engineering, 5 edn, Addison-Wesley.

Speaker, W. V. "Implementing a Risk Management Methodology: Step 1 - Defining the Process", in Proceedings of the Second SEI Conference on Software Risk Management SEI, Pittsburgh, PA.

Spendolini, M. J. 1992, The Benchmarking Book, AMACOM, New York.

Stake, R. E. 1995, The Art of Case Study Research, Sage Publications, Thousand Oaks.

Stalk, G., Evans, P., & Shulman, L. E. 1992, "Competing on Capabilities: The New Rules of Corporate Strategy", Harvard Business Review, vol. March-April, pp. 57-69.

Stamatis, D. H. 1995, Failure Mode and Effect Analysis : FMEA from Theory to Execution, American Society for Quality.

Stewart, D. W. & Shamdasani, P. N. 1990, Focus Groups: Theory and Practice, Sage, Newbury Park, CA.

Straub, D. W. 1989, "Validating Instruments in MIS Research", MIS Quarterly no. June, pp. 147-165.

Stringer, E. T. 1999, Action Research, 2 edn, Corwin Pr.

Struik, D. J. 1987, A Concise History of Mathematics, 4 edn, Dover Publications, New York.

Sudgen, R. 1986, "New Developments in the Theory of Choice Under Uncertainty", Bulletin of Economic Research, vol. 38.

Sudman, S. & Bradburn, N. M. 1982, Asking Questions, Jossey-Bass Publishers, San Francisco.

Swanson, E. B. & Beath, C. M. 1988, "The use of case study data in software management research", Journal of Systems and Software, vol. 8, pp. 63-71.

Taylor, R. N., Belz, F. C., Clarke, L. A., Osterweil, L. J., Selby, R. W., Wileden, J. C., Wolf, A. L., & Young Michael 1993, "Foundations for the Arcadia Environment Architecture", ACM SIGSOFT Software Engineering Notes, vol. 13, no. 5, pp. 1-13.

Taylor, S. J. & Bogdan, R. 1984, Introduction to Qualitative Research Methods, John Wiley and Sons.

Templeton, J. F. 1994, The Focus Group: A Strategic Guide to Organizing, Conducting and Analyzing the Focus Group Interview, McGraw-Hill Professional Publishing.

Thomas, M. & McGarry, F. 1994, "Top-Down vs. Bottom-Up Process Improvement", IEEE Software, vol. 11, no. 4, pp. 12-13.

Tichy, W. F. 1998, "Should Computer Scientists Experiment More?", IEEE Computer, vol. 31, no. 5, pp. 32-40.

Tichy, W. F., Lukowicz, P., Prechelt, L., & Heinz, E. A. 1995, "Experimental Evaluation in Computer Science: A Quantitative Study", Journal of Systems and Software, vol. 28, no. 1, pp. 9-18.

Tversky, A. & Kahneman, D. 1974, "Judgment under Uncertainty: Heuristics and Biases", Science no. 185, pp. 1124-1131.

Tversky, A. & Shafir, E. 1992, "The Disjunction Effect in Choice Under Uncertainty", Psychological Science, vol. 3, no. 5, pp. 305-309.

van Solingen, R. & Berghout, E. 1999, Goal/Question/Metric Method -- A Practical Guide for Quality Improvement of Software Development, McGraw-Hill.

Visio Corp. VISIO. Technical[4.0]. 1995. Visio Corporation.

Von Neumann, J. & Morgenstern, O. 1944, Theory of Games and Economic Behavior, Princeton University Press, Princeton.

Votta, L. G., Porter, A. A., & Perry, D. E. 1995, "Experimental Software Engineering: A Report on the State of Art", in Proceedings of the 17th International Conference on Software Engineering IEEE Computer Society, pp. 277-279.

Waller, R. A. & Covello, V. T. 1984, Low-Probability High-Consequence Risk Analysis, Plenum Press, New York.

Wang, J. X. & Roush, M. L. 2000, What Every Engineer Should Know About Risk Engineering and Management, Marcel Dekker.

Warboys, B. The IPSE 2.5 Project: Process Modeling as a Basis for a Support Environment, Technical Report, University of Manchester. 1989.

Weyuker, E. J. "Predicting project risk from architecture", in Proceedings of the Sixth International Software Metrics Symposium pp. 82-90.

Widdows, R., Hensler, T. A., & Wyncott, M. H. 1991, "The Focus Group Interview: A Method for Assessing User's Evaluation of Library Service", College and Research Libraries no. July, pp. 352-359.

Willhite, A. M. 1998, An Overview of the ESC Risk Management Process, MITRE corporation, MP96B0000120, R1.

Williams, C. A., Smith, M. L., & Young, P. C. 1998, Risk Management and Insurance, 8 edn, Irwin/McGraw-Hill, Boston.

Williamson, J. A. "Experiences with an Independent Risk Assessment Team", in Proceedings of the Third SEI Conference on Software Risk Management SEI, Pittsburgh, PA.

Wohlin, C., Runeson, P., Host, M., & Ohlsson, M. C. 1999, Experimentation in Software Engineering: An Introduction, Kluwer Academic Pub.

Yin, R. K. 1994, Case Study Research: Design and Methods, 2 edn, SAGE Publications, Thousand Oaks, CA.

Yourdon, E. 1992, Decline and Fall of the American Programmer, Prentice-Hall, Upper Saddle River, NJ.

Zelkowitz, M. V. & Wallace, D. R. 1998, "Experimental Models for Validating Technology", IEEE Computer, vol. 31, no. 5, pp. 23-31.

Zuse, Horst, History of Software Measurement, http://irb.cs.tu-berlin.de/~zuse/index.html, 1997

Appendix A Risk Management Improvement Framework Definition

This appendix presents a more detailed view of the risk management improvement process.

We have used the QIP Cycle as the framework for out risk management improvement process. An overview of the process is given in Table 69.

Process step	Description	Output
Characterize	Understand the current project and risk management environment based on the available data, models, experience, and insights. Establish baselines w.r.t. frequency of risks, frequency of risk management, and effectiveness of risk management.	Baselines Current practice descriptions
Set goals	Set quantifiable goals for the project and organizational performance and improvement in risk management.	Project and study goals defined
Choose process	Choose the processes, tools and techniques appropriate for the project, combining currently available processes with new approaches as necessary.	Project plan
Execute	Perform the project and its risk management actions, delivering the project output and data about the project and its risk management activities.	Project deliverables Data
Analyze	Analyze the data and the information gathered to evaluate the current practices, determine problems, record findings, and make recommendations for future projects.	New knowledge
Package	Consolidate the experience gained in the form of new or updated models, documents and other forms of knowledge and store this knowledge in the experience base, and disseminate information in the organization.	Packaged, deployed knowledge

Table 69: Overview of outputs and exit criteria of the risk management improvement process

As with the QIP cycle in general, the risk management improvement process occurs in parallel with the enacting project, in this case involving both the project itself and the risk management process within it. In this context, we are interested in the risk management improvement process and, therefore, model the overall process from that specific perspective. This means that the process perspective taken in this chapter does not attempt to cover the software process nor the risk management process.

The risk management improvement process interfaces with the experience base in its various stages. We have modeled these interactions on a general level in Figure 43. We will discuss the specific content of the experience base in chapter B.4.

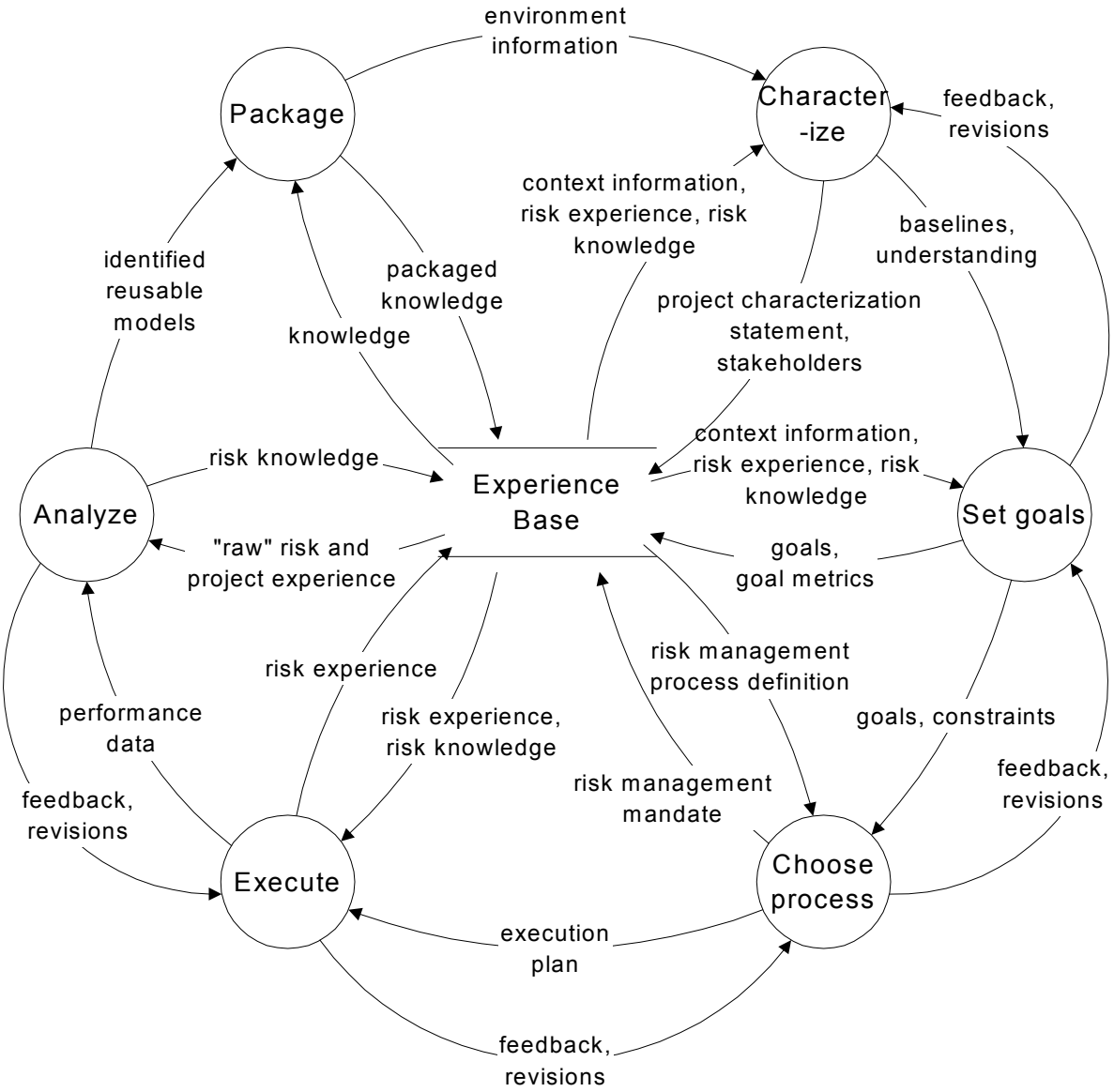


Figure 43: Risk management improvement process overview

A.1 Characterize

The *Characterize* process is one that is initiated each time a new software development project is started, which essentially is the *entry criterion* for the process ("a new project is initiated"). The Experience Base contains information about the process and the environment, but this information will need to be filtered, interpreted, and adapted for each project. The *Characterize* process takes this perspective and provides the necessary, project-specific understanding of the environment. A summary of the process is given in Table 70.

Purpose:	Understand the project characteristics and the context.
Description:	Use history data and information about the proposed project characterize it and relate to it to experiences of completed projects.
Entry criteria	A new project is initiated
Input:	Project authorization statement
Output:	Project characterization statement, i.e., understanding of the project environment and project characteristics
Methods and tools:	(none defined)
Responsibility:	Project manager
Resources:	Experience Base Other project managers or participants
Exit criteria:	Understanding of the project environment and baselines established.

Table 70: The process definition information for Characterize process

We have presented the main sources of input to the *Characterize* process in Figure 44. The project authorization (which has been modeled as an artifact in Figure 44) provides the characterization of the product and project that are being considered. The project authorization statement initiates the project planning in the organization. Initial resources are allocated to characterize the project and its environment. This is done by a role that, for the time being, acts as the project manager in this process. The characterization is guided by the information given in the project authorization statement:

- What are the general requirements, what kind of application domain is in question?
- What are the available resources and constraints in the project?
- What is the expected schedule for the project?

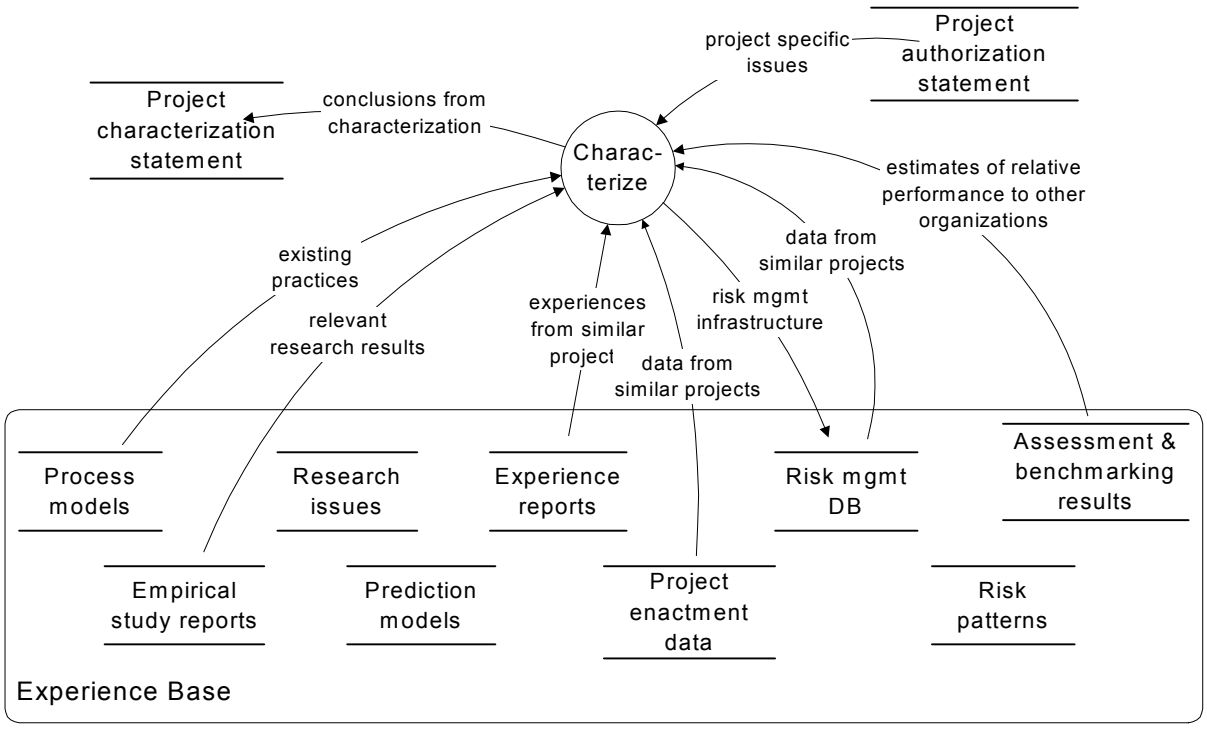


Figure 44: Characterize process and its interactions with the experience base

The project authorization statement is typically given by the management of the software development organization. It is an initial estimate of the goals and constraints for the project. The goal definition and more detailed planning (choose process) may change the initial estimates.

The *Characterize* process searches the Experience Base with this particular perspective in mind, utilizing all relevant information. Existing process models provide details on how the organization currently performs projects. The appropriateness of the existing processes to the new projects needs to be assessed. Data and experience reports from similar past projects can be used to further assess organization's capabilities and experiences for the new project. Assessment and benchmarking results may be used to evaluate how well the organization may perform with respect to other, perhaps competing, organizations. Risk management database provides information on what kinds of risks have occurred in projects with similar characteristics and what kinds of risk controlling actions were taken to control them.

The main result of the *Characterize* process is the understanding of the project environment from the perspective of the new project: what are relevant and important issues, where the strengths and weaknesses are and what has been the level of performance in previous, similar projects.

This understanding may not always be documented in a form of a document. However, as we try to formalize the results of the *Characterize* process, we recommend that an artifact called *project characterization statement* be produced. The main contents of this document are outlined below:

- Description of the main characteristics of the product to be developed; and process requirements set for the project.
- References to similar projects in the organization and main experiences from them (success factors, problems).
- Description of processes, methods, and tools used in previous, similar projects and assessment of the applicability of these experiences to this project.
- Statement of the expected performance baseline for the project, based on the analysis of the above factors.

