Mari Honkonen

# The Use of Risk Management Methods in Software Projects

Master's thesis in
Information Systems
August 19, 1999

University of Jyväskylä
Department of Computer Science and Information Systems
Finland

**Abstract**

Honkonen Mari
The Use of Risk Management Methods in Software Projects/Honkonen Mari
Jyväskylä, Finland, University of Jyväskylä, 1999
78 p.
Master's thesis

Managing software projects has proved to be difficult. Projects typically exceed their budgets and overrun their schedules, and the quality of the system is often unsatisfactory. The economical consequences of those failures are noticeable and cannot be ignored by project managers. One mitigation means – risk management – examines practices that can prevent or at least decrease the exposure to such failures.

The goal of this study is to investigate risk management methods used in software development projects in Finland and to evaluate their usefulness. The research covers a theoretical review of the literature and an empirical field study among experienced project managers.

Empirical data was collected using semi-structured interviews. Ten companies and 31 managers (11 IS managers and 20 project managers) participated in the study.

As an outcome of the study we have formed two check lists for project managers. The first list includes the most important risk items observed by the project managers. The second list reiterates mitigation methods and their success levels. Furthermore, we examine the risk management traditions and practices of the participating companies, and as an outcome compare their ability to manage software project risks.

One empirical finding from our research is a novel risk item: potential risk exposure over the system life cycle. Observed risk mitigation strategies are in line with former studies, though no complex and refined strategies were mentioned. Overall, risk management is growing in importance in particular among companies which are not IT oriented. Overall software houses were slightly more advanced in risk management than other participating companies.

Keywords: Risk, management, software project, information system, success

## Table of Contents

APPENDICES

# 1 Introduction

As information technology (IT) is becoming more common in our everyday life, software projects are growing in size and number. Consequently the importance of information system (IS) development projects has increased, and problems connected to them have become more critical (Ropponen 1993). Many organizations are familiar with software projects' cost overruns and canceled system development efforts (Keil, Cule, Lyytinen & Schmidt 1998).

Software risk management has been studied in the past as a means to overcome these culprits. Risk management examines practices that can prevent or at least decrease the exposure to failure. A systematic way of assessing and controlling risks will presumably lead to better results rather than ignoring potential risk factors. One of the latest studies in this area is the international Delphi study (Keil *et al.* 1998). In that study three panels of experienced project managers in Hong Kong, Finland and the USA identified and ranked the most important risks in software development. Lists of risk factors formed during the Delphi study are presented in Appendix A (a complete list of risk factors) and Appendix B (a list of the most important risk factors). The present investigation is based on the results of the Delphi study as a follow-up study.

Risk mitigation strategies and their usefulness have not been studied systematically. Literature on software project risk management lacks practical models (Ropponen 1993) and information on the actual utilization of the methods. The focus of the research reported in this thesis is: *what is the state of the art in risk management practices in Finland and what are the success rates of different methods?* The topic involves an examination of items considered risky and consequent overall organization of risk management and mitigation activities.

The research consists of a theoretical review of the literature and an empirical field study conducted among experienced project managers. The empirical data was collected by semi-structured interviews, using two different data collection strategies. These data were subjected to systematic qualitative analysis (Galliers 1980; Hirsjärvi, Remes, &

Sajavaara 1997; see also Grönfors 1982; Mäkelä 1990; Silverman 1997; Yin1984; Yin 1989).

Ten companies participated in the study. In every participating company three interviews were carried out. In the IS manager's interview the background and the company's overall approach to risk management were explored. The IS manager also provided two experienced project managers for the two additional interviews. The first project manager was asked to recall one recently completed project and go through it in detail and to recall risks he (or she) had faced and how he had solved them. The second project manager was asked to go over a risk item list (list of the most important risk items from the Delphi study, Appendix B) and discuss whether he had experiences of those factors in any of the projects he had been involved in, and how he had resolved them.

The research yielded three kinds of results: a list of the most important risk items, a list of mitigation methods for ranked items and their success rates, and a general overview of risk management practices and their variation within ten companies.

The resulting list of the most important risk items resembles in a remarkable way earlier lists. When compared to the list obtained in the Delphi study (Keil *et al.* 1998), there were only minor changes. This is partially explained by the fact that our research method was based on past results. One new risk item, however, was identified: a potential risk exposure over the system life cycle. This is an interesting finding, since it reflects a change in the attitudes towards system development life cycle.

The risk mitigation methods used by project managers were simple but they found them to be relatively successful. Many participants were not conscious of published risk management theory and available tools. Some methods considered to be successful were mentioned only by a single participant. These strategies are worth examining further, since they are potential novel discoveries to more efficient risk management. The strategies were often related to a particular context, and their adaptation to different organizations or different projects can be difficult. Overall, risk identification was

generally carried out well, but mitigating actions tended to be inadequate. The participants were well aware of the lack of proper risk management methods, and they had positive attitudes towards improving risk management.

Software companies had slightly better competence in risk management. This is because they consider software development to be their core business, and they had consequently invested in the ability to avoid failures. The companies regarding IT as a strategic element had more profound practices in risk management than those where IT was seen as a support function.

The remaining part of this report is organized as follows. In Chapter 2 we review the earlier risk management literature. Our overview covers the concept of risk, typical risks in information system development, and suggested risk management methods. We also evaluate the benefits and costs of risk management. The chapter forms a framework for the consequent empirical research and analysis. The research methodology is described in Chapter 3. We report on the data collection as well as the data analysis methods deployed. In the next section, Chapter 4, we present the new results. We provide background information on companies, identified risk items, observed risk mitigation strategies, and features of the risk management activities. Chapter 5 includes the discussion of our findings, and the last chapter summarizes our research.

# 2 An Overview of the Software Risk Management Literature

Software project risks form a threat to successful system development. Various risks have been studied extensively, and several scholars have presented lists of potential risk items and outlined risk management methods (e.g. Alter & Ginzber 1978; McFarlan 1982; Boehm 1991; Barki, Rivard, & Talbot 1993). When software development methods and tools are changing rapidly some risks will also change. Software risk management research should be based on practices actually in use. In order to better understand current risks and ulitize relevant risk management methods, one has to examine a short history of software risk management.

In this chapter we explain how software risk management has been reported in the literature. Our overview concentrates on two interrelated issues: *risk* and *risk management.* First, we describe how the concept of risk is defined and analyze different risk conceptions. After defining the basic terms, we look at typical risks in information system development projects. Secondly, we define risk management by discussing the general concept of software project risk management and specific risk management methods suggested in the literature. Finally, we evaluate the benefits and cost of risk management.

## 2.1 Risk

Identification of typical risks in information system development gives an insight into the relative importance of different risk items. First we will concentrate on information systems in general, following this we will focus on specific software development risks.

### 2.1.1 The Concept of Risk

The history of the word 'risk' can be traced back several centuries. According to Charette (1989, p. 50), MacCrimmon and Wehrung (1986) have dated the introduction

of the word 'risk' back to at least the 17th century, when it partially replaced the use of word 'hazard'. Risk is assumed to be derived from Italian (Charette 1989, p. 50).

Although the concept of risk has been widely used in fields like insurance business, decision theory, and game theory (Boehm 1989, p. v), it is relatively young in the information system development area. McFarlan (1982), Boehm (1989, 1991) and Charette (1989) are considered the first ones to have introduced the term 'risk' into the information system field, and their early work has been fundamental to later development. Despite some later criticisms, the majority of their ideas are still valid.

Generally, the dictionary (Collins Cobuild 1995) explains 'risk' as: *'If there is a risk of something unpleasant, there is a possibility that it will happen.'* Several scholars suggest different definitions of the concept of risk in the context of system development. Though the definitions are partially similar, they emphasize different aspects of risk. In the following section we introduce some of these concepts (see Table 1), and form a definition to be used in this thesis.

TABLE 1. Definitions of risk.

| **Author** | **Definition** |
|---|---|
| Barki, Rivard and Talbot (1993, p. 206) | Software development risk is the product of project uncertainty and the magnitude of potential loss due to project failure. |
| Kontio, Getto and Landes (1998, p. 165) | Risk is a possibility of loss, the loss itself, or any characteristic, object or action that is associated with that possibility. |
| McFarlan (1982, p. 13) | Risk suggest exposure to negative consequences, like failure to obtain the benefits, exceeding the implementation budget or time limits, shortfalls in technical performance, and incompatibility problems with the selected hardware and software. |
| Ropponen and Lyytinen (1999) | Risk is a state or property of a development task or environment; which, if ignored, will increase the likelihood of project failure. |

The definition by Barki *et al.* (1993) is analogous to Boehm's definition of risk exposure. Boehm (1991, p. 33) defines risk exposure (RE) as a product of the probability of an unsatisfactory outcome (P(UO)) multiplied by the loss to the parties affected, if the outcome is unsatisfactory (L(UO)): RE = P(UO) * L(UO).

Kontio *et al.* (1998) and McFarlan (1982) emphasize the unpleasant consequences of a risk. Actually, in decision theory risks are considered to have both profit and loss attached to them (Crockford 1980, after Charette 1989, p. 50). Thereafter Crockford (ibid.) name risks which have only negative outcomes as static risks, and those which have both profit and loss associated with them as speculative risks. Speculative risks are obvious, for example, in stock trade, but in the case of software risks the majority of the literature concentrates on negative consequences, *i.e.* static risks. In this thesis, we focus therefore solely on negative outcomes qua risks.

Ropponen and Lyytinen (1999) add an important feature to the definition of risk. Risks can be ignored, or they can be affected. Charette (1989, p. 52) states that there can be events where loss is incurred, but if there is no choice available, it is not a risk item. Therefore, a sure loss should not be considered as a risk in that the risk involves a possibility to be affected by management actions.

Conceptually, a risk also differs from a problem. Current risky situations may cause unwanted negative consequences in the future (Ropponen 1993, p. 10). A problem has occurred when a risk has materialized. Insurance companies deal with risks and first aid deals with problems. Yet, distinguishing between the two terms can sometimes be difficult. Instead of identifying the top 10 risks, the project managers tend to manage the top 10 problems, as Carr states (1997, p. 21).

Figure 1 illustrates links between risk elements recognised in the Riskit method developed by Kontio (Kontio, Englund, & Basili 1996; Kontio 1997; Kontio *et al.* 1998). As can be seen in the figure, a risk consists of a probability and a loss. The loss in turn is related to the expectations, which are valued by the stakeholders. By stakeholder we mean constituencies like customers, users, project team members, maintainers, managers, and subcontractors. Observation of stakeholders' views is also in line with Boehm and Ross, who examined means to manage participants' diverging objectives in their Theory W (Boehm and Ross 1989). What for one participant is a risk, might for others be an advantage. Charette (1989, p. 51) has noticed that different stakeholders may see the same situation either as a risk, or as an opportunity. Lyytinen

and Hirschheim (1987, p. 263) have included both different perspectives of stakeholders and expectations in their definition of system failure as *'inability of an IS to meet a specific stakeholder group's expectations'*.

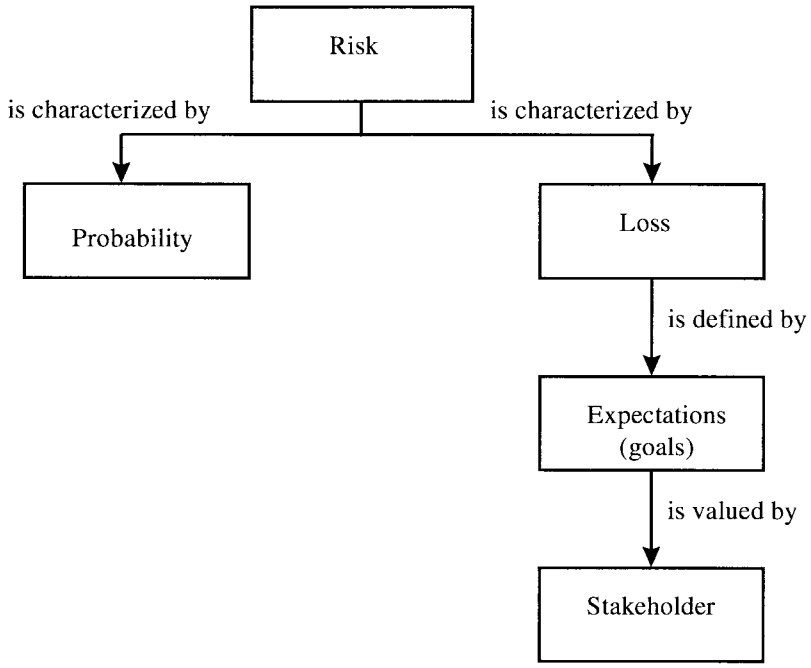



FIGURE 1. Definition of risk in the Riskit method (Kontio *et al.* 1998, p.165).

As can be noticed from the definitions, many scholars characterise risk as a combination of the probability of an event and its potential loss magnitude. Evidently, a risk has a negative consequence in the future and there should exist a possibility to affect the risk. Furthermore, the value of a risk depends on the stakeholders' point of view and expectations. As a combination of these characteristics we define risk in this thesis as follows:

> *Risk is a potential future loss, which affects at least one stakeholder and which can possibly be influenced by management actions.*

This definition includes four characteristics describing risk: probability, negative consequences in the future, stakeholders, and choice.

### 2.1.2 Risk in Information System Development

After defining the concept of risk we continue with introducing typical risks in information system development. Brainstorming and check lists are examples of techniques used to identify potential risk factors. We will concentrate on check lists, since they are commonly used and comparable. As Ropponen (1993, p. 11) points out, these risk items are not risks themselves, but they correspond to factors which may develop into risks and thus prevent successful completion of the project. Risk factor is thus *'a contingency that constitutes a serious threat to the successful completion of a software development project'* (Keil *et al.* 1998, p. 77). In this thesis, we use the words 'factor' and 'item' as synonyms. Naturally, developers of risk factors lists have sought to determine the most significant items. Thus, the lists can be used as indicators of the most typical risks in software development.

Boehm (1991, p. 35) has created a Top Ten list for risk items, which we present in Table 2. Although this list is one of the most popular ones, it has received considerable criticism lately: it is not based on typical business development situations and working methods have changed radically since the list was developed (Keil *et al.* 1998, p. 77). Recently, new lists of risk items have been reported, for example by Barki *et al.* (1993, p. 208), Fairley (1994, p. 60), and Moynihan (1997, p. 37). No two lists are similar, but they have many elements in common.

TABLE 2. Boehm's top ten risk items (Boehm 1991, p. 35).

| | **Risk item** |
|---|---|
| 1. | Personnel shortfalls |
| 2. | Unrealistic schedules and budgets |
| 3. | Developing wrong software functions |
| 4. | Developing wrong user interface |
| 5. | Gold plating |
| 6. | Continuing stream of requirements changes |
| 7. | Shortfalls in externally furnished components |
| 8. | Shortfalls in externally performed tasks |
| 9. | Real-time performance shortfalls |
| 10. | Straining computer science capabilities |

The most significant former research for the present survey is an international Delphi study reported by Keil *et al.* (1998). In that study three panels of experienced project managers in Hong Kong, Finland and the United States of America identified and ranked the most important risks for software development. The focus of the study was restricted to cover projects including coding. Altogether, eleven risk items were common to all three independent panels. Table 3 displays these items in order of their relative importance.

TABLE 3. Risk factors identified by all three panels (Keil *et al.* 1998, p. 78).

| | **Risk item** |
|---|---|
| 1. | Lack of top management commitment to the project |
| 2. | Failure to gain user commitment |
| 3. | Misunderstanding the requirements |
| 4. | Lack of adequate user involvement |
| 5. | Failure to manage end user expectations |
| 6. | Changing scope/objectives |
| 7. | Lack of required knowledge/skills in the project personnel |
| 8. | Lack of frozen requirements |
| 9. | Introduction of new technology |
| 10. | Insufficient/inappropriate staffing |
| 11. | Conflict between user departments |

The most important risk item in the Delphi study is not mentioned in Boehm's list. According to Keil *et al.* (1998, p. 82), one possible reason for the difference might be that Boehm has focused on risk factors which are under the strict control of the project manager. The Delphi study, on the other hand, has a much wider scope implicating that the project managers should also consider and assess risks which are beyond their total control.

We have given here examples of typical risk items. However, risks in information system development projects may consist of many other risks, and the spectrum of different types and kinds of risks in projects is broad (Charette 1989, p. 51). In the same vein, risks are usually interrelated, and this interaction makes risk management complicated and difficult.

## 2.2 Risk Management

Information system development projects are becoming larger and more critical. Problems connected with software projects have become more complex, and managing these projects has proved to be difficult. Many organizations are familiar with software project cost overruns, schedule slippage, and canceled system development efforts (e.g. Barki *et al.* 1993; Genuchten 1991; Conrow & Shishido 1997; Keil *et al.* 1998). Furthermore, the quality of the system is often unsatisfactory, or the systems are never used at all.

These serious problems with information system development projects have raised the question of improving risk mitigation strategies. One promising answer to the question is risk management methods. Risk management examines practices that can prevent or at least decrease the exposure to failure. For example, risk items can be recognized using brainstorming or a checklist, and project managers may benefit from the use of the list of 'risk management best practices' when they are choosing a suitable risk management strategy for their particular situation.

In this section, we introduce risk management objectives and reasons for risk management in software projects. We restrict the discussion from the overall information system development to software projects. Next, we examine risk management methods suggested in the literature. We introduce methods for software risk management in general, *i.e.* how to arrange development process in a risk-proof way, and specific mitigation strategies for particular risk items. We continue by describing risk management approaches based on the classification of the risk items. Finally, we evaluate the possible benefits and costs of risk management.

### 2.2.1 Software Project Risk Management

A project is *'a temporary endeavor undertaken to create a unique product or service'* (Duncan 1996, p. 4). Key characteristics of a project are uniqueness, a specific

objective, predetermined beginning and end, and a distinct organization (Pelin 1990). A project is a success, if it meets its objectives concerning economic goals, time limits, and produced results (ibid.).

Information system development projects may involve several activities, e.g. defining the requirements, planning, design, coding, and testing. Information system is defined here quite broadly as encompassing manual procedures as well.

> *'An information system is defined as an integrated, user-machine system for providing information to support operations, management, and decision-making functions in an organization. The system utilizes computer hardware and software; manual procedures; models for analysis, planning, control and decision-making; and database.'* (Davis and Olson 1985, p. 6)

Information system development includes software development as well. Software development is here defined according to Lyytinen, Mathiassen, and Ropponen (1998). In their opinion, software development covers requirements analysis, design, implementation, and organizational adoption of the software system. Their definition is quite broad, but convenient and defensible.

In this thesis our emphasis is on software development projects. Like other projects, software projects are one-time-only activities. They have a beginning and an end which are defined in advance. The difference distinguishing software projects from other information system development projects is that software projects include programming; their primary output is an immaterial software product. A software project is also highly people-intensive (Boehm and Ross 1989). Furthermore, controlling the process is complex, since the completion level of the output is difficult to specify.

Software project risk management concentrates on identifying and controlling risk items threatening the success of software projects. As Boehm (1989, p. 1) states, the objectives of software risk management is *'... to identify, address, and eliminate*

*software risk items before they become either threats to successful software operation or major sources of software rework.'*

Ropponen and Lyytinen (1997, p. 41) define software risk management as *'an attempt to formalize risk oriented correlates of development success into a readily applicable set of principles and practices'*. Risk management applies a variety of techniques and practices in assessing and controlling risk items in order to minimize the potential negative effects on the organization.

### 2.2.2 Methods for Risk Management

A systematic way of assessing and controlling risk is expected to lead to better results as opposed to ignoring potential risk factors. However, in order to be effective risk management should focus on the most important and the most critical risk factors, whereas minor items can be neglected, if their mitigation is more expensive than the harm they can cause. In fact, risk management should be an integrated part of normal project management, offering valuable tools to accomplish the software projects successfully. Charette (1989, p. 65) remarks that *'identifying a risk does not mean it will occur and we are trying to lower the impact if a risk does occur'*. There is no perfect way for risk management, which could be useful in every situation. Although organizations may have certain risk management policies, the actual operations depend on a particular situation. Choosing a method is problematic, because projects are complex and risks are multidimensional.