The coverage of these issues acts as the *exit criterion* for the process.

The *Characterize* process requires cooperation between several key stakeholders. These stakeholders represent different interests and roles in the *Characterize* process. The customer is the user organization that has the need for the software product. Organization management acts as a contractual interface to the customer and has the resource allocation authority within the development organization. The project management represents the authority to plan and enact the project, within the constraints that the organization management has set. Finally, the experience factory analyst contributes to the process by being able to access and interpret the Experience Base information efficiently.

The main requirements for the project come from the customer (i.e., user organization) that, through interactions with the development organization, allows the initiation of the project and provides an initial description of the needs as well as commitment for the project, at least an initial one. The *Characterize* process results are used by the set goals and choose process steps to conclude the planning phase of a project.

A.2 Set goals

The *Set goals* process is initiated when the new project is started. Although it is dependent on the *Characterize* process, it can be started concurrently as the goal setting typically takes several iterations to agree with the goals. The *entry criteria* for the *Set goals* process is the same as for the previous process: "a new project is initiated".

The main goal of the *Set goals* process is to define quantifiable goals for the project. Product goals refer to the characteristics of the product to be developed and process goals cover the development process, time, and cost in particular. From the QIP perspective, the main characteristic of the process is the negotiation process between the customer, development organization, and the Experience Factory. Each of these stakeholders has their own expectations and objectives that will need to be balanced. The customer expects a given software product within a certain time and budget. The development organization is interested in making a profit in the development project and balancing the resource load with other projects in the organization. The Experience Factory looks at the project as an opportunity to learn more about the software development process. The goal-setting process requires these, sometimes even conflicting, goals to be balanced in a way that is acceptable to all parties. As it is the customer that bears the cost of the project, there is an implied hierarchy of these objectives. The development organization will need to "sell its case" for the customer if it has constraints that make it impossible to satisfy customer's ideal situation. Consequently, the Experience Factory will need to convince the project management of the benefits of data collection, experimentation and analysis that cause extra cost to the project.

We have presented the main sources of input to the *Set goals* process in Figure 45 and Table 71 highlights the key aspects of the process. A natural starting point for the goals are the customer requirements for the product and the process. The product requirements may not be documented in detail but they form the basis of other goals. The process requirements cover issues like the completion time of the project, cost of development and mode of cooperation and control between the customer and the development organization.

Purpose:	Define quantifiable goals for the project.
Description:	Given project characterization, understand stakeholder interests and define a set of challenging but realistic objectives and define metrics for them.
Entry criteria	Project constraints known (e.g., cost, time). Requirements known. Corporate objectives known.
Input:	Project characterization statement Business goals and policies Customer values, competition Customer process requirements Application requirements
Output:	Initial project plan
Methods and tools:	GQM
Responsibility:	Project manager
Resources:	Project steering group
Exit criteria:	Project stakeholders identified Project objectives defined Research objectives defined

Table 71: The process definition information for Set goals process

The development organization's local goals, policies, and priorities also play an important role in the goal-setting process. The organization will need to allocate resources for the project and this may mean having to make compromises between on-going projects or other new projects. Organization's policies about prioritizing projects of different type, be that by customer, technology or application area, may influence the willingness to compromise with the customer. There may also be some business reasons influencing the goal setting: possibility to enter a new market (developing a new kind of product) or to attract a new customer.

A major element in local business goals is the need to improve software development. The development project being specified represents a possibility to obtain more information about the process and even to implement and try out some changes in the process. The open *research issues* that have documented in the Experience Base provide a candidate list of experimental goals for the project. These will need to be balanced against corporate and local business objectives as well as the practical constraints of the project and resources available.

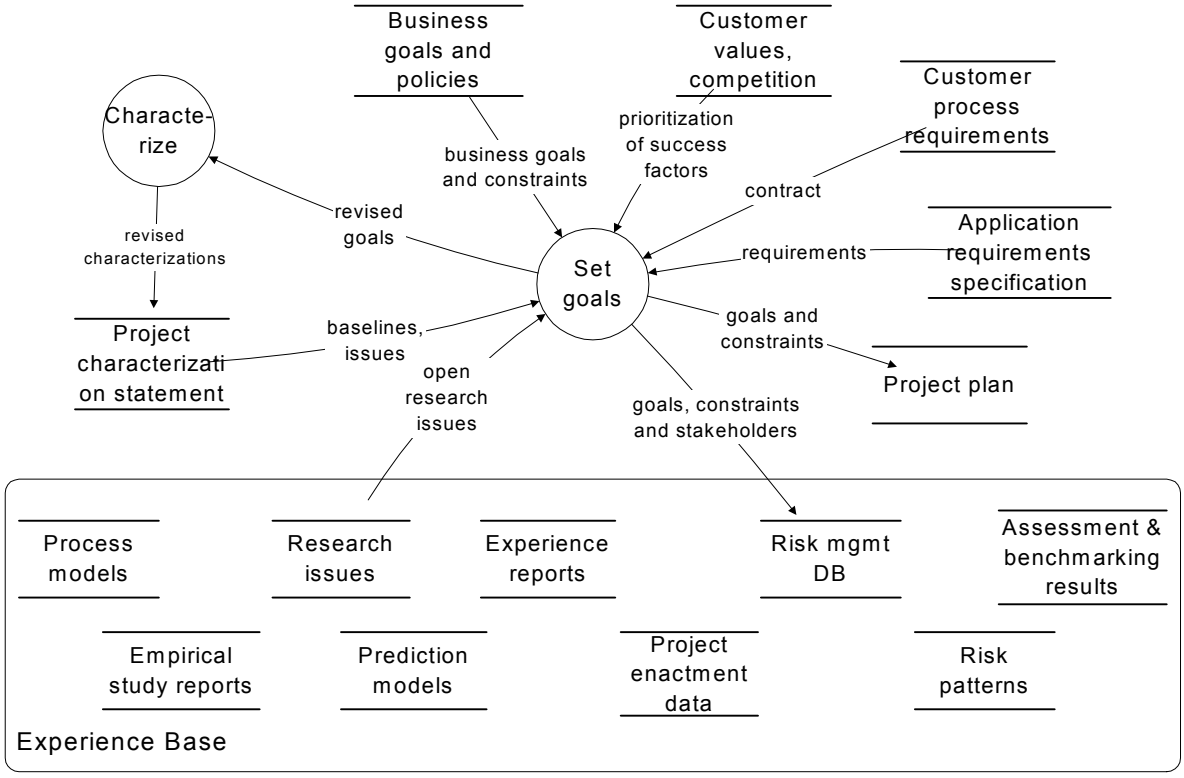


Figure 45: Set goals process and its interactions with the experience base

To help this process, we have defined some templates that can be used to define the specific study objectives in more detail.

From the perspective of improving the risk management process, we have identified some general "values" and dependencies for the attributes in the GQM goal definition template. In other words, we have proposed certain guidelines and patterns for objects of study, purpose, attributes, point of view, and context. These are presented and discussed in the following.

The object of study attribute can be characterized both by granularity and types of objects being studied in the risk management process. The granularity dimension defines the level of detail in the analysis. e.g., whether the process is analyzed as a whole or a subsection of it is

under scrutiny. The object of study can also be categorized by the types of entities that are involved in the process. We have previously identified several process information entities that are also relevant for the risk management process, as presented in Table 72 (Kontio 1995b).

Process information entities	Examples
Process activities	• process steps • hierarchical composition of processes and subprocesses
Process behavior	• process sequence • initiation condition for processes • decision points in process activation • process concurrency • re-activation of processes
Process artifacts	• content of artifacts • structure and format of artifacts • relationships between artifacts, e.g., dependencies, groupings
Process agents	• roles for agents: project manager, risk owner, etc. • agent types: software agents, managers, engineers, QA team
Process resources	Resources that are used in the process, other than personnel. • computer hardware • software tools (programming tools, case, testing tools, project management tools, wordprocessors) • methods and techniques • software development process support (project plan templates, process measurement database, project history reports) • communications media (email, telephones, video conferencing) • physical office equipment
Process infrastructure	Organizational infrastructure that affects the process. • organizational structure (organizational hierarchy, chain of command, informal communication links, power and authority structures, personnel management procedures) • data collection and reporting procedures (effort reporting procedures, process and product measurement procedures) • administration and support functions (accounting and personnel department)
Process information flow	• flow of information between processes and subprocesses

Table 72: List of possible objects of study in a process

The selection of appropriate objects of study and the level of granularity depends on the risk management process improvement goals. However, in most cases it is reasonable to start the analysis from coarse level of granularity and study the most important process objects, such as activities, artifacts, agents, and methods and tools used.

A.3 Choose process

The goal of the *Choose process* process is the definition of a detailed execution plan for the project, i.e., the project plan with the appropriate resource allocations. Table 73 and Figure 45 present an overview of this process. Ideally, the *Choose process* process tries to find the best combination of resources, processes, and tools to satisfy the goals and constraints set for the project. In many cases, however, this process requires interaction with the *Set goals* and

Characterize processes and even changes in goals and constraints. An important part of this process is the planning and balancing of how the project's business objectives and the learning objectives can be both achieved in the project.

The Experience Base plays a significant role in the planning process. It is critical in providing models and empirical data to predict process performance, process definitions to support planning, and experience reports on similar projects. The *Choose process* process results in several important commitments that may have significant consequences to the organization. Consequently, the reliability of the prediction models used in the planning process is critical in reducing risks contained in these commitments.

Purpose:	Define and select the processes, methods, and tools to be used in the project.
Description:	Given project's objectives, choose appropriate life cycle models, processes, methods and tools that can deliver projects intended results.
Entry criteria	Project constraints known (e.g., cost, time). Requirements known. Corporate objectives known.
Input:	Goal statement Requirements specification
Output:	Final project plan
Methods and tools:	Estimation tools
Responsibility:	Project manager
Resources:	Project team
Exit criteria:	Project plan written and approved Risk management mandate defined Data collection practices defined

Table 73: The process definition information for Choose process

The *Choose process* process can be initiated as soon as there is an adequate understanding of the product and process requirements, the environment, and goals and constraints for the project. Initial versions of the project plan and, if available, the application requirements specification provide the necessary information to start the planning. Again, these may need to be refined in some areas to provide better basis for planning.

The main challenge in the *Choose process* process is, however, the utilization of organization's existing knowledge as effectively as possible. The process models in the Experience Base can act as planning templates. They should be embodiments of organization's validated experiences. The Experience Base may contain alternative process models for some specific tasks and the planning process should consider trade-offs between these processes. The prediction models, correspondingly, are most reliable when existing processes are used as such. Prediction models may be used to predict costs, schedule, resource load and product quality. As we will discuss later, there can be several types of prediction models, ranging from statistically calibrated mathematical models to heuristic rules of thumb or having access to experienced project managers.

The existing experience reports on past projects with similar characteristics may provide some insights as to what has worked before and what problems can be expected. Similarly, some experiment reports may help in estimating the impact of new processes or technologies.

The exit criteria for the *Choose process* process is the completion and approval of the project plan, containing detailed definitions of the processes to be used, technologies involved and resource allocations for the project.

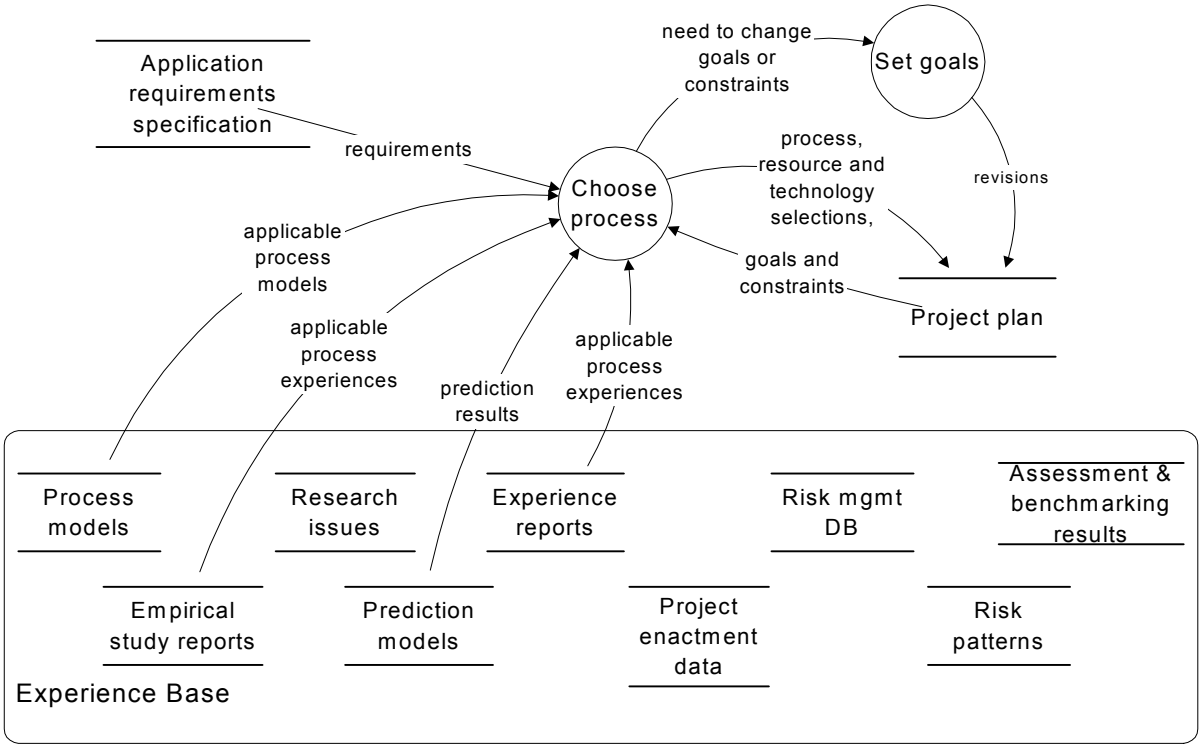


Figure 46: Choose process and its interactions with the Experience Base

While the customer requirements provide the basis for planning, the most critical part of work is done by the project manager and the Experience Factory analyst. The Experience Factory analyst contributes by providing the project manager the right background information and appropriate tools for planning. The Experience Factory analyst has better knowledge of the experiences and tools available for in the Experience Base, how they can be used, and what limitations they may have. The project manager, on the other hand, has better knowledge of the specific characteristics of the project and he or she will have the ultimate responsibility for the results of the planning. The Experience Factory analyst acts as a consultant or a support person to the project manager and they should work closely together to exploit their special knowledge.

Planning may involve the use of several overlapping models, alternative scenarios, and even simulation, depending on the tools and models available. As a general principle, the use of multiple planning approaches helps in increasing confidence in the plans. At the end of the planning process, the plan will need to be submitted for approval by the development organization management and by the customer. The project plan may be changed during execution if new situations or information makes it necessary.

A.4 Execute

The *Execute* process produces the software product. It consists of several subprocesses involved in the technical development of the software. The driving factors for the technical

development are the application requirements and the project plan. However, we will not go into details with these development activities as our focus is process improvement.

The outputs of the *Execute* have been presented graphically in Figure 47. The software product is the primary outcome from the customer's perspective. The process also produces information about the development process, i.e., enactment data and qualitative development experience. Depending on the type of data collection procedures established for the organization in general and for the project in specific, some project enactment and product development data is collected as the project goes on.