Software risk management should not be a one-off activity, but it should rather follow the project through its life-cycle. This principle of continuous risk management has also been emphasized by Williams, Walker, and Dorofee (1997) and Carr (1997, p. 24). Furthermore, risk management should be included in normal project management practices as one organic component. Integrating risk management into system development models may improve the use and impact of risk management methods. For example, Boehm (1988) has suggested a spiral model as an approach, where risks

analysis is accomplished in every 'cycle' of system development. The spiral model is displayed in Figure 2.

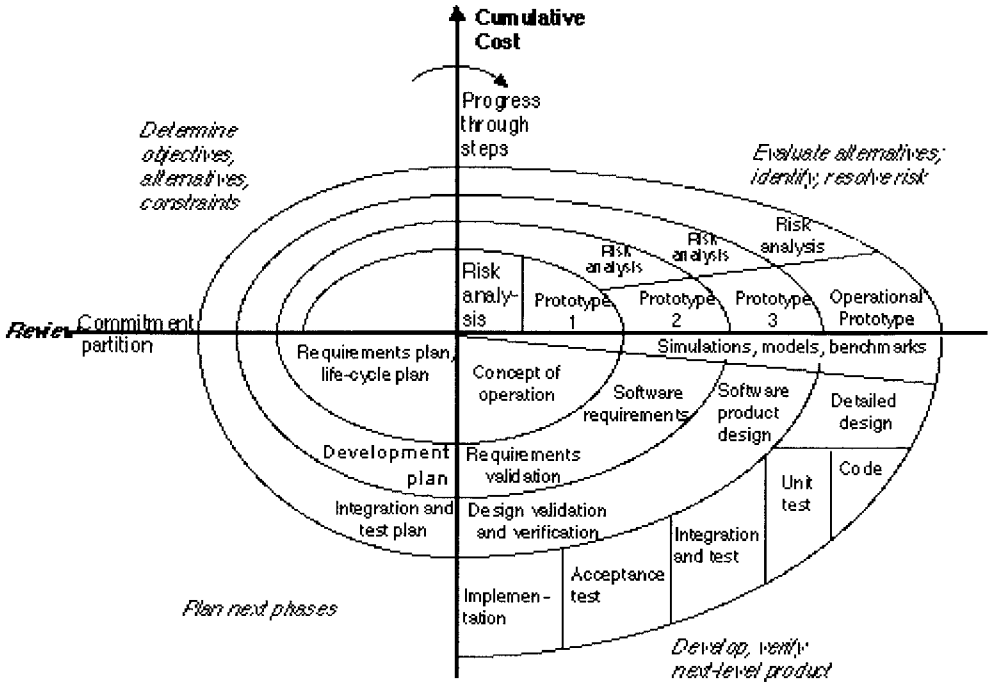



FIGURE 2. Spiral model of the software process (Boehm 1988, p. 64).

Boehm (1991, p. 34) and Charette (1989, p. 48) have both presented their classification of software risk management steps. Their models are quite similar. Therefore only Boehm's model is provided in Figure 3. Boehm has divided risk management into two parts: risk assessment and risk control. *Risk assessment* consists of risk identification, risk analysis, and risk prioritization, whereas *risk control* involves risk-management planning, risk resolution, and risk monitoring. In Charette's model the terms have slightly different names, but the principal idea is similar. After Charette, risk analysis consists of risk identification, risk estimation and risk evaluation, and consequently risk management consists of risk planning, risk control and risk monitoring (Charette 1989, p. 58).

Fairley (1994) has formed a seven-step procedure for risk management. These steps are: identify risk factors, assess risk probabilities and effects on the project, develop strategies to mitigate identified risks, monitor risk factors, invoke a contingency plan,

manage the crisis and recover from a crisis. According to Fairley, this procedure can be applied to all types of software projects.

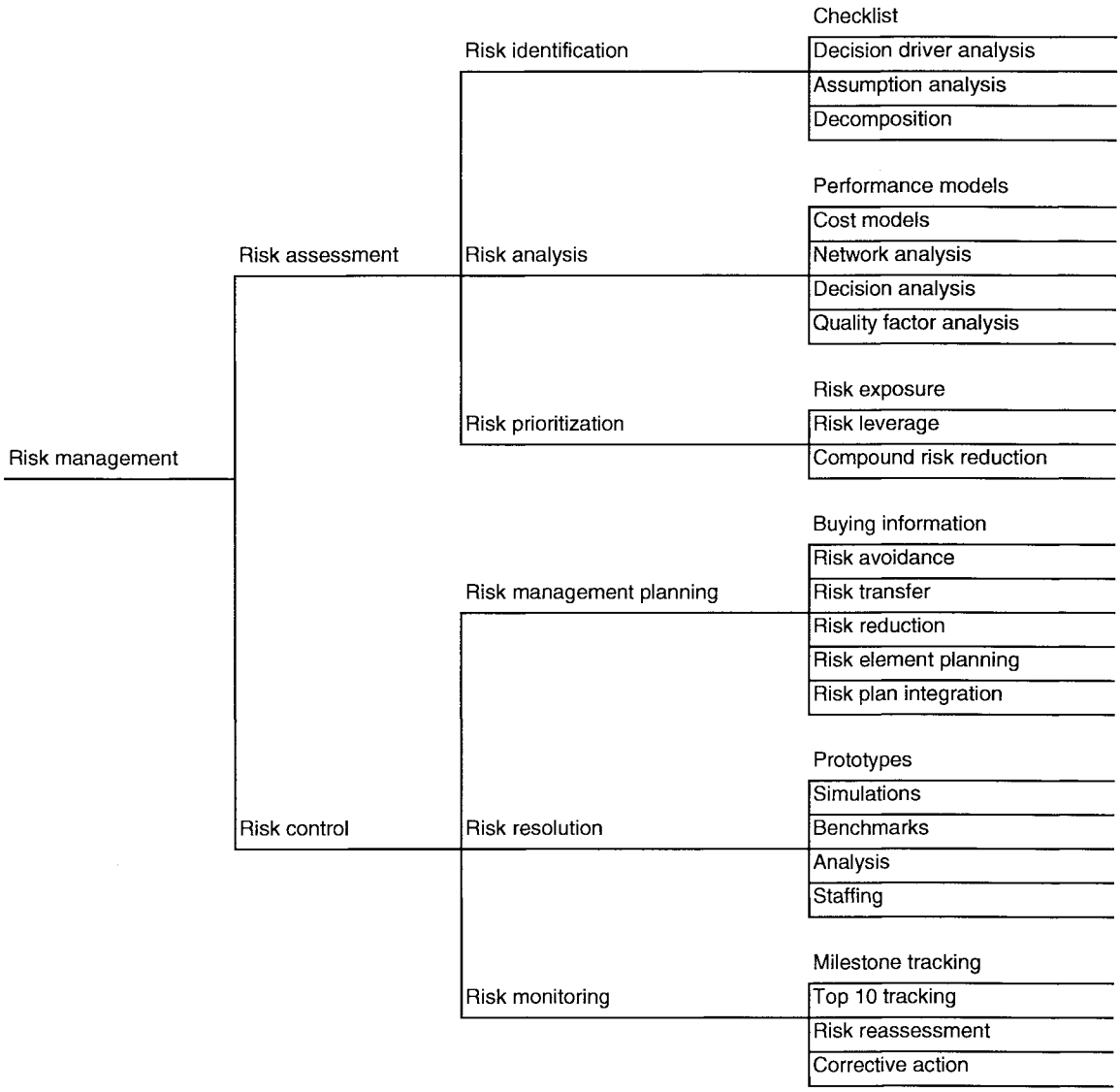



FIGURE 3. Software risk management steps by Boehm (1989, p. 2).

A new method for software project risk management is the Riskit method by Kontio (Kontio *et al.* 1996; Kontio 1997; Kontio *et al.* 1998). This method also has seven steps, which are: risk management mandate definition, goal review, risk identification, risk analysis, risk control planning, risk control and risk monitoring. The risk management cycle of the method is shown in Figure 4 as a dataflow diagram, and an overview of outputs and exit criteria of the Riskit process is summarized in Table 4.

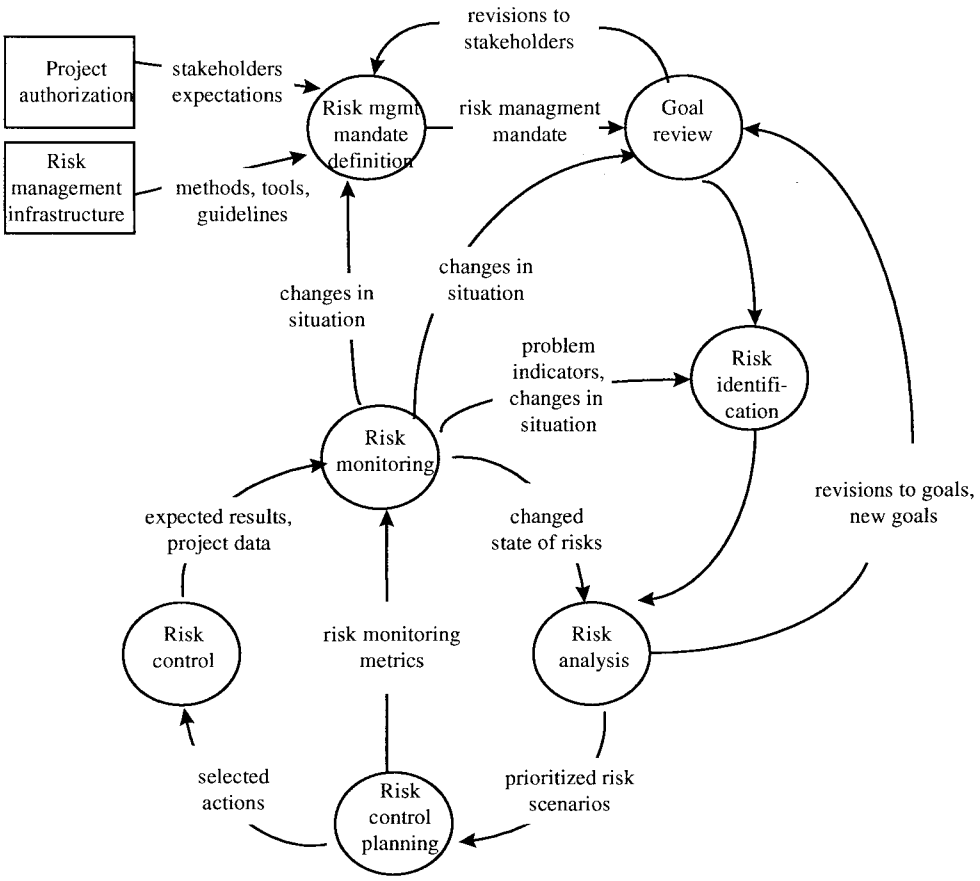



FIGURE 4. The Riskit risk management cycle (Kontio *et al.* 1998, p. 164).

TABLE 4. Overview of criteria of the Riskit process (Kontio *et al.* 1998, p. 164).

| **Riskit step** | **Description** | **Output** |
|---|---|---|
| Risk management mandate definition | Define the scope and frequency of risk management.<br>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.<br>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.<br>Complete risk scenarios for main risk events.<br>Estimate risk effects for all risk scenarios.<br>Estimate probabilities and utility losses of risk scenarios. | Completed Riskit analysis graphs for all analyzed risks.<br>Ranked risk scenarios. |
| Risk control planning | Select the most important risks for risk control planning.<br>Propose risk controlling actions for most important risks.<br>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. |

All these approaches have some characteristics in common. First, risks must be identified and analyzed. Risks are estimated by their relative importance and the most significant ones are included for further processing. Second, mitigation actions have to be planned and executed. In this stage, risks are mitigated by diverse strategies. Finally, risks have to be monitored. These approaches are procedural models explaining the logic of project risk management. Boehm's spiral model presents a means to integrate risk management in general development process. Risk management steps introduce a model of classical division of risk management activities. It also suggests mitigation methods to be used within each risk management task. This model, however, neglects the aspect of stakeholders' views, which is in turn included in the Riskit method. The Riskit risk management cycle shows how risk management actions introduced in the other models are actually carried out in practice.

Practical guidelines for risk management are presented among others by Karolak (1996) and Duncan (1996). The former concentrates on software engineering risk management and the latter forms a complete guide of project management. Both of these books introduce models for practical risk management which are easy to follow. The theoretical basis as well as practical adaptation of software risk management have been discussed profoundly by Ropponen (1999).

### 2.2.3 Managing Individual Risks

Besides introducing risk item lists some scholars (Alter and Ginzberg 1978; McFarlan 1982; Boehm 1991; Neo and Leong 1994) have suggested a number of management techniques for each risk item. The aim of matching methods with risks is to assist in choosing an appropriate method for a particular situation. Table 5 presents Boehm's suggestions of risk management techniques concerning his list of the Top Ten risk items.

Alter and Ginzberg associate a set of risk reducing strategies for each of their risk factor (see Table 6). In addition to Boehm's suggestion, they divide the methods into two categories: compensating (C) and inhibiting (I) strategies. *Compensating strategies* are

used to correct a previous error or problem after it has occurred, and *inhibiting strategies* are invoked to avoid a risk beforehand (Alter & Ginzberg 1978, p. 28).

TABLE 5. Risk items and their management methods (Boehm 1991, p. 35).

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

TABLE 6. Risk factors and their management methods (Alter & Ginzberg 1978, p. 29).

| **Risk factor** | **Risk-reducing strategies** |
|---|---|
| Designer lacking experience | Use prototypes (C). Use evolutionary approach (C). Use modular approach (C). Keep the system simple (C). |
| Nonexistent or unwilling user | Hide complexity (C). Avoid change (C). Obtain user participation (I). Obtain user commitment (I). Obtain management support (C). Sell the system (I). Insist on mandatory use (C). Permit voluntary use (C). Rely on diffusion and exposure (C). |
| Multiple users or designers | Obtain user participation (C). Obtain user commitment (C). Obtain management support (C). Provide training programs (C). Permit voluntary use (C). Rely on diffusion and exposure (C). Tailor system to people's capabilities (C). |
| Turnover | Obtain management support (C). Provide training programs (C). Provide ongoing assistance (C). |
| Lack of support | Obtain user participation (I). Obtain user commitment (I). Obtain management support (I). Sell the system (I). Permit voluntary use (C). Rely on diffusion and exposure (C). |
| Unspecified purpose or usage patterns | Use prototypes (C). Use evolutionary approach (C). Use modular approach (C). Obtain user participation (I). Provide training programs (C). |
| Unpredictable impact | Use prototypes (I). Use evolutionary approach (I). Obtain user participation (I). Obtain management support (C). Sell the system (C). |
| Technical and cost-effectiveness problems | Use prototypes (I). Use evolutionary approach (I). Use modular approach (I). Keep the system simple (I). |

Neither Boehm nor Alter and Ginzberg have reported the success rates of those strategies. Some of these risk-reducing strategies are useful for controlling multiple risks, whereas some of them are highly dependent on the context. The authors do not provide detailed guidelines in choosing a specific method among several strategies associated with each risk item. Moreover, Alter and Ginzberg admit that their recommendations seem like common sense (Alter & Ginzber 1978, p. 31).

### 2.2.4 Managing Classified Risks

The traditional approaches of identifying and handling individual risks has been lately criticized: managing a large variety of individual risks is difficult and time consuming (Keil *et al.* 1998; Cule, Schmidt, Lyytinen, & Keil 1999). A better solution is to classify risks into a manageable set of categories, and to develop mitigation strategies for each category. The same idea has been reported e.g. by Boehm and Ross (1989), Neo and Leong (1994), Keil *et al.* (1998), and Cule *et al.* (1999).

Boehm and Ross (1989, p. 908) divide project risks into two primary classes: generic risks and project-specific risks. *Generic risks* are shared by all projects, and they can be mitigated by planning and improving the development process. Use of standard methods and techniques removes, or at least reduces, generic risks. *Project-specific risks* depend on a particular project and its situation. Typical project-specific risks are personnel shortfalls, unrealistic schedules and budgets, and inappropriate requirements. Managing project-specific risks calls for risk mitigation plans adjusted for the given project. (Boehm and Ross 1989, p. 908)

Neo and Leong (1994) classify both risk factors and risk management strategies in four subsets. The categories of risks are: *task, organization, technology,* and *market factors.* Risk management strategies classes are: *risk pre-emption, risk reduction, risk isolation* and *risk sharing.* The authors evaluate the usefulness of each risk management strategy for dealing with different risk categories, but their analysis covered only one case project and the result cannot be generalized. (Neo & Leong 1994)

In the international Delphi study, Keil *et al.* (1998) presented a framework for classifying software project risks. The dimensions used in the classifying grid are the perceived importance and the perceived level of control (Figure 5). The grid is divided into four quadrants, all of which have different sets of risk factors and mitigation strategies.

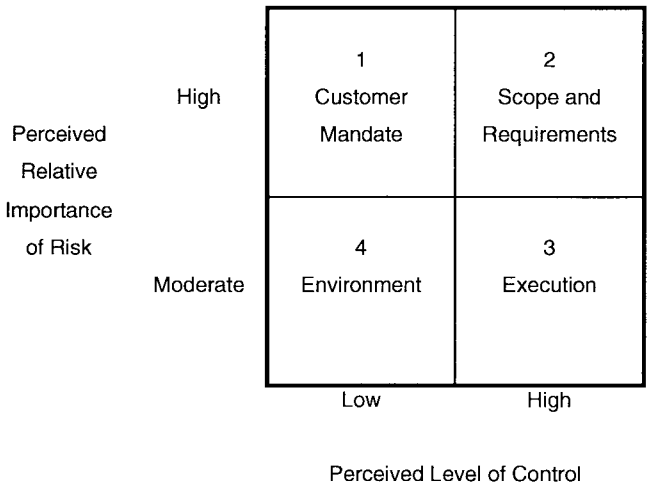



FIGURE 5. A risk categorization framework (Keil *et al.* 1998, p. 80).

The majority of the most important risks fell in quadrant 1. The name of the quadrant, *customer mandate*, refers to both senior managers and end-users of the system. Their commitment to the project is often critical to the success of it. After Keil *et al.* (1998, p. 80), the project manager has to have a clear charter or mandate to complete the project. Creating and maintaining good relationships with customers and promoting customer's commitment to the project are advised to be suitable risk mitigation strategies concerning this quadrant (ibid.).

Quadrant 2 consists of risks related to *the scope and the requirements*. Specifying the requirements in the beginning of the project is not always possible. As a result, the changes tend to cost and cause rework. Project managers must be able to distinguish between desired and necessary features of the forthcoming system. Users should actively participate in the project, and they should perceive the system as their own. Procedures of managing change and ambiguity are mentioned as appropriate risk mitigation methods for this quadrant. (Keil *et al.* 1998, p. 81)

Risks in quadrant 3 concern the actual *execution* of the project. These risks are mainly within the project manager's realm of control. As a mitigation strategy, the project managers must obey a disciplined development methodology. According to Keil *et al.* (1998, p. 81) the project managers has to be aware of possible threats like staffing problems or new technology, and they should have proper plans to cope with them.

Quadrant 4 includes risks caused by the *environment* of the project. The environment stands for both inside and outside the organization (Keil *et al.* 1998, p. 81). The project manager has little or no control over the risks in this quadrant. The most sensible mitigation strategy for dealing with these risks is contingency planning. Scenarios may be useful as well.

The same authors have presented also another classification of risks. This method is based on behavior model (Cule *et al.*, 1999). Figure 6 illustrates the different classes and links from the project manager's point of view.