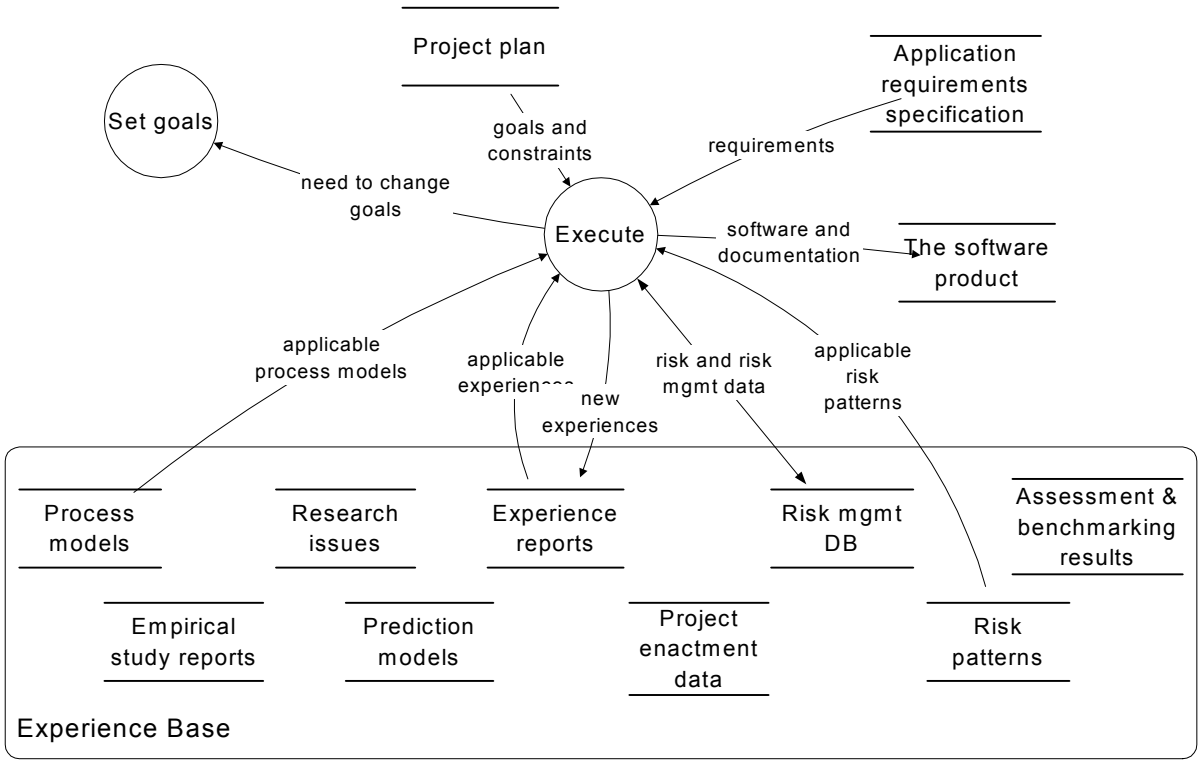


Figure 47: Execute process and its interactions with the Experience Base

As the project is being finished, the project participants should also document and summarize the main experiences from the project. It is often the case that the enactment data cannot provide adequate insights to all aspects of the project. Qualitative analysis of the experiences is therefore very important way to pass on the lessons learned. These experience reports should describe the main problems and their causes, how well the development processes, methods and tools fitted to the project, what deviations from the plan happened and why, and what were the special characteristics of the project. It is recommended that the Experience Factory analyst is involved in the documentation of these experiences in order to make sure that an adequate level of consistency is maintained across these reports and that each report covers the project in adequate depth.

The *Execute* process may also reach a situation where it becomes necessary to change the goals or constraints of the project. This may be due to unexpected change in the situation, changes in the requirements, wrong assumptions made in the planning phase, risks that realized, or any other reason. As soon as the project management finds out that project goals cannot be met, a conflict resolution needs to be initiated. This results in re-initiation of the

Set goals process, which is represented by a dataflow arrow to that process in Figure 47. This may result in changes in goals (e.g., more time for the project, relaxed requirements) or changes in the constraints (e.g., more money or resources for the project).

The entry criteria for the *Execute* process is the authorization of the project manager, existence of a project plan and allocation of resources to the project manager.

The *Execute* process contains actually several important subprocesses. From our perspective, it is important to highlight the analysis and control activities in this context in more detail. This is especially important to point out the differences between analysis from the project perspective vs. analysis from the process improvement perspective. The *Execute* process has been detailed further in Figure 48.

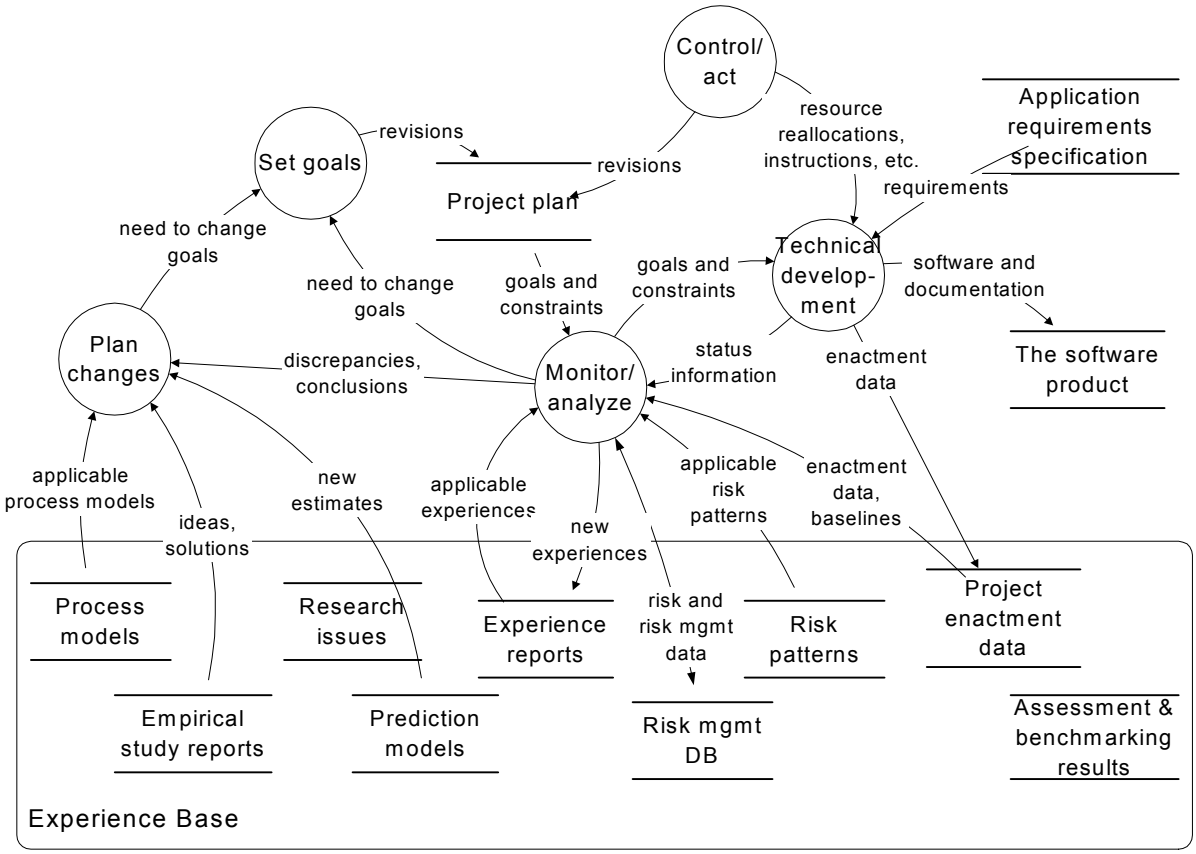


Figure 48: Subprocesses of Execute process and the interactions with the Experience Base

As we stated earlier, the technical development of the software product produces information on the status and progress of the process and product. Some of this information has been formalized and collection procedures established so that it can be collected and entered to the Experience Base, i.e., a database defined for this purpose. This information can be used later in the Analyze process to plan changes to the whole process. However, the project manager can also use this information for project specific management and control. The arrows from *Technical development* process to the *Enactment data* database and from there to the *Monitor/analyze* process represent this dataflow in Figure 47.

The *Enactment data* database, however, cannot capture all relevant aspects of the project for effective project management purposes. The need to obtain other, more informal

information about the project is represented by a dataflow arrow between the *Technical development* process and the *Monitor/analyze* process. This information can be obtained from discussions with the project personnel, meeting and review reports, or external sources and can include information about problems encountered, new issues that are emerging, risks, resources availability, etc. For a perceptive and agile project manager, this informal project information often provides valuable insights to the projects status sooner than the enactment data.

The *Monitor/analyze* process uses the information about the project to check whether problems have occurred, whether there are deviations form plans, or whether the current situation seems to indicate problems in the future. This process may use *prediction models* or past *experience reports* to evaluate the impact of the situation to project objectives. The *Monitor/analyze* process also often recognizes minor issues that are a natural part of the management: whether personnel needs to be allocated to new tasks, design issues need to be resolved or development issues prioritized. If any of these issues require action, this is passed on to the *Plan changes* process.

At the end of the project, the *Monitor/analyze* process consolidates the experiences and lessons learned during the project and documents them in the Experience Base. It is recommended that the Experience Factory analysts participate in this process in order to make sure that all relevant experiences are recorded.

The *Plan changes* process considers alternative ways of action to resolve any problems that have been identified in the *Monitor/analyze* process. This may involve the use *prediction models,* if the effect of proposed actions need to be estimated, consideration of different *process models*, or review of *experiment reports* in case problems may be solved by new techniques. Minor issues may not involve lengthy planning, they may be just identification of the easiest working solution and passing this on to the *Control/act* process. The more critical the problem is and the more variety alternatives contain, the more carefully must the planning be carried out.

During the *Plan changes* process it is necessary to verify that the project goals and constraints are all still valid in the new situation. If this is not the case, or if there is a risk of this, the revision of goals and constraints should be considered, as indicated by a dataflow arrow to the *Set goals* process in Figure 47.

A.5 Analyze

The goal of the *Analyze* process is to evaluate the process performance from the perspective of future projects. The purpose is to identify strengths and weaknesses and identify processes, practices, resources, or technologies that should be improved or changed to enhance process performance. In the Experience Factory context, changes to the process should only be made after there is enough evidence of the impact of changes. The assessment of when there is "enough evidence" about the impacts is, of course, sensitive to several factors, such as the level of risk the organization is willing to bear, how serious the current problems are, or competitive or environmental factors. To point out a couple of examples, change in industrial de facto standards (such as the popularity of MS-Windows) may force some companies to adopt that development platform, regardless of how much experience they have in it or an object oriented development method may require years of validation before it can be introduced in an organization involved in safety critical systems.

The *Analyze* process is central from the point of view of process improvement. It is the process that identifies problems, evaluates solutions and initiates change. Due to its central role, it utilizes the Experience Base information extensively, as can be seen in the dataflow arrows in Figure 49.

The *Analyze* process not only considers process enactment data and process experience. It is important to consider corporate and local business goals and strategies, competitive situation, customer values, and technology advances. These issues may strongly influence the timing, appropriateness, and prioritization of possible process changes. We will analyze these inputs in the next few paragraphs.

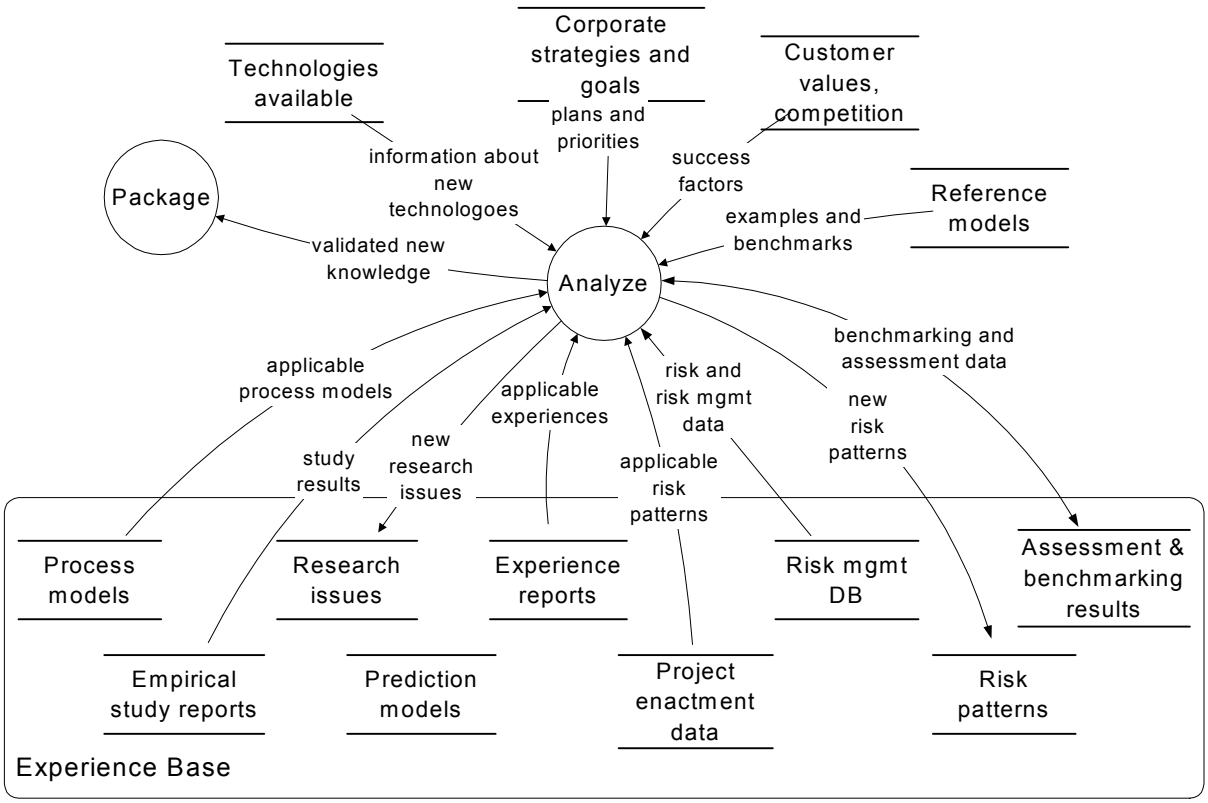


Figure 49: Analyze process and its interactions with the Experience Base

The *Analyze* process can be initiated at any point in the process cycle. The triggering events are changes in any of the items that have in-coming arrows in Figure 49. Naturally, a completion of a project typically is an event that has produced some data to be analyzed. However, each project may not provide new data that would justify a lengthy analysis and, on the other hand, many other events can make the *Analyze* process necessary to be initiated, such as changes in technology, corporate goals, or customer values. There may also be a need for assessing the process in certain time intervals.

Each time the *Analyze* process is initiated, its goal is first to understand the environment and the process itself from the improvement perspective and then identify whether changes are necessary and what they should be. Thus, the exit criteria for the process are the recognition, definitions, and prioritization of changes, i.e., improvements to the process.

For maintaining consistency in our presentation, we presented the *Analyze* process as a single process in Figure 49. However, the *Analyze* process can be viewed as consisting of three distinct tasks: (i) analysis of the software development process itself, (ii) analysis of the external factors that need to be taken into account, and (iii) prioritization of the identified process changes. The first one (i) analyzes the process performance and characteristics from several perspectives, the second (ii) helps in systematically accounting for all relevant external factors and the third one (iii) finally decides what needs to be done. These three subprocesses are discussed in detail in the following and the dataflows between them are presented in Figure 50.

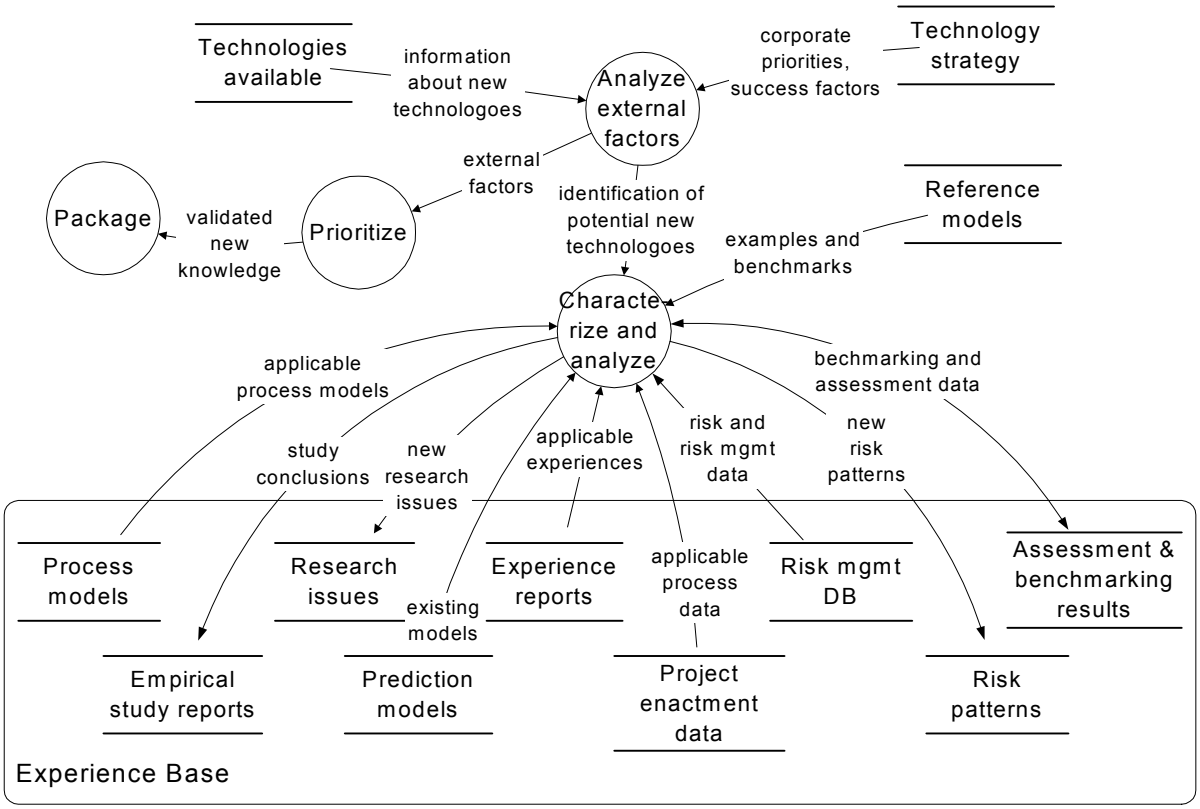


Figure 50: Subprocesses of the Analyze process and their interactions with the Experience Base

The characterization and analysis of the software development process ("Characterize and analyze" in Figure 50) is based on the information available in the Experience Base. Enactment data provides a detailed view on the projects. This data can be compared to other similar projects and baselines, used to compare how well the prediction models worked for the project, and review other aspects of projects, such as problems encountered or how well standards and processes were followed. Note that the enactment data not only includes measurements collected during development, it should also include other project documentation (project plans, meeting minutes) and even copies of the actual artifacts produced during the project (e.g., requirement and design specifications, source code, etc.).