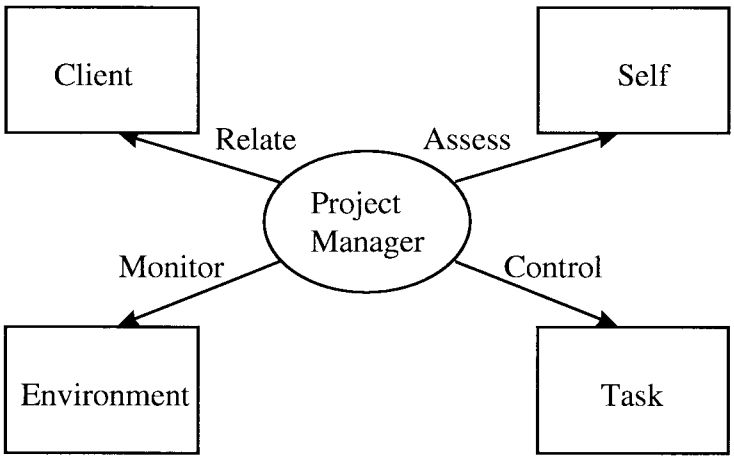



FIGURE 6. A risk classification and behavior model (Cule *et al.* 1999, p. 10).

Categories *self* and *task* are labeled as *inside risks*, since they are totally within the project manager's purview. Correspondingly, *Environment* and *client* are labeled as *outside risks* (Cule *et al.* 1999, p. 7). The project manager can *assess* the risks related to himself or herself. He can use independent auditors, disciplined assessment mechanism, or benchmarking (Cule *et al.* 1999, p.11). Risks in the category Task imply *control*

behavior by the project manager. After Cule *et al.* (1999, p. 12), this is the subject of education and should be ingrained in any experienced project manager. Environmental risks should be *monitored*, since the project manager can do little to them but he needs to maximize the time to react (ibid.). *Relate* is the behavior for handling risk in the category of Client. It includes relationship management, marketing, and maintaining long term relationships with customers. All of these mitigation strategies must be tailored to the particular project context (Cule *et al.* 1999, p. 21).

### 2.2.5 Effects of Risk Management

The significance of software risk management has been stated, for example, by Boehm:

> *'Software risk management is important primarily because it helps people avoid disasters, avoid rework, avoid overkill, and stimulate win-win situations on software projects.'* (Boehm 1989, p. 1)

Charette (1989, p. 66), on the other hand, has listed some of the benefits of risk analysis and management. The benefits are presented in Table 7.

TABLE 7. Benefits of risk analysis and management (Charette 1989, p. 66).

| Benefit |
|---|
| • Better and more well-defined perceptions of risks, clarification of options, trade-offs, their effects on a project, and their interactions. |
| • Systemization of thought, thereby providing a consistent view of the problem situation. |
| • Confidence that all available information has been accounted for, as well as the explicit identification of project assumptions. |
| • Improved credibility of plans produced, and communication of rationale for actions made, inside and outside the organization. |
| • Better contingency planning, and a better selection of reactions to those risks that do occur. |
| • More flexible assessment of the appropriate mix of ways of dealing with risk impacts, allowing for less reactive management, and more pro-active management. |
| • Better means to identify opportunities, and ways to take advantage of them. |
| • Feedback into the design and planning process in terms of ways of preventing or avoiding risks. |
| • Feed-forward into the construction and operation of projects in ways of mitigation the impacts of risks that do arise, in the form of responsible selection and contingency planning. |
| • Decisions compatible with project policies, goals, and objectives ensured. |
| • Insight, knowledge, and confidence for better decision making, and overall reduction in project exposure to risk. |

Intuitively, a systematic way of assessing and controlling risks will lead to better results rather than just ignoring potential risk factors. Ropponen and Lyytinen (1997, p. 41) pinpoint that we do not have enough empirical evidence of the practical usefulness of risk management. Consequently they conducted a research project on risk management practices in IS projects and reported that when 2-8% of the project's time is allocated to risk management, it can support accurate and reliable resource allocation, and help to manage complexity (Ropponen and Lyytinen 1997, p. 44). They also claim that too little or too much risk management can be useless or harmful, since both cases can decrease performance. Ropponen and Lyytinen also revealed two features affecting the ability to manage software risks. These features are cumulated experience in using methods, and the amount of resources spent (ibid.).

Costs of risk management lack also systematic research. The costs of risk analysis and management include expenditures of funds, time, personnel, and management involvement (Charette 1989, p. 69). Charette also reminds that ignoring risk analysis forms a substantial risk, which can lead to considerable losses like increased cost, business failure, loss of business credibility, litigation, contract cancellations, tarnished image, loss of revenue and slippage in deliveries (1989, p. 68).

In conclusion we may argue that the amount of work spent on risk management has a positive impact on risks related to development process; like the risk of time slippage in delivery or incorrect estimation of resource needs (Ropponen and Lyytinen 1997, p. 46). Yet, risk management will not guarantee a project's success (Charette 1989, p. 70). As Lister (1997, p. 20) states, *'any form of risk management is better than none'*, but *'nothing you can do will make your risks go away completely'*.

## 2.3 Summary of Literature Overview

In this section we have discussed risk and risk management, specially in the context of software projects. This literature review forms a theoretical basis for our research. In the beginning we analyzed some interpretations of the concept of risk. By comparing several definitions we achieved a better understanding of different aspects of the term

risk. After clarifying the terminology, we presented typical risks in information system development projects by probing risk item check lists.

The identification of potential risk factors is covered relatively well in the literature. Several identifying methods are suggested. Some of the methods, e.g. brainstorming, are simple and easy to perform, and they can be performed without extra training or expensive equipment. Check lists and questionaires are available for interested project managers both free and commercially.

Next, we examined objectives and significance of software project risk management. Software projects are difficult to control, and project managers are often inexperienced in systematic risk management. We introduced some risk management methods presented in the literature and analyzed their similarities. Furthermore, we considered specific mitigation strategies for particular risks, and suggested ways of classifying risk in groups and managing risks within each group diversely. Finally, we identified the effects of risk management.

Though the literature offers a plethora of risk management methods, the evidence of their actual use is poor. Do the project managers use these methods in practice? If they do, what are the advantages of using them? These questions have been reported only recently. Moreover, risk management methods are not widely used. The benefits of risk management are often based on ad hoc experience reports. Distinguishing both the costs and the benefits of risk management is difficult or even impossible.

# 3 Research Methodology

The literature of software risk management has a paucity of evidence of the actual use of the mitigation methods. The methods have been introduced in academic papers, but their utilization among practitioners has been studied only a little. In this study, we examine which risk mitigation methods have been observed to be applicable and effective. In addition, we will investigate the overall organization of the project management activity, and the integration of risk management in project management.

The methodology used in the research is explained in the following chapter. First, we describe research problems investigated in the study. After identifying objectives of the research, we introduce the research method and discuss limitations of the research approach. The collection of the empirical data is explained in detail, as well as the analyzing method of the material. The activities are examined with suitable accuracy in order to make the study repeatable. The aim of the precise description of the method is to enable the evaluation of the validity of it.

## 3.1 Research Problem and Method

The main goal of this study is to clarify how risk management methods are used in software projects and to evaluate their usefulness. Emphasis is on investigating the state of the art in risk management practices in Finland and to find out what are the success rates of different methods. We formulate the main research problem as follows:

> *What types of methods, if any, are used to control and mitigate risk items in software development projects in Finland, and how successful are these?*

Addressing these questions will deepen our understanding of the most common risk items, and how software risk management is organized in general. Analysis of risk management practices will also clarify risk factors and their mitigation methods, as well as overall approaches to risk management.

Keil *et al.* (1998) conducted a comprehensive study of the risk items just recently. Therefore we selected the 11-item risk list developed in their Delphi study (Appendix B) as a starting point for our research. While investigating risk management methods, we simultaneously sought to examine the validity of the developed risk item list. Our study was part of an international study, as a similar field-study was conducted in Hong Kong. This, of course, limited our possibilities of choosing a research methodology, since both studies should be carried out in the same way. In field-study research, the results are qualitative and they concern the present situation (Jenkins 1985). The generalizability of the results is low, as Benbasat (1984) reminds.

The empirical data was collected using semi-structured interviews. One strength of this research approach is its ability to describe real world situations, as Galliers (1992, p. 150) observes. Yet, interviewing is laborious and expensive, and it restricts the number of subjects to a modest level. A postal questionnaire could have reached more respondents. Yet, the nature of the research topic demands personal face-to-face conversations, because project managers are supposed to explain and describe their own experiences. Interviews offer a possibility for additional, specifying questions, whereas e.g. a mailed Delphi-study is unsuitable for collecting information of such unsettled processes. Besides, formulating an easy-to-answer questionnaire of the research topic would have been difficult, because the area is poorly understood. At least it would have demanded respondents to write long explanations, which would have raised the rejection rate and made analysis difficult. Furthermore, the subjects are better committed to interviews.

In this research we seek to establish a list of risk items, and a list of successful risk management methods for these most important risk items. The produced list of risk items can be used as a check-list in the risk identification phase. Project managers may use the resulting list of 'best practices' when they are choosing a suitable risk management strategy for their particular situation. The report of the overall organization of risk management activities offers IS managers a possibility to compare their risk management ability. Such a benchmarking might invoke fresh ideas, and therefore lead to improvements in risk management practices.

## 3.2 Data Collection

A letter introducing the research was sent to twenty information systems managers in August, 1998 (see Appendix C[1]). The criterion for the selection was that the companies included were assumed to form a representative sample of Finnish enterprises operating in the information technology area as well as in other business areas. The companies should not be competing with each other, and those concentrating on Internet technology were not included, since their area of operation is unstable. The majority of the companies were unknown to the researchers in advance, and in most cases their level of risk management was not known beforehand.

Ten of the companies decided to participate in the study[2]. This made the participation rate as high as 50%, which can be considered as a sign of noteworthy interest in the topic area.

In case of a possible sampling bias a part of the reasons for rejecting the participation was investigated. Two IS managers told that their departments were working under immense pressure because of resource problems, and they therefore avoided all additional work. One of the companies was undergoing a reorganization process. One company announced that they systematically refuse to participate in any research projects. These explanations are reasonable, and they do not implicate any bias in the sample. The reasons for rejections of the remaining five companies were not clarified by the researchers. Though the companies refused to participate in the research, they were interested in the issue. Two of them requested the results, and the results were sent to them when the study was finished.