This "raw" enactment data is complemented by the project experience reports that are written at the end of the project. The experience reports contain project personnel's own analysis and conclusions of the experiences in the project. Access to the enactment data and experience reports allows analysis to take place at two levels, locally within the project where

the project experiences are fresh and more details are available for making conclusions, and globally across different projects, where new patterns may be recognized.

The project experiences and enactment data can also be used to assess the effectiveness of the process models used in the projects. This analysis naturally includes the assessment of projects' process conformance, i.e., how well the projects have followed the defined process models.

The analysis of the process also includes the evaluation of the usefulness of the prediction models used for projects. As the actual data is available from projects, the prediction models can be compared against this data, discrepancies identified, their reasons analyzed and changes to the models can be proposed.

The process performance can also be periodically compared to external reference models. This may give important insights as to how other organizations are developing their software and what their experiences are with different processes and technologies. A single organization may simply not have the time to experiment and try out all possible changes to its processes. External reference models may therefore represent a significant "short cut" into identifying potential process changes. However, it is important to point out that there are significant risks involved in transferring experiences between organizations. If the situational factors are not properly understood and the environmental characteristics of the organizations are different, the changes may have quite different or unexpected results in different organizations. Therefore, we recommend that the external reference models are primarily used for identifying potential changes and they are experimented first by the organization.

The two main types of utilizing external reference models are benchmarking and process assessments. Benchmarking is based on the idea of selecting an organization with similar software development characteristics and engaging in, often mutual, analysis of processes and exchange of experiences. It is often recommended that one should choose a leading organization to benchmark against, in order to learn advanced practices (Bendell et al. 1997; Spendolini 1992). The success of benchmarking is sensitive to the adequate access to process information, adequate understanding of the context, i.e., similarities and differences between the organizations, and ability to interpret the benchmarking results. Benchmarking data and conclusions, as far as they are available and not subject to confidentiality restrictions, should be documented in the Experience Base for further use. The use and providing of these reference model assessments are represented by a two-way arrow between the corresponding Experience Base component and the Characterize and analyze process in Figure 50.

The process assessments can be used for two purposes. First, they can be used to assess the "maturity" of the development organization. This is based on the assumption that the assessment model realistically characterizes the factors that determine good or mature software development. Most assessment models make this assumption and even assume that the factors are the same for all organizations. Assuming this, it is possible to determine how good each software production unit is and give them scores and rankings. As we have discussed earlier in this document, there are strong reasons to question the validity of the assumption made.

Another, perhaps more reasonable, way of using the assessment models is to use them as checklists for modeling and analyzing the software development process. For instance, the key process areas of the CMM can be used to identify areas of the process that may need to be addressed by the organization. Furthermore, the assessment models even describe what the organization should do in each of these areas and this, again, can be taken as a proposal for a

process change. However, just like we warned about the dangers of copying benchmarked practices, this may be risky and even counter productive way to change processes, if the situational characteristics and organization's goals are not taken into account.

One of the purposes of the Experience Factory is to support the conducting of experiments as projects are carried out. Many projects may include experiments as part of their product development. In such a case, the analysis of the experiment results is necessary to determine the applicability of the results to other projects. As the definition and conducting of experiences is primarily in the interests of the Experience Factory, we have included the analysis of the experiment results in the *Analyze* process. Essentially, this consists of writing the report where experiment definition, data, and conclusions are documented. Note that experiments can also be carried out as separate projects.

The analysis of the process may result in clearly identified problems and solutions, but it may also result in identification of new open questions, i.e., research issues that need to answered before conclusions about the process can be made. These research issues are also documented in the Experience Base and they act as a pool of experiment candidates in the *Set goals* phases of new projects.

The analysis of the process results in a characterization and understanding of the process – how the process performs, what are its weaknesses and strengths. Before any conclusions about possible changes to the process can be made, it is necessary to analyze the impact of external factors to the process, marked *Analyze external factors* in Figure 50. They determine what changes to the process serve the goals of the organization the best.

The analysis of the external factors considers the three groups of factors, as indicated in Figure 50. Technological changes and developments may open up new opportunities for improvement. Given the reasonably rapid rate of technological development in software engineering, this clearly is a critical activity for any organization to undertake. New technologies may need to be evaluated and experimented with. Monitoring of new technologies is particularly relevant for two reasons. First, because they are likely to be available to all organizations approximately at the same time, the competitors may adopt them and improve their performance. Second, there may be experimental data available about the impacts and effectiveness of these technologies, helping in assessing their applicability to the organization. When there is potential for experimenting or implementing a new technology, this needs to be explicitly analyzed and this may result in an experiment or, in some cases, direct implementation of the technology. This technological impact is modeled as a dataflow arrow from "analyze external factors" to "characterize and analyze" in Figure 50: first new potential technologies are identified and then their impact to the process is analyzed.

When the technological analysis is being carried out, this also includes considering the compatibility of the existing technological portfolio to the software technology strategy. The software technology strategy is a part of the corporate strategy definition. In Figure 50 the impact of competitive situation, targeted customer groups and the corporate strategy planning process are taken into account by the artifact marked as software technology strategy.

The analysis of customer needs and values, as well as the competition, provides a way of identifying the product and process characteristics that can influence success for the organization. An organization that can produce software that corresponds to customers' values better than competition has an advantage. Knowledge of the customer needs and values helps delivering products that are easy to sell. It is not difficult to find examples of

successes and failures in this respect. Microsoft was early in recognizing that early PC users were attracted by fancy features and inter-product integration more than by product reliability. They were soon able to increase their market share in many product segments.

In addition to customer needs and values, it is also important to understand where the competition is. If you are targeting the same customer group, it makes sense to implement such changes to the process which result in largest increases of customer perceived value over your competitors, either over time or per dollar invested.

The most common technique for analyzing customer needs, values and expectations is the Quality Function Deployment, QFD, which has been adapted to software development (Erikkson & McFadden 1993; Brown 1995), as well as competitors (Stalk et al. 1992; Pralahad & Hamel 1990; Porter 1985). Although the selection and use of these techniques is beyond the scope of the Experience Factory and this work, knowledge of them may make the software process management easier.

The positioning towards competition, especially changes in this positioning, are actually strategy and policy issues. They essentially determine how the organization wants to compete now and in the future. The corporate goals and strategies strongly influence which process characteristics are essential in the future. The organization may be entering new markets, changing its customer base or trying to gain competitive advantage by streamlining its internal operations. The software development process will need to be developed in accordance with these corporate objectives. For example, if the organization is entering a new market where short delivery times are critical to success, this is where the process improvements should be aimed at. Correspondingly, the need for highly reliable software or low cost development may result in totally different process changes.

The final step in the *Analyze* process is the prioritization of the proposed process changes. As we have described earlier, this is based on evaluating existing process characteristics, evaluating the impact of potential process changes, and using the results of external factor analysis to determine the appropriate priorities for the proposed changes. Depending on the type and magnitude of the change, this may result in a straight-forward revision of the Experience Base contents or the initiation of a process improvement project. Both of these cases are discussed separately in the following chapters.

A.6 Package

The *Package* process documents the synthesized knowledge, stores it in the Experience Base and disseminates it to the organization. The *Package* process can be initiated when there is some synthesized, validated knowledge that needs to be documented. Packaging consists of two main activities: documenting the knowledge and disseminating it as presented in Figure 51. Documenting means expressing the newly formulated knowledge in a form that makes it easy to understand and use in practice. Dissemination means making the packaged knowledge available, promoting its usage, providing training etc.

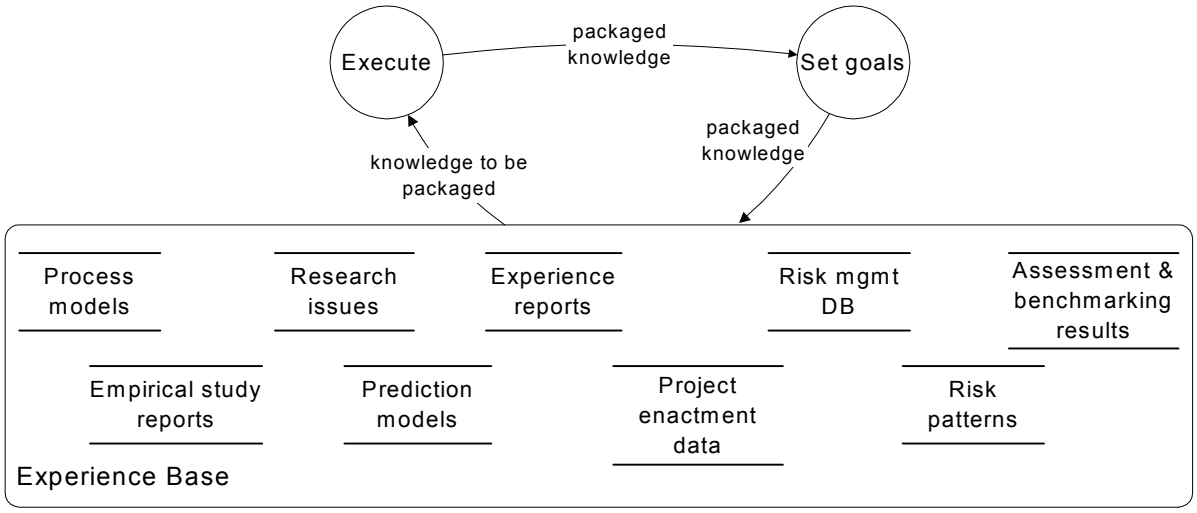


Figure 51: The Package process and its interactions with the Experience Base

The *Package* process usually packages knowledge that has been formulated in the analyze process. In addition, it can use any Experience Base contents as input. Its output can consist of various different types, such as:

- updated Experience Base elements,
- training material,
- presentations, and
- newsletters, email announcements.

Note that in Figure 51 we have attached the dataflow arrows to the edge of the whole Experience Base symbol to indicate that any Experience Base item can be part of these dataflows. The *Package* process exit criteria are complete for each knowledge item when they have been documented and appropriately disseminated.

Finally, the improvement process and risk management process are enacted simultaneously and Figure 52 presents how they interact with each other.

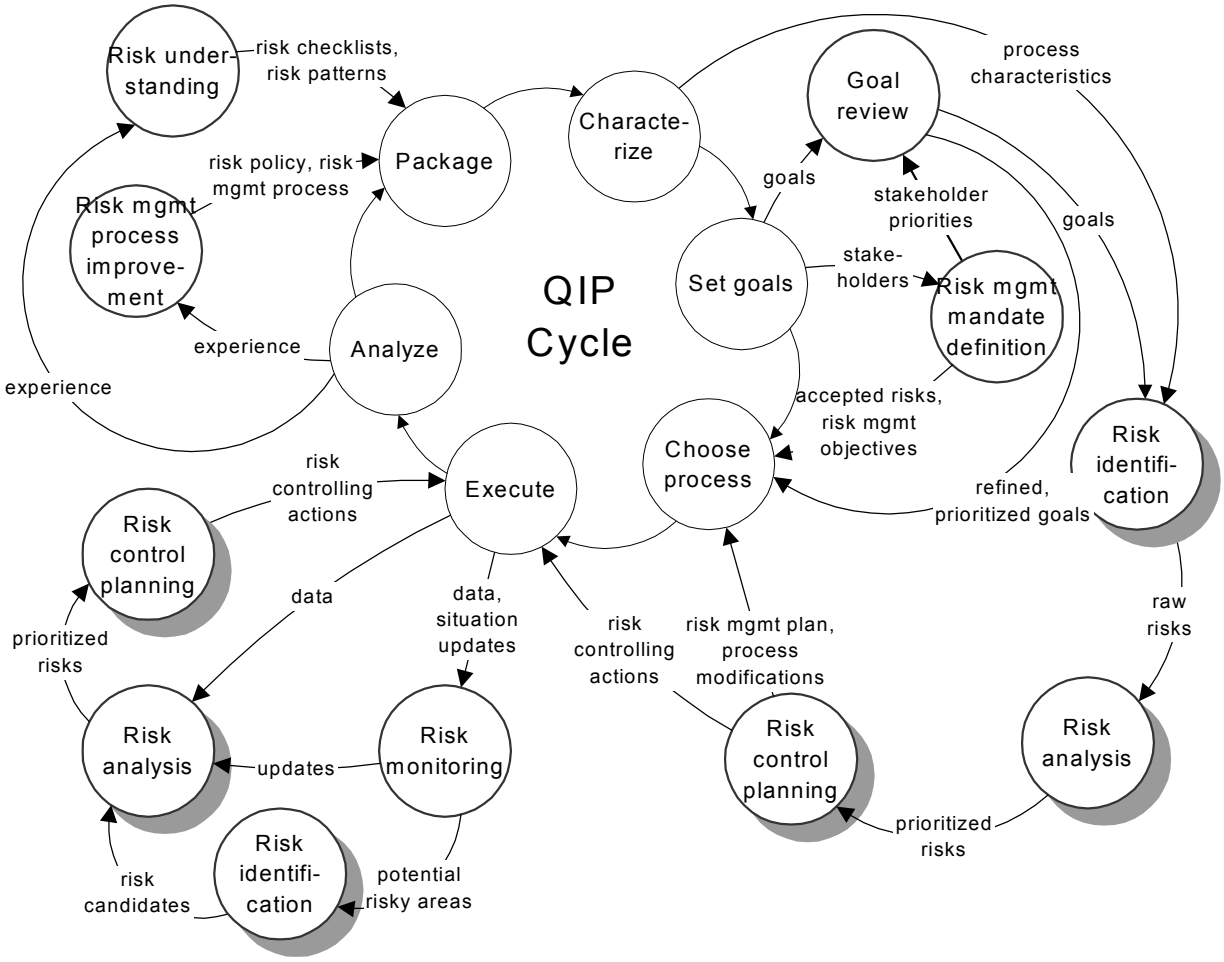


Figure 52: The Riskit process and its interfaces with a projects QIP cycle

Appendix B eRiskit Application Description

This appendix presents an overview of the eRiskit application.

B.1 Introduction

eRiskit is a www browser based comprehensive risk management tool that implements the Riskit method developed by J. Kontio. It provides support for all phases of the method, from risk management mandate definition to continuous cycles of risk identification, analysis, control, and monitoring. The application has been designed to support risk management by a team of geographically distributed people. Since the software can be accessed with a standard www browser, such as Netscape or Microsoft Explorer, no installation of software to workstations is required. The user interface metaphor is similar to that of web pages.

The main functionality in the software has been implemented and tested. Of the different user groups presented in Chapter B.3.2, only administrator has been implemented. We have also created another type of user, who has full control to risk management information, but cannot create new users, projects, or assign user rights. There are possibly a very large number of different reports of the risk management information stored in eRiskit that could be useful. We have defined and implemented five of them.