The data collection was started using the Internet. Basic information on the companies, with one exception, was available on the WWW. The Internet offered a quick and easy

---

[1] The letter is in Finnish.
[2] Actually, one more company also announced their willingness to be involved with the study, but that was only after the deadline and unfortunately there was no opportunity to accept their announcement.

way to familiarize oneself with the economic key figures and business of the participating companies.

In every participating company three different interviews were executed. The interviews were semi-structured, and they were accomplished in Autumn 1998. The objectives and the organization of the interviews is illustrated in Figure 7. Appendix D provides the instructions for preparing for the interviews[3]. The content of interviews is explained below.

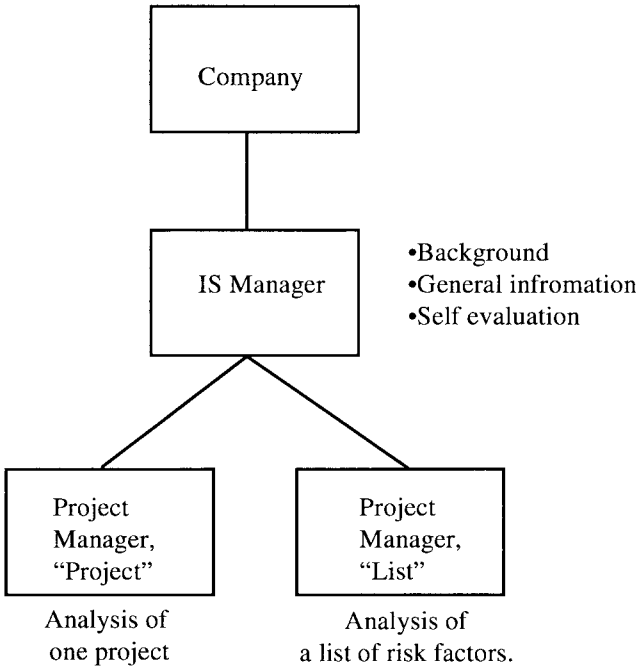



FIGURE 7. Organization of the interviews.

First, we interviewed the IS manager. In some cases we could not arrange this interview as the first one for practical reasons, but it was intended to be the opening session for every company. In the IS manager's interview the background of the company and their overall approach to risk management was explored. The interview concentrated on three principal areas: the organization, the tools and support, and the people. The outline of the interview topics is shown in Table 8. Topics like the size of the firm and its organizational structure, number of people in the information technology group, what

[3] The instructions are in Finnish.

technologies and methods they are using and so forth were discussed during the interview. The IS manager also evaluated their ability in risk management. In addition, the IS manager identified two experienced project managers for the two remaining interviews.

TABLE 8. The topics of conversation in the IS manager's interview.

| **Topic are** | **Questions** |
|---|---|
| The Organization | • size of the firm<br>• type of industry<br>• experience with IT<br>• company's mission and success of the firm<br>• organizational structure of the company and the IS department<br>• MIS-supported business activities |
| The Tools and Support | • system development tools and methods<br>• software (including programming language)<br>• the balance of resources between system development and maintenance |
| The People | • top management commitment<br>• IS experience level of the users<br>• age distribution of the users<br>• educational level of the IS personnel and the users |

The first identified project manager was asked to go over one recently completed project in detail and to recall risks he (she) had faced and how he had solved them. The aim of the interview was to analyze one single project from the risk management perspective. The project manager was asked to choose a project for consideration beforehand. The project was supposed to be a recently completed software project, which was still fresh in the memory. There were no additional requirements, e.g. size, success, area of operation, for the project. Every participating company was allowed to use the definition of 'project' of their own. Both successful projects and failures were welcomed in the study, and the project manager was encouraged to recall both successful and unsuccessful risk management methods he had used. The project manager was also advised to take the project documentation with him to the interview.

The other project manager was asked to 'go down' a risk item list (The list of the most important risk items identified in the Delphi study, Appendix B) and discuss whether he had experiences of those factors in any of the projects he had been involved in during his career. The list of risk items was sent to the project manager in advance, so that he

had time to prepare for the interview. He was advised not to allow the other project manager (the one interviewed in the single project) to see the list beforehand, since it might have biased the other interview and prohibited the triangulation of the data.

The aim of the IS manager interview was to fix the context, whereas information of risks and their management methods was collected in the project managers' interviews. The use of two different types of project mangers' interviews within each company is justified by the nature of our study. The type of data collection method is known to affect risk items mentioned by project mangers, since the unit of study was either a single project, or a single person. By this procedure we sought to increase the reliability and the validity of the study. Furthermore, we aimed to test whether the 11-item list of the Delphi study is exhaustive or whether some essential item were missing.

In our study the focus of interest is on the project managers' (and IS managers') point of view, regardless of other stakeholders of a software project. This restriction is made because of the unclear position of the other stakeholders, which is not necessarily defined in a formal way. The project manager acts as an connecting link between all these parties. Balancing diverging opinions is the duty of a project manager. Concentrating on project managers was supposed to offer the best possible results with limited research resources.

All the participants were asked about their education and experiences in project management. These questions are shown in Table 9. Although each type of interview had a different focus, the conversations were not restricted. The interviewees were also encouraged to share their experiences in risks management in general.

TABLE 9. Questions on the background of the interviewees.

| Question | Scale |
|---|---|
| The level of education | Doctor/Master/Bachelor/Other |
| The subject area of education | Technological/Business/Mathematical/Other |
| Number of projects | Number |
| Size of projects (smallest, largest) | Work years |
| Education in project management | None/Some/Moderately/Plenty |
| Education in risk management | None/Some/Moderately/Plenty |

Before the collection of the data we conducted a test interview. It was arranged in a company which was not included in the twenty companies invited to participate in the research. The principal goal of the interview was to explore the success of an interview where a project manager analyses one project. This type of interview was supposed to be the most difficult one. Besides analyzing a project, however, the test interview covered parts of all three types of interviews. An experienced project manager answered some of the questions addressed later to the IS managers, and went through the list of risk factors as well. The interview was successful, and it increased confidence in the chosen research method. Based on experiences from the test interview, we were prepared to explain the risk items carefully in Finnish and to answer specifying questions. We did not made changes to the interview questions, but added the evaluation of risk management ability to IS manager's interview topics.

All interviews were recorded with the permission of the interviewees. Afterwards the tapes were transcribed. The interviewees reviewed the transcriptions and made changes if necessary. The participants had the opportunity to correct possible errors (usually years and numbers), enter new comments, or remove details they did not want to be documented. All the analyzed data is based on the checked transcripts, not on the tapes. Some of the subjects wanted to have the tapes returned after the transcription. The tapes were delivered to them, and the rest of the tapes were re-recorded with radio programs.

Altogether we arranged thirty interviews. Professor Kalle Lyytinen attended most of the IS managers' interviews, whereas the author conducted all of them. Both interviewers made questions to the interviewees. In addition, the author took care of the tape recorder and wrote down the most essential parts of the conversation. The notes were made to ensure the preservation of the results in case the recording failed. Actually, this occurred twice. Normally the interviews lasted little more than one hour, the average total time per company being four hours. The interviews were arranged in peaceful places, e.g. negotiation rooms, and usually they were not interrupted. The annual reports of 1997 were collected from the companies if they were available, and some of the interviewees provided the interviewers with project documents.

## 3.3 Data Analysis

The IS managers' interviews were analyzed separately from the project managers' interviews. The data from the ten IS managers interviews was organized on a spreadsheet. The analysis was mainly qualitative[4]. It focused on describing variation in the background information and finding patterns in risk management practices.

The main emphasis of the analysis was placed on the project manager's interviews. Ten of them concentrated on the risk item list and ten on the analysis of a project. In addition to analyzing all the results together, we also compared the results of the two different types of interviews with each other.

In the analysis phase, risks and their management methods were collected from the transcripts. Risks were given a consecutive number for identification, and the success level of the mitigation methods was evaluated. Suggested countermeasures were evaluated[5] by using five levels of 'success' as presented in Table 10. Similar risks were grouped together. Methods connected to each risk were listed jointly, and similar methods within each risk were grouped together.

TABLE 10. The levels of 'success' used in the research.

| Success Value | Description |
|---|---|
| 1 | Failure |
| 2 | Not very effective |
| 3 | Moderate success |
| 4 | Fairly successful |
| 5 | Absolute success |

The criteria of evaluating the success levels of the methods is explained in the following paragraph. Methods, which have been unsuccessful and which the interviewer claims never work, were given a success value of 1. Using these methods does not remove

[4] In order to make the results comparable, the analysis included quantification. However, the conclusions are drawn qualitatively, since no quantitative (statistical) method was used in the analysis.

[5] Notice that these values were given by the researcher, not by the respondents. The values are based on the interviewee's verbal description of the success of the method.

risks. If a method has not been very effective, it was assigned a success value of 2. These methods might improve the situation, but they alone are inadequate. Success value 3 covers all the methods, which are useful, but not necessarily particularly effective. These methods can be considered either successful or ineffective, depending on the context. Possible unclear cases will be included in this 'neutral' category. Methods, which have proved to be quite successful, but which cannot be utilized on every occasion, were categorized into success level 4. The participants found these methods useful, but cannot guarantee their success in different contexts. The highest success level includes only methods which were an 'absolute' success.

After an identification of risks, the risks were connected to the 53-risk factor list as shown in Appendix A (a complete list of risk factors from the Delphi study). The list consists of 14 categories of risks. Some of the risks mentioned by project managers belonged to two different categories, while some of them did not belong to any of the previously identified groups. New categories were defined. Overall, we identified sixteen categories of risks, and had 276 observations of risks items.

The final step of the analysis concentrated on the most important risks. The risk items identified by at least half of the project managers, *i.e.* ten persons, were accepted for further analyses. The project managers did not themselves arrange risk items in relative order. The ranking of risk items and their selection for further analysis is based solely on frequencies, *i.e.* how often they were mentioned, not how important they were considered to be.

Similar and analogous countermeasures within each risk were grouped together. Each group was assigned three indicator values: the *average success level* of the countermeasures belonging to the group, the number of persons mentioning the countermeasures belonging to the group (*frequency*), and the *relative importance* of the measure, which was formed by multiplying the other two indicators (Relative Importance = Success Level * No# of Persons). The relative importance was used to arrange the methods so that the 'relatively' most successful and general methods were ranked highest.

We also adapted Leavitt's (1964) socio-technical model for organizational change to categorize the risks and their resolution techniques. This analysis method has been introduced by Ropponen (1999), and the theory behind it is explained by Lyytinen *et al.* (1998). They used the model to analyze four classical approaches of risk management: Alter and Ginzberg (1978), Boehm (1991), Davis (1982), and McFarlan (1982) (Lyytinen *et al.* 1998). The model is described in Figure 8. The key element of the model is to understand how organizations or socio-technical systems react to change. The elements of the model are related to each other, and a change in one component will cause changes in the others.

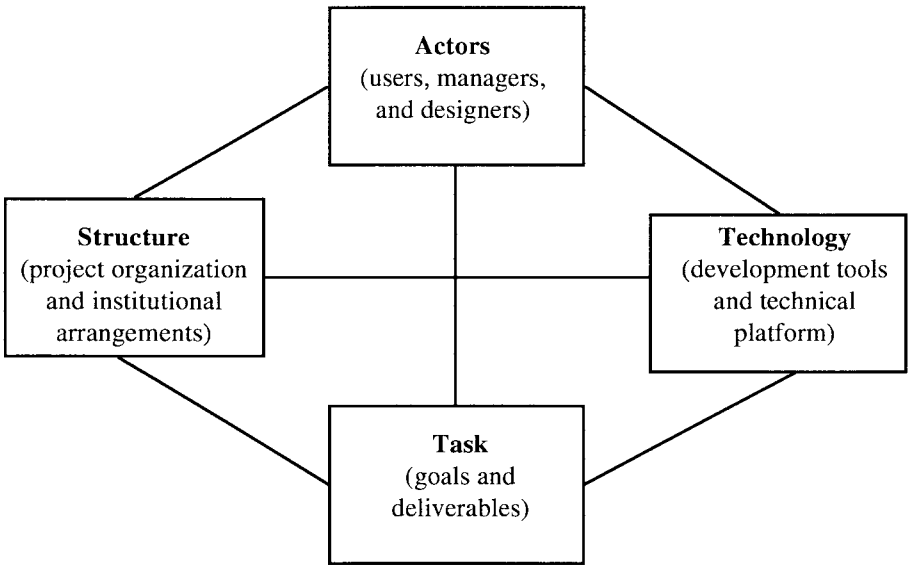



FIGURE 8. A Socio-technical Model of System Development (Ropponen 1999, p. 119).

A system development *task* describes project goals and deliverables. It denotes what should be accomplished, and how the process is implemented. *Structure* refers to the organizational aspects of system development. It represents arrangements of communication, authority, and work flow. *Actors* are individuals or groups who are somehow connected to the project. They include developers, managers, end-users, and customers. The actors can either have an effect on the project, or their work might be affected by the system produced in the project. *Technology* covers the technological infrastructure, tools, and methods of development and implementation. (Ropponen 1999)

The risk items and the resolution techniques were categorized according to the socio-technical model. Furthermore, the mitigation methods were classified as compensating or inhibiting strategies introduced by Alter and Ginzberg (1978, p. 28).

## 3.4 Summary of Research Design

The goal of this research is to clarify which methods are used in software risk management. This question was inspired by the lack of empirical research on the usefulness and applicability of the risk mitigation methods as suggested in the literature. At the same time we also studied which risk items are considered to be the most important, and how effective are the mitigation methods. In addition, we gathered information on the overall organization of the risk management activity.

The main data collection method was interviewing. The interviewing method is suitable when studying the state of the art at a particular point in time (Galliers 1992, p. 50). Furthermore, personal contact with the subjects offered a possibility to specify and check the interpretation of the data. The reliability and validity of the data was a concern in this study. Therefore the transcripts were verified by the interviewees.

The data analysis was mainly qualitative. The interpretation of the data was based on classification, and calculation of simple indicator values. No complex statistical techniques were utilized, since the size of the sample was not large enough. Achieving generalizability is difficult in qualitative research. In most cases the research objective is in revealing facts rather than in verifying a hypothesis (Hirsjärvi *et al.* 1997, p. 161).

# 4 Research Results

In this chapter we present the results of the study. First, we describe background information of the participating companies, interviewees, and the analyzed projects. This gives an indication of the scale and the scope of the study. We also discuss project management methods used in the participating companies, and the integration of risk management into project management. We continue with risk items identified by project managers. Risk items are presented in groups, indicating the most difficult areas, and in terms of individual risks. Finally, we introduce risk management methods and their success rates. In this stage of the analysis, we concentrate on the most significant risk items. Only those risks, which at least half of the project managers have mentioned, are included for further analysis.

## 4.1 Background Information

The research included three kinds of study subjects: companies, interviewees, and projects. This section describes the size and operating area of the participating companies, as well as their IT organization. The education and the experience of the interviewed persons is illustrated with a few figures. The projects are presented shortly to give an impression of their size and domain area.

### 4.1.1 Companies Included in the Sample

The ten participating companies operate in the information technology area (4 companies) as well as in other businesses (6). These business areas include industry (2), trade (1), transportation (1), and energy (2).

Turnover of the companies is presented in Figure 9. Six of the companies have experienced strong growth during the recent years. The companies are located geographically in different areas of the country, following the overall location of industry in Finland.

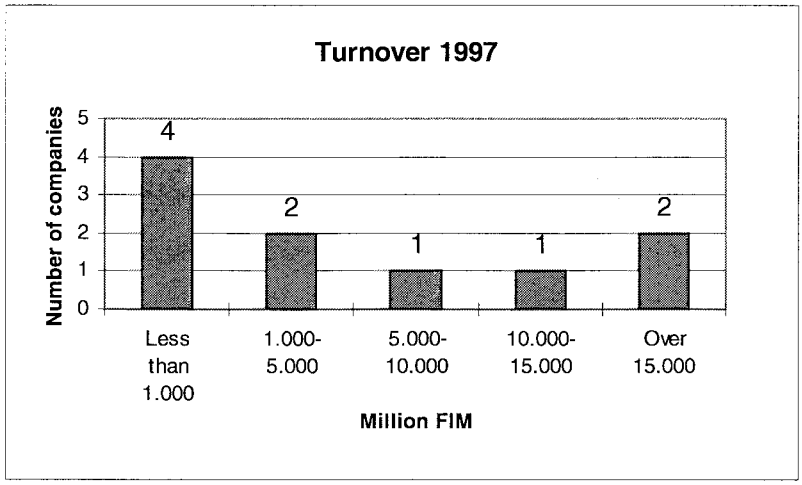



FIGURE 9. Turnover[6].

The total personnel of the companies is illustrated in Figure 10. Software companies did not report having specific IT-departments. Among other firms, the budget of the IT-departments varied from a couple of million FIM to over 200 million FIM, and the number of IT-personnel from less than ten to over 200 persons. The figures for some companies were not available.

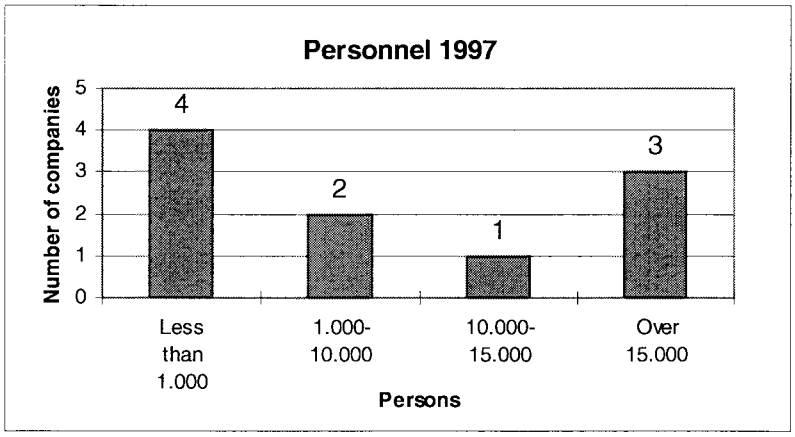



FIGURE 10. Personnel

The amount of resources devoted to maintenance has increased in almost all the companies, though they insist on concentrating on system development. As one of the IS managers said: *'Half of the personnel should develop new systems, but in practice all of*

[6] FIM 100 = US $ 18.52 (as of 31 December, 1997).

*them maintain the old ones'*.[7] Giving exact figures of the balance of devoted resources between system development and maintenance was difficult, but those who had collected such information reported about 40-50% of the resources being focused on new development.

The significance of the IT varied from being an essential part of the company's mission to more of a support activity. Overall, the organizations were experienced in their use of IT, and some of them were expecting competitive advantage to be achieved by utilizing information technology. Half of the companies had a MIS-system of some kind. The IS managers told their top management to be committed to IT, or at least the commitment had increased lately. There was one exception to this situation.

The tools and methods of system development covered all normal techniques. Many companies had outsourced development, and they did not have any specific tools or methods. IS managers mentioned several programming languages that were in use. Those mentioned most often were C and C++.

The age of the IS department (or corresponding unit) personnel was relatively young in two companies, relatively old in two companies, and more even in one company. The remaining five companies had a gap in the age groups: they had both old and young employees, but were missing 'middle-aged' workers. In three companies the typical education among IT personnel was a master's degree, whereas in five it was a bachelor's degree. Two companies had a majority of non-graduate workers.

### 4.1.2 Background of the Interviewees

Altogether we arranged 30 interviews. Since two people attended one IS manager interview, we had a total number of 31 interviewees. Information on the interviewee's educational background is presented in Figures 11, 12, and 13. Most of the interviewees had a university or college level education background, and two of them had a doctoral degree. The number of those in the group 'other', however, can be considered high.

---

[7] The actual interview was conducted in Finnish.

Members of this group had either a vocational degree, or no education other than comprehensive school. It was noteworthy that four interviewees had two degrees, typically a technical and a business degree.