The development of eRiskit was started by a group of students at the Helsinki University of Technology on a course Tik-76.115 Software projects. The team members were Joni Hahkala, Veli-Pekka Kröger, Esa Rosendahl, Matias Turkkila, Ari Tervo and Sami Visti. The development has been continued by Esa Rosendahl and Hua Huang. The application is owned by R & D-Ware Oy.

B.2 Architecture and Technical Characteristics

eRiskit is a www browser based application that uses HTML pages, applets, servlets and ODBC-JDBC connection to implement a functioning software product. This chapter describes the architecture of the application.

B.2.1 Hardware Architecture and Technical Platforms

The application consists of the client software, server software, a database management system, and the middleware between them. The user connects to the server with a client computer that has a JDK1.1 compatible www browser, for example Internet Explorer 5. A physical picture of the system is presented in Figure 53.

The actual software is located on a Windows NT or Windows 2000 server. From the server, there is an ODBC-connection to database, which holds the risk information. On the server computer, there is also a www server capable of handling servlets. We have used Microsoft Internet Information Server 4 as a web server completed with Live Software JRun program for servlet support.

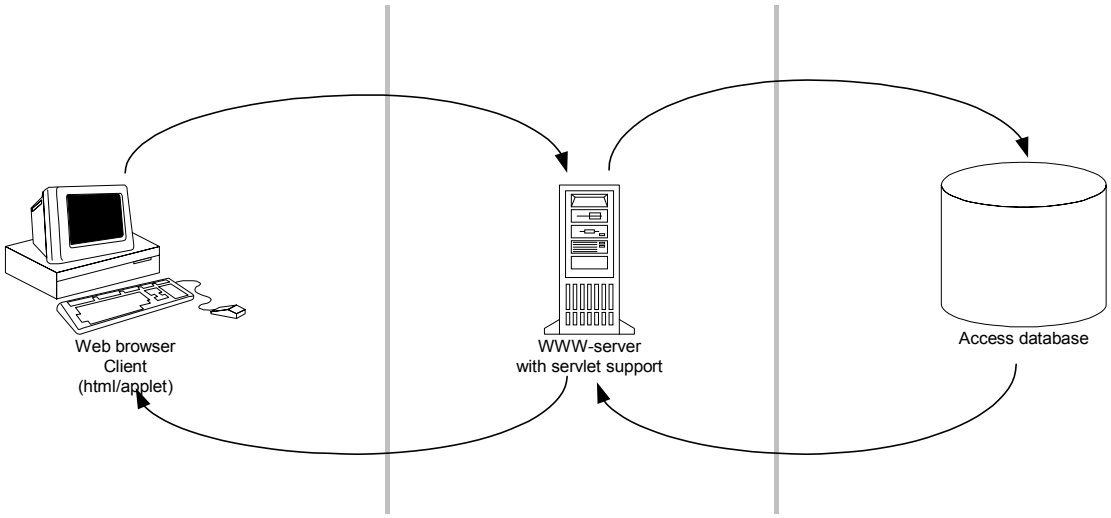


Figure 53: Application architecture

The database can either be located in the same computer as the server software or on a different computer. We have built the database using Microsoft Access and the software required for the hosting server is available free of charge.

B.2.2 Software Architecture

The application consists of the client software (applet), server software (servlet), the database management system (DBMS), and the middleware between them. The user connects to the server by opening a specific Internet address with his browser, which downloads a Java applet. The applet contacts the servlet on the server, which in turn connects to the database and fetches the data. The servlet then sends the data to the applet, which presents the information on the user's browser. This basic concept of the system is illustrated in Figure 54. The application has been separated from the database product used, so that the underlying database can be changed with only minimum modifications to the software code.

The eRiskit application is divided into separate modules based on the different phases of the Riskit process. There are also separate modules for system administration, report generation and different types of risk management policies that a company may have. All together, there are 19 different modules. Each module contains at minimum one applet class, one servlet class, and one data class for holding the risk management data needed by the applet. Each module represents a separate Java package. In addition, there are two more packages: general and dataClasses. General package contains classes used by several different packages, such as user interface components, etc. The dataClasses package contains a class for each table in the database.

There are four important classes in the solution: AppletNetInterface, ServletNetInterface, and DBInterface in the general package and BaseDataClass in the dataClasses package. AppletNetInterface is the mother class of all the applets in the application; it contains the functionality needed to communicate with servlets. Similarly, ServletNetInterface is the mother class of all servlets; it contains the functionality for communication with applets and for creating the HTML pages with applet tags, as discussed later in this chapter. DBInterface

is a class that handles the communication with the database. This is the only class that needs modifying, should the underlying database solution be changed. BaseDataClass is the mother class of all the classes in dataClasses package.

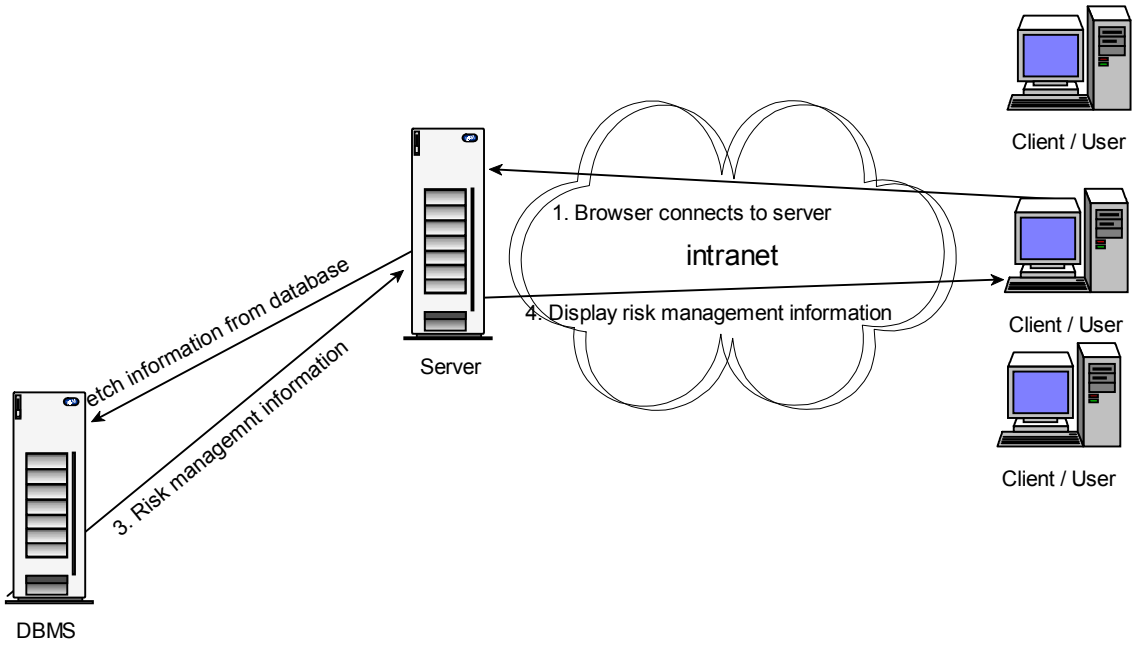


Figure 54: The software architecture of eRiskit application

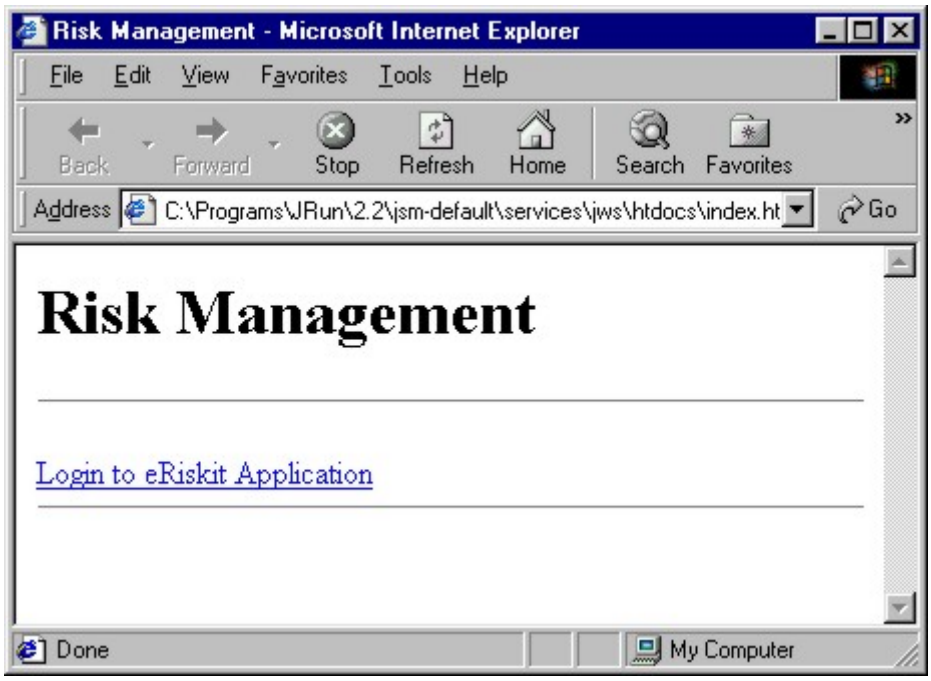


Figure 55: eRiskit start page

As the user starts to use eRiskit, he connects to the www server by opening a connection to a specific Internet HTTP-address with his browser. On this page, there is a link to eRiskit application. This page is an ideal place to present information about the risk management practices of the company that is using the application, as well as tutorials, training material, etc. Figure 55 presents this page without the additional information. As the user clicks on "Login to eRiskit Application" a new window is opened with no menus, buttons, or the address bar.

The HTML page that is opened to the new window contains an applet tag, which causes the browser to load the login applet. The user writes his username and password into corresponding fields on the applet. The data is posted using HTTP-POST to the servlet, which checks authentication data from database. By comparing the user input and database data the servlet decides if authentication is successful. If the username and password are valid, the servlet creates an HTML page and sends it to the browser. If authentication fails, a new HTML page is created that announces about invalid username or password and offers possibility to go back to the login screen.

B.2.3 Application Characterization

The application has almost 70 000 lines of code, most of which is automatically generated. All together, there are about 190 different classes and Java interfaces, about 40 of which are located in the general package and another 40 in the dataClasses package. The rest are located in different modules.

There is online help available in each module. The help hyperlink opens a new browser window with help for the current module. There is Javadoc documentation available for each interface, class, and method.

B.3 Application Functionality

B.3.1 Main Risk Management Functionality

The eRiskit application provides support for all phases of the Riskit method, from risk management mandate definition to continuous cycles of risk identification, analysis, control and monitoring. The main functions can be accessed from the main screen that opens after successful user login. The main screen is presented in Figure 56.

The software provides workflow guidance to users, i.e., all Riskit method modules have pointers to next steps in the Riskit process. However, experienced users can maneuver in the application without using these process cues as well. As shown in Figure 57 each eRiskit screen contains three buttons (Save, Main, and Next) that allow users to navigate either independently or according to the Riskit process (indicated by the "Next" button).

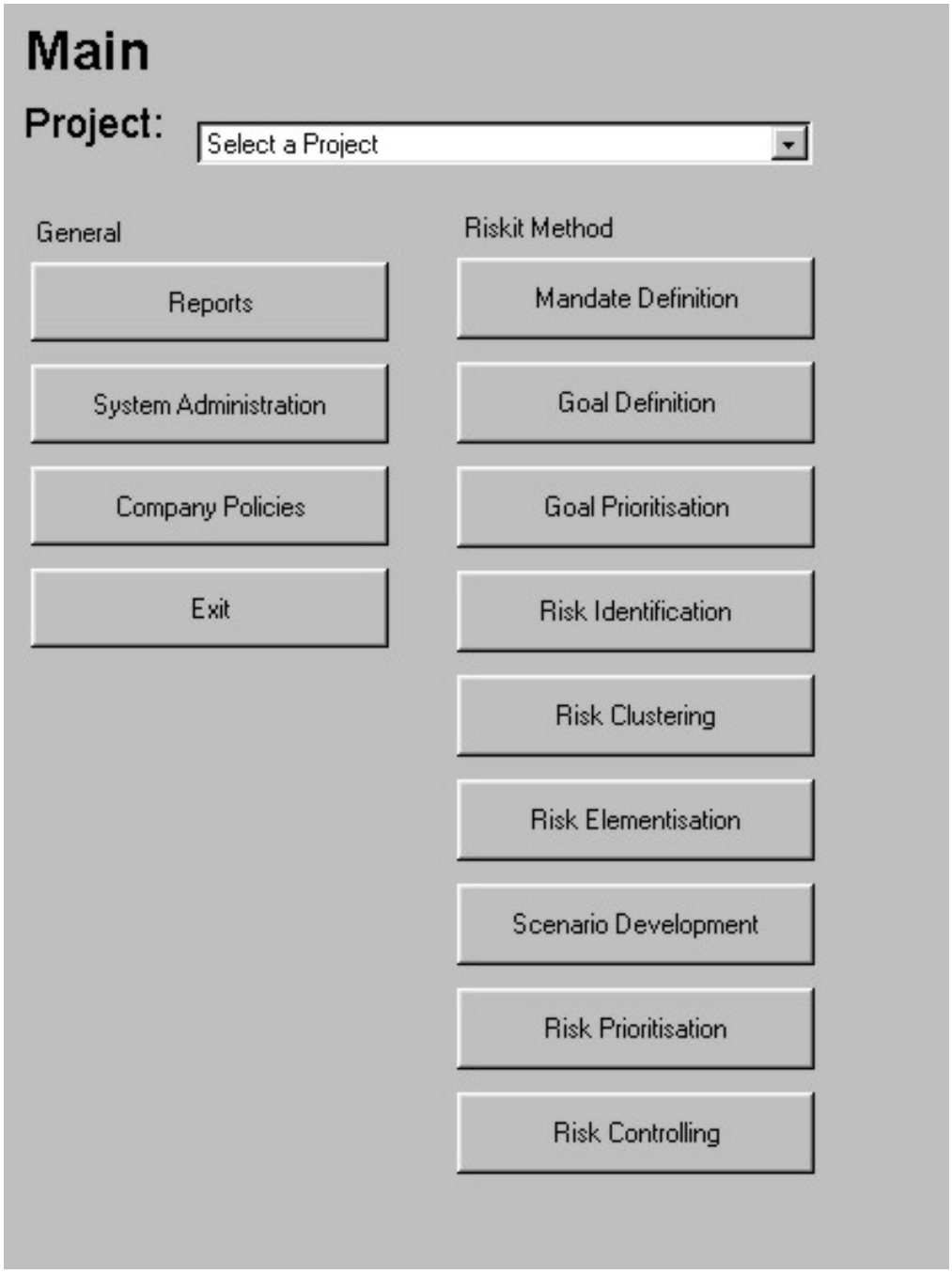


Figure 56: eRiskit main menu

The software supports risk information entry both through forms and through graphical editing of Riskit Analysis Graphs. Figure 27 shows an example of the graphical editor. Information entered in textual form can be viewed in graphical form and vise versa. The system also allows saving of incomplete graphs so that they can be completed over several sessions or even by several individuals.

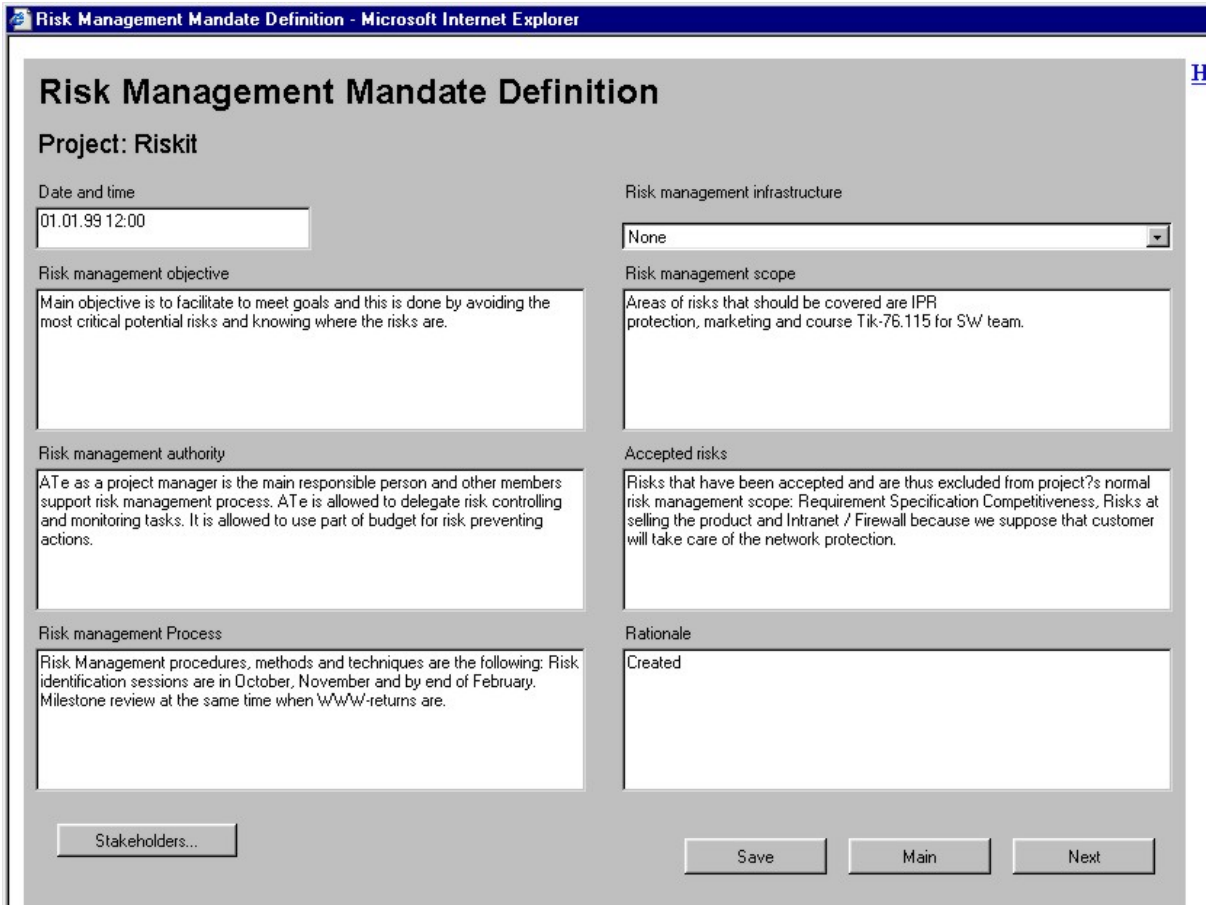


Figure 57: Example screen from eRiskit: risk management mandate definition

The eRiskit application also has functionality to support clustering of raw risks, and converting them to risk elements (called "elementization" in the tool). Raw risks can be entered at a meeting or imported to the application. They can be selected and converted to different risk factors or risk events, an action that instantiates the corresponding risk element for further editing. This functionality avoids manual entry of information that is already in the system.

The risk scenario prioritization is done in separate screens that support ordinal scale rankings of losses and probabilities. Loss rankings can be performed for each stakeholder so that stakeholder priorities for risks can be obtained and analyzed. An example of this is shown in Figure 59.

Finally, the application has extensive online help that provides context sensitive guidance for most screens.

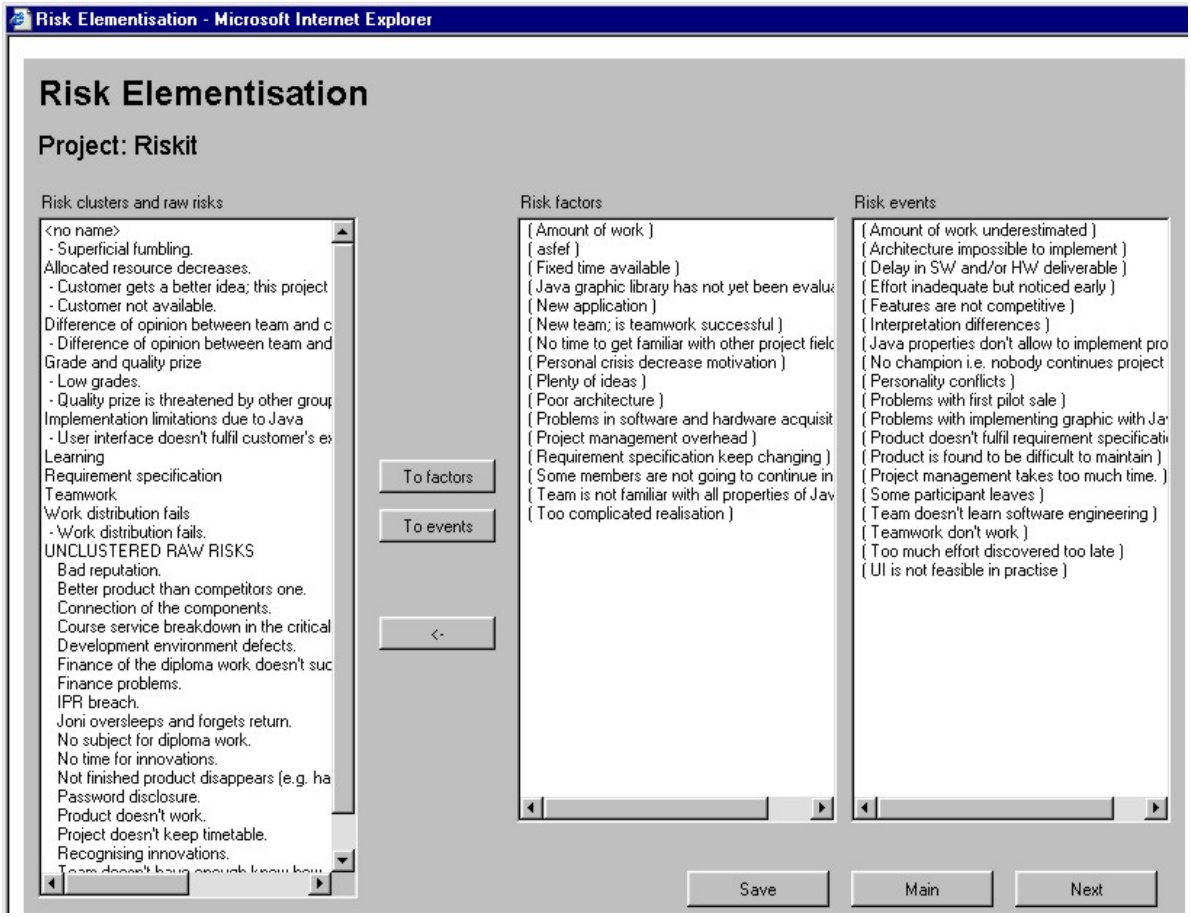


Figure 58: Example screen from eRiskit: risk clustering and "elementization"

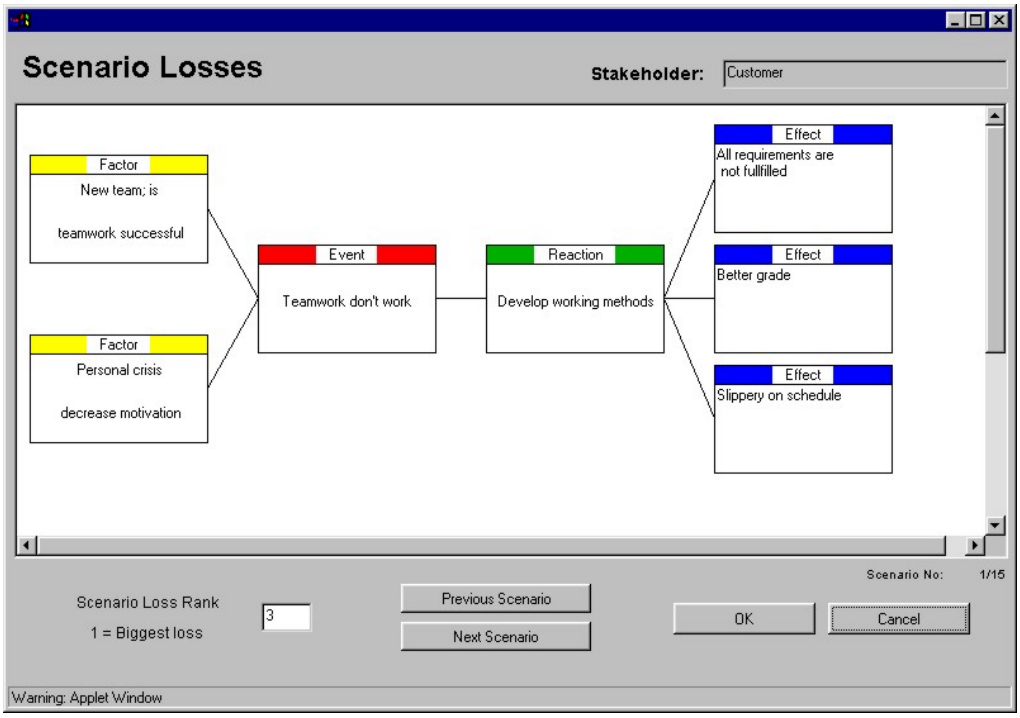


Figure 59: Example screen from eRiskit: ranking of loss

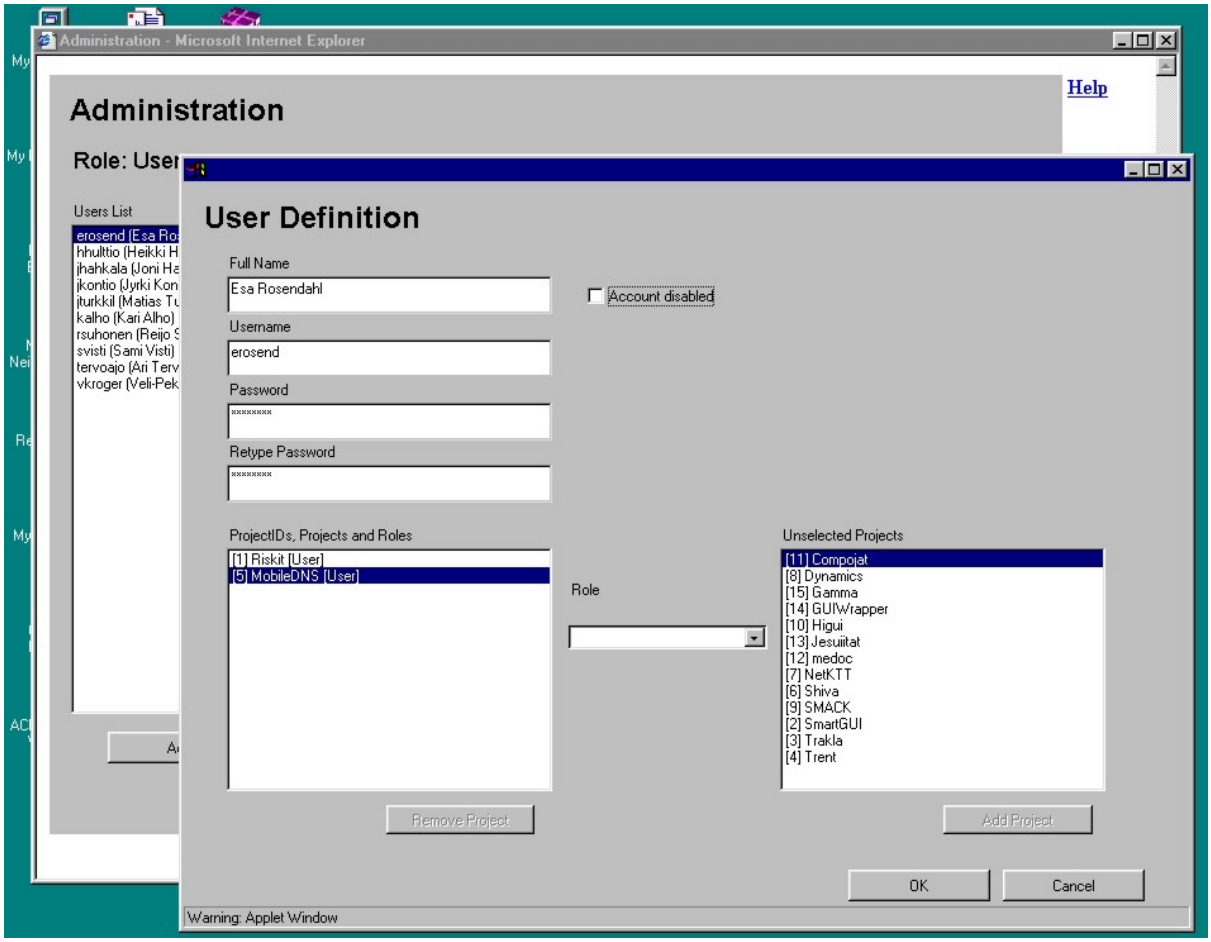


Figure 60: System administration

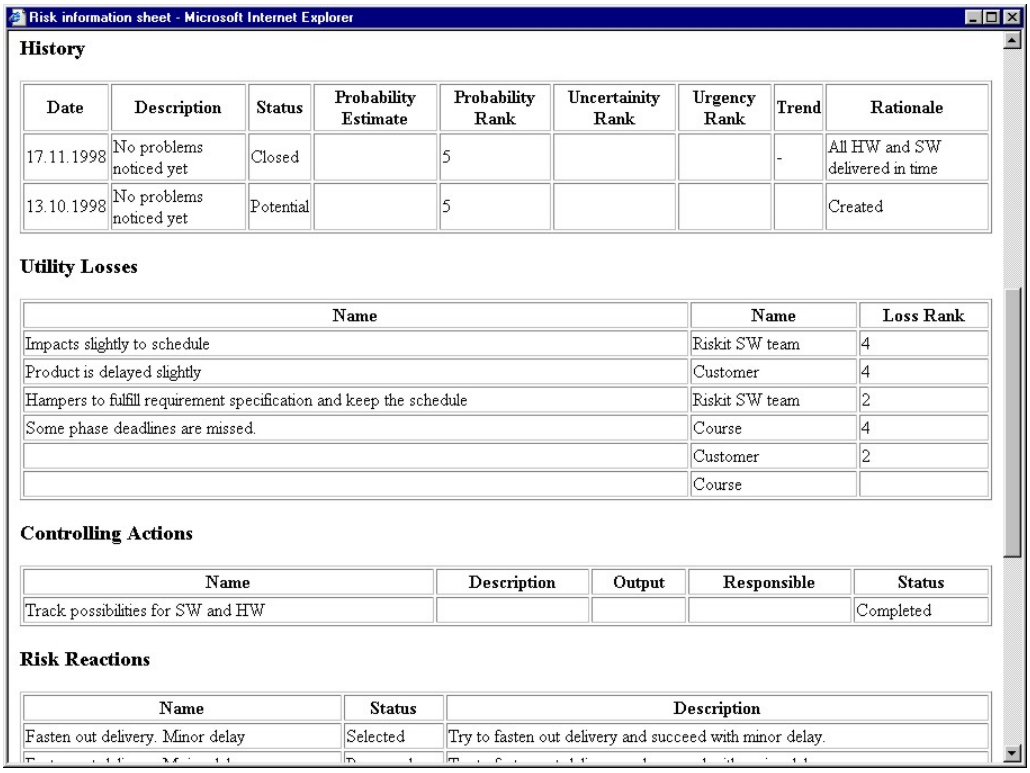
B.3.2 Support and administration Functionality

The creation of new projects and new users, and giving the users specific rights to specific projects is done in system administration module. The definition of new users and assigning rights is presented in Figure 60.

The system supports several user groups and their different access levels in the system. Also, the underlying database can contain information about several projects, yet each user only is able to see the projects that they have been granted access to. The user groups have been listed in Table 74.

User group	Functions
Project Manager and other team members	Risk identification, analysis and review reporting
Project Steering Committee	Risk analysis and review
Risk Management Process Owner	Risk analysis, review and experience capture
Risk Analyst	Risk analysis
Quality Organizations	Audit control, risk analysis and experience capture
Any other project member	Risk identification
System administrator	Full control, including the creation of new projects and users

Table 74: The main functionality by user groups



Risk information sheet - Microsoft Internet Explorer

History

Date	Description	Status	Probability Estimate	Probability Rank	Uncertainity Rank	Urgency Rank	Trend	Rationale
17.11.1998	No problems noticed yet	Closed		5			-	All HW and SW delivered in time
13.10.1998	No problems noticed yet	Potential		5				Created

Utility Losses

Name	Name	Loss Rank
Impacts slightly to schedule	Riskit SW team	4
Product is delayed slightly	Customer	4
Hampers to fulfill requirement specification and keep the schedule	Riskit SW team	2
Some phase deadlines are missed.	Course	4
	Customer	2
	Course	

Controlling Actions

Name	Description	Output	Responsible	Status
Track possibilities for SW and HW				Completed

Risk Reactions

Name	Status	Description
Fasten out delivery. Minor delay	Selected	Try to fasten out delivery and succeed with minor delay.

Figure 61: Example screen from eRiskit: sample report

B.3.3 Other Functionality

The eRiskit application allows the tracking of risk status information. Status of risks and their controlling actions are kept up-to-date in risk element and risk controlling action forms. Summary reports of the information can be printed or viewed as required in reports module of the application. An example is shown in Figure 61; the figure presents a part of a sample report.

The software has been built to support easy language adaptation, i.e., terms shown to user are kept in a single location, allowing cost-efficient translation to local languages.

B.4 Database Content

We have identified four main types of risk management empirical data that can be collected and utilized: context information, risk management process information, risk element information, and risk monitoring information. We will introduce each of these in the following.

Context information refers to such information that determines the circumstances and setting where the project and its risk management are carried out. Context information is relevant for all software engineering measurement data, but it is particularly important for risk management. The probability of a risk event is often influenced by many factors. By capturing as much as possible of the risk management context information we make it easier to interpret risk management data in the future. From risk management perspective, context information can be further classified into three types. The *organization context information* describes the overall context of the project, that is, what is the application domain, what is the level of personnel experience and training, what methods and tools are used, reporting

procedures, organizational structure, etc. The definition of project context information collection procedures is generally relevant to the whole organization, not just to risk management, and thus it is the responsibility of the software measurement program to implement the necessary data collection procedures.

The second subtype of context information is *project information*, which defines the project itself and it includes the definition of the goals, customers, schedule, and constraints of the project. It also includes the definition of the risk management mandate for the project: The risk management mandate is a project-specific statement of the scope of risk management in a project. It defines which stakeholders are to be defended in risk management, stakeholders' priorities, which risks may be excluded from project management's risk management scope and how (e.g., organization management may be willing to take responsibility of some risks without burdening project management with any risk controlling responsibility), and define any other procedures that are not addressed by the risk management infrastructure.

The third context information type is *risk management infrastructure*, which deals with the risk management principles, methods, tools and practices that are available and implemented in the organization. We have adopted Hall's risk management evaluation framework as a tool to document and capture risk management infrastructure for each project (Hall 1995). Hall's framework provides a consistent way to take a snapshot of the risk management infrastructure during a project's duration.

We will discuss how these data are stored in the eRiskit database and discuss the main structures of the database in the following. The high-level database schema is presented in Figure 28.

Project context data is captured in a separate entity to allow the documentation of contextual information about the project, as well as changes in the context. However, the current implementation of context data in the application is simply an informal text field entry.

Stakeholders and goals play an important role in the application. Each stakeholder can be associated to several goals and each goal to several stakeholders. Stakeholders can also be active in several projects.

Each project is associated with a risk management mandate. In fact, risk management mandates can be updated and the old mandates are archived to provide a record of changes for experience capture.

Information about risks is captured in risk elements, according to Riskit method. The controlling action is always associated with risk event. This was done to simplify use of the system, even though in reality controlling actions can also influence other risk elements.

The utility loss rankings are associated to each stakeholder and, on the other hand, to risk reaction, which acts as the unique identifier for the effect set of each scenario.

Risk management action entity captures the risk management process enactment data, i.e., what Riskit process activities were performed, how much time they took, who was involved etc.

All key entities in the database have mechanisms to capture changes in their values during their life cycle. This is shown by the state entities in Figure 28, i.e., the system captures the history of changed values for risk factors, risk events, risk effects, loss rank, stakeholder, goal, and project entities.

During the course Tik-76.115 Software projects, in which the development of eRiskit application was started, we used Riskit method to manage the risks of that project. That information was inserted into the eRiskit database by hand at first and as the development progressed, using eRiskit. The information includes approximately 60 identified risks, about 20 risk scenarios as well as quite a large amount of other risk management data. This information has been used in testing the software and validating the application functionality.

Appendix C Glossary of Risk Management Terms

This appendix contains a glossary of key terms used in this document, the corresponding term in Finnish, and their definitions in English.

Accepted risks	Hyväksytyt riskit	Description of risks that the project owners have accepted and are thus excluded from project's normal risk management scope
Constraint	Rajoite	A limitation or rule that must be respected, e.g., " … while obeying all traffic regulations"
Controlling action	Kontrolloiva toimenpide	A proactive maneuver that is taken before risk occurs (or before it is known whether the risk has occurred)
Driver	Vaikutin	A goal that indicates a " direction" of intentions without clearly defined criteria for determining when the " goal" has been reached, e.g., " drive from A to B as fast as possible"
Expected utility loss	Koetun menetyksen odotusarvo	Expected utility loss(RS) = probability(RS) * utility loss(RS), where "RS" indicates a given risk scenario
Goal	Tavoite	A characteristic that the project or product should have. Goals in the Riskit method are categorized into objectives, drivers and constraints
Goal review	Tavoitteiden analysointi	A process step in risk management. The stated goals for the project are reviewed and refined. Implicit goals and constraints are defined explicitly. Stakeholders' associations with the goals are analyzed
Mitigation Strategy	Riskien pienennyssuunnitelma	A strategy that is used to lower the probability and utility loss of risk scenarios
Objective	Päämäärä	A goal that has an achievable, well-defined target level of achievement, e.g., " drive from A to B in one hour"
Project	Projekti	
Reaction	Reaktio	A (set of) corrective action(s) that are taken after the risk has occurred
Risk	Riski	A possibility of loss, the loss itself, or any characteristic, object or action that is associated with that possibility
Risk analysis	Riskianalyysi	A process step in risk management. Risks are classified and consolidated, risk scenarios are completed for main risk events and risk effects, probabilities and utility losses are estimated for all risk scenarios

Risk cluster	Riskiryhmä	A grouping of risk items
Risk control	Riskien kontrollointi	A process step in risk management. The selected controlling actions are implemented
Risk control planning	Kontrolloivien toimenpiteiden suunnittelu	A process step in risk management. The most important risks are selected for risk control planning, risk controlling actions are proposed for most important risks and the risk controlling actions that are to be implemented are selected
Risk controlling action impact delay	Kontrolloivan toimenpiteen vaikutuksen viive	The delay it takes for a controlling action to take effect from the time of it's implementation
Risk controlling action implementation margin	Kontrolloivien toimenpiteiden suorittamisen marginaali	The urgency of implementing controlling action. Risk controlling action implementation margin = The time of risk event occurrence - Risk controlling action impact delay
Risk effect	Riskivaikutus	The combined impact of risk event and resulting reactions to goals of the project
Risk element	Riskielementti	Any item in the Riskit Analysis Graph
Risk event	Riskitapahtuma	An occurrence of an incident with some negative consequences
Risk-event influence	Riskitapahtumavaikutus	An influence of a risk factor or a risk event to a risk event. E.g., a risk factor can increase or decrease a probability of a risk event.
Risk factor	Riskitekijä	A known fact or characteristic that influences some risk event
Risk identification	Riskien tunnistaminen	A process step in risk management. Potential threats to the project are identified using multiple approaches
Risk item	Riskialkio	A risk that has not been analyzed and categorized into risk elements or described in the Riskit Analysis Graph
Risk management authority	Riskienhallintavaltuutus	A definition of authority or budget available for risk management
Risk management action	Riskinhallintatapahtuma	An action that takes place in the risk management process, such as risk identification, risk analysis, and risk control planning.
Risk mgmt mandate	Riskienhallinta-mandaatti	An explicit definition of the scope and frequency of risk management
Risk mgmt mandate definition	Riskienhallinta-mandaatin määrittäminen	A process step in risk management. The scope and frequency of risk management and the relevant stakeholders are defined in this step
Risk monitoring	Riskien seuranta	A process step in risk management. Risk situation is monitored

Risk outcome	Riskin seuraamus	The resulting situation after the risk event but before any reactions have taken place
Risk prioritization	Riskien priorisointi	Ranking of risk scenarios. Risk scenario probabilities are estimated and utility losses of scenarios are ranked separately for each relevant stakeholder
Risk scenario	Riskiskenaario	A combination of risk elements that describe the causes, triggering events and the impact of a risk. Normally a scenario consists of a risk event, risk reaction and risk effect set
Risk scenario development	Riskiskenaarioiden luonti	Risk scenarios are documented using the Riskit Analysis Graph
Riskit	Riskit	Riskit risk management method
Riskit Analysis Graph	Riskit-analyysikaavio	A graphical formalism used to document risk scenarios in the Riskit method
Riskit controlling action taxonomy	Kontrolloivien toimenpiteiden luokittelu	A classification of risks. One such taxonomy has been defined by the Software Engineering Institute.
Riskit element review	Riskielementtien tarkastelu	The Riskit element review is based on the risk elements presented in the Riskit Analysis Graph. This technique simply calls for a focused review of all risk elements in a scenario and prompts participants to consider ways to influence the elements either by controlling them, finding alternatives or preventing them
Riskit Pareto ranking technique	Riskit Pareto priorisointitekniikka	A risk scenario ranking technique used in Riskit method
Risks item clustering	Riskien ryhmittely	The act of grouping risks into sets that contain similar risk items
Stakeholder	Asianomistaja	Any individual, group, organization, or institution who can affect or be affected by the software project or its results
Utility loss	Koettu menetys	The harm a stakeholder experiences on a set of risk effects in a situation

Appendix D Study 4 Goals

This appendix presents the goals and the associated questions and metrics of the DaimlerChrysler and Nokia empirical study.

In the following, the first level number indicates the goal level, second level numbering indicates characterizing questions, and the third level numbering lists the metrics. Reference numbers in square brackets refer to study internal documentation that is listed in chapter D.2.

D.1 Study Goals, Questions and Metrics

Goal 1: Characterize Riskit

Analyze	The Riskit method
in order to	Characterize it
With respect to	Its pros and cons
From perspective of	Risk management process owner
in the context of	DaimlerChrysler and Nokia projects
because	Understanding pros and cons help evaluate and improve the method.

1. Characterize Riskit
   - 1.1. How much extra time is spent on risk management
     - 1.1.1. Hour logs from the case study ([8]: table Risk mgmt action, field Effort)
     - 1.1.2. Risk management effort data from earlier projects ([6]: questions P1, P2)
     - 1.1.3. Effort spent in training and learning in this project ([1]: B11 and [2]: question M8)
   - 1.2. How much added complexity does Riskit cause
     - 1.2.1. Interview: subjective questions (log books, [6]: questions S4 and S5)
     - 1.2.2. Count the number of information items in risk scenarios ([8]: tables associated to Risk table)
   - 1.3. Are people confused?
     - 1.3.1. Interview: subjective questions (log books, [6]: questions S4, S5, S6)
     - 1.3.2. Follow-up questions: why ([6]: questions S7)
   - 1.4. Did risk management influence actual decisions made in the project
     - 1.4.1. Track recommendations ([8]: table Risk controlling action, field Status)
   - 1.5. Subjective assessment of the overall benefits and disadvantages
     - 1.5.1. Structured interview ([6]: questions S1, S2, S8)
   - 1.6. Open-ended questions on each Riskit component
     - 1.6.1. Structured interview ([3]: questions G3 and G4, [4]: questions ID6 and ID7, [6]: questions A4, A5, A10, A11, A15, A16, C2, C3, C7, C8, C13, C14, M4, M5, S1-S8)

Goal 2: Evaluate feasibility of Riskit

Analyze	The Riskit method
in order to	Evaluate it
with respect to	Its feasibility in industrial projects
from perspective of	Risk management process owner
in the context of	DaimlerChrysler and Nokia projects
because	Feasibility is critical for future use of the method.

2. Evaluate feasibility of Riskit
   - 2.1. How much effort is used?
     - 2.1.1. DB: risk management action entity ([8]: table Risk mgmt action, field Effort)
   - 2.2. What risks are identified?

2.2.1. DB: risk factor and risk event entities ([8]: table Risk factor and Risk event)

2.3. Subjective assessment of feasibility

2.3.1. Structured interview ([6]: questions S9, S10, S11, S12)

2.4. Open-ended questions on each Riskit component

2.4.1. Structured interview [9] (same as GQM 1.6.1)

Goal 3: Evaluate effectiveness and efficiency of Riskit

Analyze	The Riskit method
in order to	Evaluate it
with respect to	Its effectiveness, i.e., its impact on the project's risk situation. Its efficiency, i.e., its added value to risk analysis and project management vs. the added cost.
From perspective of	Risk management process owner
In the context of	DaimlerChrysler and Nokia projects
because	Feasibility is critical for future use of the method.

3. Evaluate effectiveness and efficiency of Riskit

3.1. How much effort is used?

3.1.1. DB: risk management action entity ([8]: table Risk mgmt action, field Effort)

3.2. What risks are identified?

3.2.1. DB: risk factor and risk event entities ([8]: table Risk factor and Risk event)

3.3. What risk controlling actions were proposed and implemented?

3.3.1. DB: risk controlling action entities ([8]: table Controlling action, field Status)

3.4. How much effort was spent in risk controlling actions?

3.4.1. DB: risk controlling action entity: effort data ([8]: table Controlling action, field EffortActual)

3.5. What was the perceived impact of risk management process?

3.5.1. Structured interview ([6]: questions S1, S2, S4, S8)

3.6. Is there any evidence of risks having been avoided by risk management activity?

3.6.1. Interview ([6]: questions S13)

3.6.2. DB: controlling action entity impact estimate ([8]: table Controlling action, field ImpactDesc)

Goal 4: Characterize applicability of Riskit components in different situations

Analyze	The Riskit method and its components
in order to	Characterize
with respect to	its applicability in different situations
from perspective of	Risk management process owner
in the context of	DaimlerChrysler and Nokia projects
because	This will help determine whether the method can be applied in other situations and how it might need to be adapted.

4. Characterize applicability of Riskit components in different situations

4.1. What is the project context

4.1.1. project context data characterization set ([8]: table Context)

4.2. What are the individual characteristics and backgrounds of participants?

4.2.1. Questionnaire ([1]: questions B1-B11)

4.3. How was Riskit applied?

4.3.1. Process fidelity ([3]: G1, [4]: question ID1, [6]: questions A1, A7, A14, C1, C6, C11, M1, F1, F2)

4.3.2. Daily logs

4.4. Open-ended questions on each Riskit component

4.4.1. Structured interview template (same as GQM 1.6.1)

Goal 5: Monitor risk factors and events

Analyze	Risk factors and risk events of the case study project
in order to	Monitor them
with respect to	Changes in their status
from perspective of	Risk management process owner and process manager
in the context of	DaimlerChrysler and Nokia projects
because	Will guide the risk monitoring activity

5. Monitor risk factors and events
   - 5.1. What changes take place in risk situation?
     - 5.1.1. DB: all history entities ([8])

Goal 6: Characterize stakeholder and goal concepts

Analyze	The stakeholder and goal concepts of the Riskit method
in order to	Characterize them
With respect to	Their pros and cons
From perspective of	Risk management process owner
in the context of	DaimlerChrysler and Nokia projects
because	Understanding pros and cons help evaluate and improve the method.

6. Characterize stakeholder and goal concepts
   - 6.1. Were new goals and stakeholders identified
     - 6.1.1. Goals and stakeholders before and after analysis ([8]: tables Goal state, Stakeholder state)
   - 6.2. Were there changes in identified goals and stakeholders?
     - 6.2.1. Stakeholder and goal documentation before and after goal review step ([8]: tables Goal state, Stakeholder state)
   - 6.3. What happened to risks that related to goals that were not initially identified
     - 6.3.1. DB ([8]: tables Goal state)
   - 6.4. How did users perceive the usefulness of these concepts ?
     - 6.4.1. Structured Interview ([3]: questions G2, G3, G4, G6)
   - 6.5. Did stakeholders and goals impact risk prioritization?
     - 6.5.1. DB ([8]: tables Goal state, Stakeholder state)
     - 6.5.2. Observation logs
     - 6.5.3. interruption analysis questions ([5]: IA1, IA2, IA3)
   - 6.6. Did stakeholders and goals impact risk control planning?
     - 6.6.1. DB ([8]: tables Goal state, Stakeholder state)
     - 6.6.2. Observation logs
     - 6.6.3. interruption analysis questions ([5]: IA4, IA5)

Goal 7: Characterize risk control planning activity of Riskit

Analyze	The risk control planning activity of the Riskit method
in order to	Characterize it
With respect to	Is additional support needed, how well can different options be identified by project personnel, are the current checklists and guidelines beneficial
from perspective of	Risk management process owner and project manager
in the context of	DaimlerChrysler and Nokia projects
because	This area is important but there is little empirical information about potential problems in this step.

7. Characterize risk control planning activity of Riskit
   - 7.1. Are current checklists used?
     - 7.1.1. Observation logs
     - 7.1.2. Structured interviews ([4]: ID1, ID10)
   - 7.2. What other support would be useful?
     - 7.2.1. Structured interview

Goal 8: Characterize risk information attribute usefulness

Analyze	The risk element information attributes of the Riskit method and DaimlerChrysler risk information sheets
in order to	Characterize their use
with respect to	Pros and cons; their added value vs. cost to the project and to risk mgmt process improvement
from perspective of	Risk management process owner and project manager
in the context of	DaimlerChrysler and Nokia projects
because	This area is important but there is little empirical information about potential problems in this step.

8. Characterize risk information attribute usefulness
   - 8.1. Are risk element information attributes used actively in the project, why, why not?
     - 8.1.1. Observation ([8]: database information)
     - 8.1.2. Structured interview ([7]: question DB4, DB5, DB6)
   - 8.2. How much time is spent on entering information?
     - 8.2.1. Structured interview [7]: question DB1, DB2, DB3)
     - 8.2.2. DB ([8]: effort data from records marked by value "Risk mgmt database" in field RiskitStep in table Risk mgmt action)
   - 8.3. What information is filled in, what is left out and why?
     - 8.3.1. Structured interviews [7]: question DB4, DB5)
     - 8.3.2. DB ([8]: database information)
   - 8.4. Is risk element attribute information beneficial?
     - 8.4.1. Structured Interview ([7]: question DB4, DB5, DB6)
     - 8.4.2. DB ([8]: database information)

Goal 9: Characterize risk controlling action impact estimation and tracking

Analyze	The risk controlling action impact tracking
in order to	Characterize how it is done
with respect to	Techniques used for tracking, approaches used for impact estimation, reliability of estimates
from perspective of	Risk management process owner
in the context of	DaimlerChrysler projects
because	This is important for estimating the impact of risk controlling actions and, consequently, the benefits of risk management

9. Characterize risk controlling action impact estimation and tracking
   - 9.1. How are risk controlling actions tracked?
     - 9.1.1. Observation (Log books)
     - 9.1.2. Structured interview ([6]: question C12)
   - 9.2. Are impact estimates done (why, why not), how are they done?
     - 9.2.1. Structured interview ([6]: question C16)
     - 9.2.2. DB ([8]: Field ImpactEstimate in table Controlling action)
   - 9.3. How reliable are the impact estimates?
     - 9.3.1. Structured interview ([6]: question
     - 9.3.2. DB ([8]: Fields ImpactEstimate and ImpactActual in table Controlling action)

Goal 10: Compare risk management infrastructure

Analyze	Risk management support organization
in order to	Compare project internal (train and go) and project external (facilitate) support structure
With respect to	Pros and cons
from perspective of	Risk management process owner
in the context of	DaimlerChrysler and Nokia projects
because	This will help determine how risk management should be supported in projects.

10. Compare risk management infrastructure
    - 10.1. How was support organized in a project?
      - 10.1.1. Observation [10]
      - 10.1.2. Interviews ([2]: question IS6)
    - 10.2. What problems were encountered in risk management?
      - 10.2.1. Log books
      - 10.2.2. Interviews ([3]: questions G3, [4]: questions ID6, [6]: questions A4, A10, A15, C2, C7, C13, M4, S2, S3, S5, S6, S7)
    - 10.3. What additional support would have been beneficial?
      - 10.3.1. Observation (log books)
      - 10.3.2. Questionnaire ([6]: questions S3)
    - 10.4. What available support was not used?
      - 10.4.1. Structured Interview ([2]: questions IS3, IS4, IS5, IS6 vs. [6]: question S14)
    - 10.5. How familiar were participants with risk management and the Riskit method?
      - 10.5.1. Amount of training given ([1]: questions B11 and [2]: M8)
      - 10.5.2. Structured Interview ([1] question B12)

Goal 11: Compare risk description sheets and Riskit Analysis Graphs

Analyze	Risk description sheets and Riskit Analysis Graphs
in order to	Compare their characteristics
with respect to	Information content; clarity of description; and user perceived benefits and disadvantages
from perspective of	Risk management process owner
in the context of	DaimlerChrysler and Nokia projects
because	The use of Riskit Analysis Graphs has been noted to be difficult

11. Compare risk description sheets and Riskit Analysis Graphs
    - 11.1. What is the information content in Riskit Analysis Graphs and risk description sheets
      - 11.1.1. comparison analysis
    - 11.2. How clearly is risk information presented
      - 11.2.1. Interviews ([1]: question A19)
    - 11.3. What are the benefits and disadvantages of the two approaches
      - 11.3.1. Structured interview ([1]: question A19)

Goal 12: Compare the DaimlerChrysler risk ranking matrix and Riskit Pareto efficient ranking technique

Analyze	DaimlerChrysler risk ranking matrix and Riskit Pareto efficient ranking technique
in order to	Compare them
with respect to	whether they yield same results; how much confidence users have on them; and what underlying assumptions they require
from perspective of	Risk management process owner
in the context of	DaimlerChrysler projects
because	The ranking methods may have implicit biases that influence the results

12. Compare the DaimlerChrysler risk ranking matrix and Riskit Pareto efficient ranking technique
    - 12.1. Are resulting rankings the same
      - 12.1.1. comparison analysis
    - 12.2. How much confidence users have on results
      - 12.2.1. Interviews ([6]: question 18)
    - 12.3. What are the theoretical or underlying limitations and assumptions for the approaches
      - 12.3.1. Results of analysis
    - 12.4. What justification did users have for ranking risks w.r.t. “impact” in the DaimlerChrysler method
      - 12.4.1. Interviews ([6]: question A17)

Goal 13: Compare checklist-based and focused brainstorming approaches for risk identification

Analyze	Checklist based and focused brainstorming techniques in risk identification
in order to	Compare them
With respect to	Coverage, i.e., do they yield different important risks Granularity, i.e.,
from perspective of	Risk management process owner
in the context of	DaimlerChrysler projects
because	One of DaimlerChrysler projects uses

13. Compare checklist-based and focussed brainstorming approaches for risk identification
    - 13.1. How many risks were identified?
      - 13.1.1. Session notes, log books, DB ([8])
      - 13.1.2. Meeting notes
      - 13.1.3. Interview ([4]: ID5)
    - 13.2. How much time was spent on risk identification?
      - 13.2.1. DB ([8]: field Effort in table Risk mgmt action in records marked by value "Risk identification" in field RiskitStep)
      - 13.2.2. Log books
    - 13.3. What types of risks were identified?
      - 13.3.1. DB ([8]: field ClassID in table Risk)
      - 13.3.2. TBQ [11] mapping
    - 13.4. Open-ended questions from participants
      - 13.4.1. Interviews ([4]: ID6, ID7, ID8)

Goal 14: Characterize experiences from the risk management database

Analyze	Risk management database
in order to	Characterize
With respect to	Usage experiences, feasibility, pros and cons,
from perspective of	Risk management process owner
in the context of	DaimlerChrysler and Nokia projects
because	Feedback from database implementation is important for its future development

14. Characterize experiences from the risk management database
    - 14.1. How was the database used?
      - 14.1.1. DB ([8])
      - 14.1.2. Interview ([7]: question DB0)
    - 14.2. What kind of experiences users had?
      - 14.2.1. Interviews ([7]: questions DB5, DB6, DB7, DB8)
    - 14.3. What are the disadvantages and benefits of using the database
      - 14.3.1. Interview ([7]: questions DB8, DB9)
    - 14.4. Did users perceive the use of the database to be feasible in a real project?
      - 14.4.1. Interview ([7]: questions DB10, DB11)
    - 14.5. How much effort was spent in using the database?
      - 14.5.1. Interview ([7]: questions DB1, DB2, DB3)
      - 14.5.2. DB ([8]: effort data from records marked by value "Risk mgmt database" in field RiskitStep in table Risk mgmt action)
    - 14.6. What are the immediate benefits from the database
      - 14.6.1. Interview ([7]: question DB9, DB12)
    - 14.7. What are the long-term benefits from the database
      - 14.7.1. Interview ([7]: question DB9, DB13)
    - 14.8. What are the disadvantages of using the database
      - 14.8.1. Interview ([7]: question DB8, DB13)
    - 14.9. What improvement ideas are recommended?
      - 14.9.1. Interview ([7]: question DB8, DB14)

D.2 References Used in the GQM Statements and Questions

[1] Kontio, J. Riskit Empirical Studies: Interviewee Background Questions. 1998.
Ref Type: Unpublished Work

[2] Kontio, J. Riskit Empirical Studies: Risk Management Infrastructure Questionnaire. 1998.
Ref Type: Unpublished Work

[3] Kontio, J. Riskit Empirical Studies: Goal Review Questionnaire. 1998.
Ref Type: Unpublished Work

[4] Kontio, J. Riskit Empirical Studies: Risk Identification Questionnaire. 1998.
Ref Type: Unpublished Work

[5] Kontio, J. Riskit Empirical Studies: Risk Analysis and Controlling Action Planning Session Instructions. 1998.
Ref Type: Unpublished Work

[6] Kontio, J. Riskit Empirical Studies: General Questionnaire. 1998.
Ref Type: Unpublished Work

[7] Kontio, J. Riskit Empirical Studies: Risk Management Database Questionnaire. 1998.
Ref Type: Unpublished Work

[8] Kontio, J. Riskit Database Definition. 1998.
Ref Type: Unpublished Work

[9] Kontio, J. Interviewing Guidelines: Risk Management Experiences. 1998.
Ref Type: Unpublished Work

[10] Kontio, J., Getto, G., and Landes, D. Experiences in improving risk management processes using the concepts of the Riskit method. 163-174. 1998. Proceedings of the Sixth International Symposium on the Foundations of Software Engineering (FSE-6).

[11] Carr, M. J., Konda, S. L., Monarch, I. A., Ulrich, F. C., and Walker, C. F., *Taxonomy-Based Risk Identification, SEI Technical Report SEI-93-TR-006* Pittsburgh, PA: Software Engineering Institute, 1993.

D.3 Interview Template

This section of Appendix D presents a structured interview template for the risk management experiences study done with Daimler-Benz and Nokia. This interview template is to be used to support consistent, semi-structured interviews for the cases that are analyzed in the study.

Interviewee Briefing

The interviewee should be briefed as follows:

> *This purpose of this interview is to collect your observations and experiences from the risk management activities in your project.*
>
> *It is of vital importance that you answer the questions as objectively and candidly as possible. We are using the interview information for research purposes only and, if you wish, we can guarantee total anonymity for your or your organization's participation in this study.*

Questions Background Information

1 Interviewee's name:
2 Position at the organization:
3 Role in the project:
4 Open characterization of project planning and management experience of the interviewee:
5 Years of experience in project management:

6 Training received in project planning or estimation:
7 Were you involved in the definition of project goals, schedule and project contract?
8 Who else was involved in this process?
9 How important was your role in it?
10 Years of experience in risk management:
11 How much training have you received in risk management?

Interview Questions

The interview will be carried out by main steps of the Riskit process.

Risk Management Infrastructure

In your own words, characterize your project's risk management infrastructure along the following main attributes:

12 *Culture* – the level of awareness about risk management and attitude towards risks and risk management. The risk management culture can be characterized by question as is organization risk-averse or risk-taking, is the discussion about risks encouraged, is risk management recognized as a legitimate activity.
13 Policy – the stated management commitment to risk management and how it is enforced.
14 Methods: what methods and techniques are used and supported for risk management.
15 Tools – what tools and templates are used in risk management.
16 Skills and competence – what risk management skills and competencies exist, what training is available and given to personnel for risk management.
17 Support structure – what type of organizational support exists to help perform risk management in projects, how much resources are made available for this task.
18 Experience capture process – what mechanisms exist to capture, accumulate and analyze risk management experience.

Risk Management Mandate

19 Was risk management mandate defined (informally or formally)?

Characterize whether the following attributes of the risk management mandate were defined at the beginning of the project and how they were characterized:

20 Objectives:
21 Scope:
22 Risk management authority:
23 Accepted risks:
24 Risk management procedures:
25 Stakeholders:

Goal Review

26 How were goals defined?
27 What was the impact of having goals defined?
28 What problems occurred in this step?
29 What do you think are the main benefits of the approach used?
30 What technique was most useful technique in this step?
31 What impact did the goal definition have on the project, in your opinion?

Risk Identification

32 How were risks identified:
33 What techniques were used?
34 How much time was spent?
35 Who participated?
36 How many risks were identified?
37 What problems occurred in this step?
38 What do you think are the main benefits of the approaches used?
39 What technique was most useful technique in this step?

40 How much confidence did you have in having had adequate coverage of the risks?

Risk Analysis

Risks Item Clustering

41 How were risks clustered?
42 How many groups?
43 What criteria were used for clustering?
44 What problems occurred in this step?
45 What do you think are the main benefits of the approach used?
46 What technique was most useful technique in this step?

Risk Scenario Development

47 Were risk scenarios defined?
48 How many scenarios were defined?
49 How complex were scenarios?
50 What problems occurred in this step?
51 What do you think are the main benefits of the approach used?
52 What technique was most useful technique in this step?
53 What impact did the scenarios have on the project, in your opinion?

Risk Prioritization

54 How were risks prioritized?
55 What problems occurred in this step?
56 What do you think are the main benefits of the approach used?
57 What technique was most useful technique in this step?
58 How much confidence did you have in having prioritized the risks correctly?

Risk Control Planning

Defining Risk Controlling Action

59 How were risk controlling actions defined?
60 What problems occurred in this step?
61 What do you think are the main benefits of the approach used?
62 What technique was most useful technique in this step?
63 How much confidence did you have in having had enough potential risk controlling actions considered?

Selecting Risk Controlling Action

64 How were risk controlling actions prioritized and selected?
65 What problems occurred in this step?
66 What do you think are the main benefits of the approach used?
67 What technique was most useful technique in this step?
68 How much confidence did you have in having selected the right risk controlling actions?

Risk Control

69 How were risk controlling actions implemented?
70 Was their implementation tracked?
71 What problems occurred in this step?
72 What do you think are the main benefits of the approach used, if any?
73 What technique was most useful technique in this step, if any?

Risk Monitoring

74 How was risk situation monitored?
75 Frequency of monitoring?
76 Responsibility?
77 What problems occurred in this step?

78 What do you think are the main benefits of the approach used?
79 What technique was most useful technique in this step?
80 How much confidence did you have in having performed the risk monitoring activity adequately?

Concluding questions

81 Overall, what was the impact of risk management in the project?
82 What are the most critical problem areas in risk management?
83 What techniques would require more clarification or help in the methods used?

Appendix E Evaluation Bias Reduction Checklist

This checklist has been developed to be used during the risk analysis step to remind participants about the potential biases that may influence risk analysis results. To use this checklist, it is recommended that users participate in Riskit training so that the underlying theory and concepts related to each question can be understood and taken into account in risk analysis.

1. Is there information available about past frequency of risk occurrence?
2. Does the situation specific information about the project justify adjustments to probabilities?
3. On how many similar observations or projects is the estimate based on?
4. How representative are the reference observations or projects?
5. Are you sure that you have used information about past events and projects only to obtain reference point for probabilities – and not to assume that "our luck must change now"?
6. Have you used several independent information sources?
7. Are the characteristics of the past and current project similar and relevant so that past data can be used in the assessment?
8. Have you used reliable information sources to obtain project status and background information, are you sure such information is reliable?
9. If you are using historical, time-series data, are you sure that you understand the reason for changes in data values? (watch out for the effect of regression to the mean)
10. Are you sure you have considered all relevant projects and examples, not only the ones that are recent or well known?
11. Have you involved people with experience in the type of project, domain or technology in risk identification and analysis?
12. Are you sure that the assumptions or causes of problems in the past projects are well founded and understood?
13. Are you sure that you have sufficiently taken into account the situation specific information about the project?
14. Have you considered the conjoint effects of several risks to the project?
15. Have you modeled the dependencies between risks?
16. Have you checked whether your effect sets actually account for worst and best case scenarios?
17. Are you sure that you have objectively evaluated all available data?
18. Are you sure that all stakeholder views and positions are accounted for?
19. Are there any special situations or circumstances that might change the project goals or success criteria?
20. Are there any unstated assumptions about the project?
21. Is there anything to be gained (e.g., more information) by postponing making decisions about risks?
22. What might be the disadvantages for not taking action on risks now?