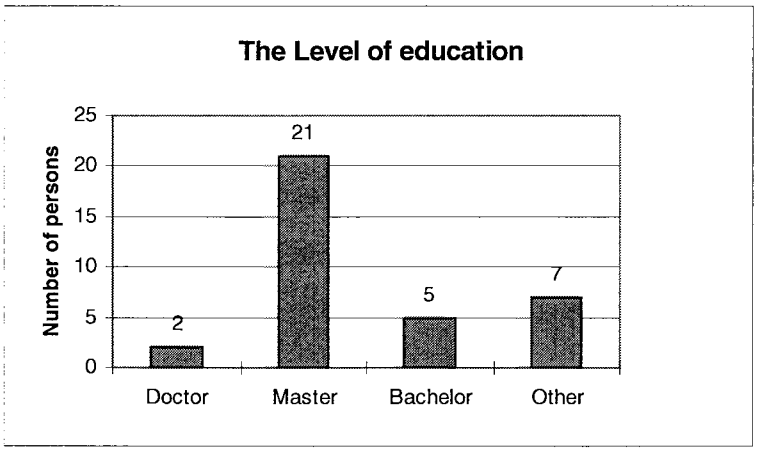



FIGURE 11. The level of education of the interviewees.

The distribution of the level and type of education is flat, as Figures 11 and 12 show. Three fields from where IT professionals have traditionally come from are: business, mathematics, and engineering. All these areas are more or less closely related to computers. The group 'other' consists of a large variety of educational experience, since several people had specialized in computing only during their work life. In fact, the technical group includes persons whose original education had little to do with computers.

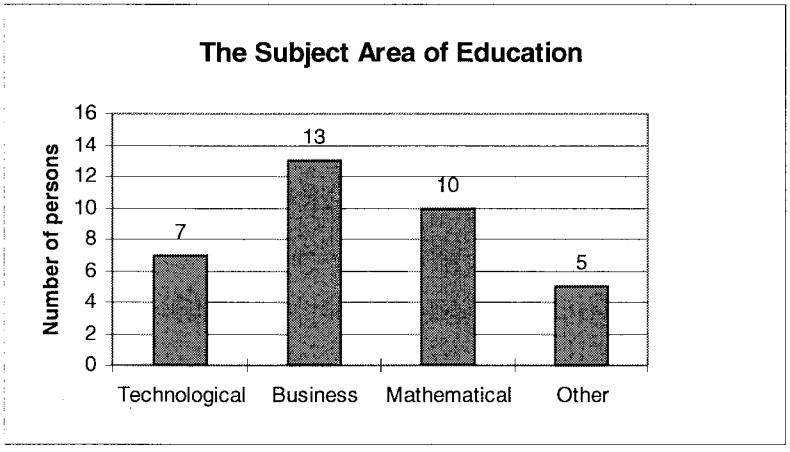



FIGURE 12. The subject area of education of the interviewees.

Figure 13 presents the level of education in project management and in risk management. Overall, the interviewees claimed to have sufficient education in project management. Yet, the majority of them had hardly any education in risk management.

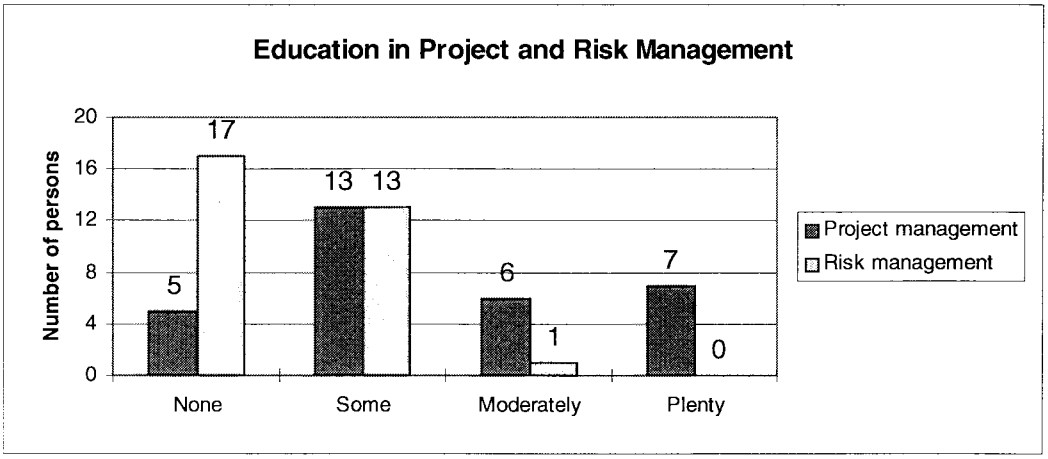



FIGURE 13. Education in project management and risk management.

Interviewed project managers, with a few exceptions, were experienced. Most of them had taken part in over twenty software projects, and some of the projects they had managed had been large. The project managers had altogether experiences of over 430 projects. The number of projects per project manager is shown in Figure 14 and the size of the largest projects in Figure 15.

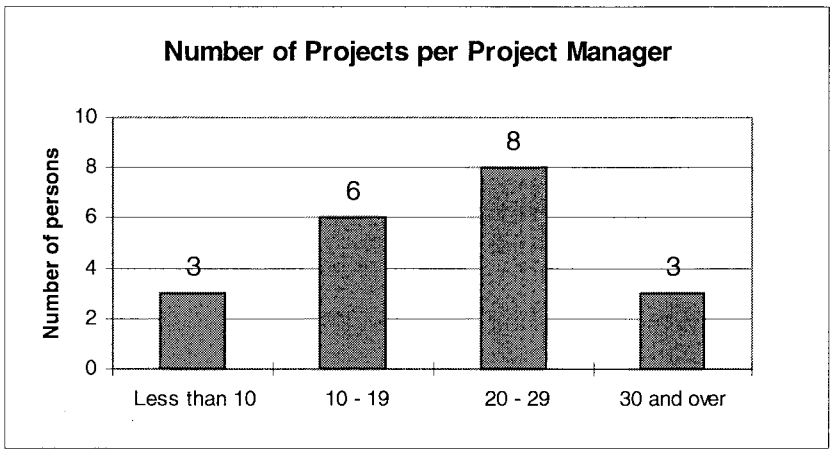



FIGURE 14. Number of projects per project manager.

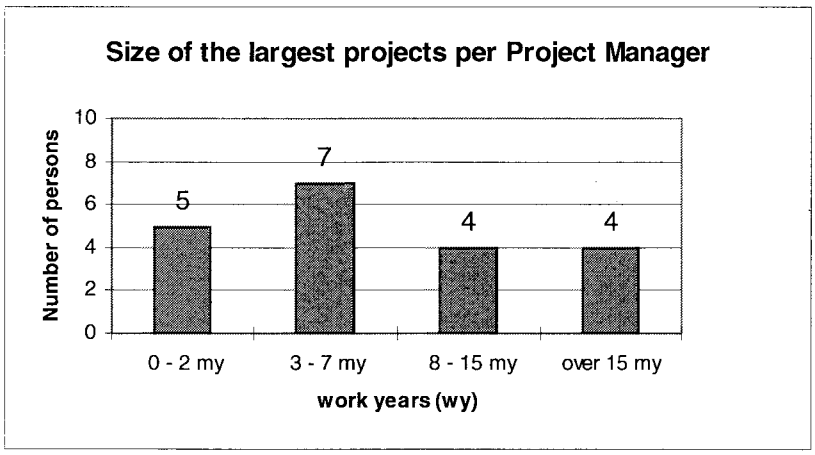



FIGURE 15. Largest projects.

For the majority of the project managers the size of the smallest project had been a couple of work months, whereas the largest had been several work years. Some of the project managers had managed really large projects by any standards.

### 4.1.3 Projects Examined

Ten managers analyzed one of their own projects from the risk management perspective. They were advised to select a recently completed software project, but there was no expectations for the success of the project, since either successful or failed projects were assumed to be suitable for the analysis. However, half of the projects were not finished by the time of the interviews. This is unfortunate, since possible risks of the final phases of the projects had not appeared and those risks were thus not included in the analyses. Nevertheless, three of these projects can be considered finished, since the produced systems were already in use or ready to be taken into use. The systems were not yet accepted by the customer and hence the projects were not finished officially. Two projects were uncompleted; one was in the coding phase and another in the testing phase.

The size of the projects varied from a couple of work-years to over thirty work-years. Two projects were very large, but the exact figures of their work load are not available. In four cases the actual implementation of the system was outsourced. Two projects included tailoring or fitting on-the-self systems to the organization and one project was

adding a new feature to an old tailored system. The projects included several fields: business and management applications, inventory control, ERP-systems, embedded systems, etc. All of them included programming, but the amount of it varied as part of the total work required.

Only two of the projects had been completed in time. Similarly, only two of the ongoing projects was on schedule by the time of the interview (in another one the schedule had, however, been redefined earlier during the project). The remaining projects exceeded their deadlines and – in most cases – their budgets. Below are some examples of how the project managers described their projects:[8]

*'The total workload was 25% over the estimated.'*
*'The personnel cost was 87% higher than expected.'*
*'It was planned to take half a year – it took three years.'*
*'The project was remarkably more expensive than expected.'*

The project managers evaluated the successfulness of their projects for example as follows:[9]

*'The project was a bit worse than our average projects.'.*
*'The output of the project has proved to be profitable, and hence the project is considered successful.'*
*'The project went quite well.'*
*'This project has been a real problem. It has been an unprofitable failure.'*
*'According to the feedback from the users they have been satisfied with the system. Therefore, I would say that the project has been successful.'*

Overall, the majority of the projects were considered successful despite the delays or exceeding the budget. However, arguments for 'success' varied being often based on subjective attitudes. In software houses the project profitability was usually the main

[8] The actual interviews were conducted in Finnish.
[9] The actual interviews were conducted in Finnish.

criteria for evaluating a project's success, whereas users' opinions were dominant in assessing the success of a whole project in the companies with their own system development.

### 4.1.4 The Organization of Risk Management

Project management and risk management are closely related. Both of these topics were discussed in interviews. Some features of the project management practices are presented in Table 11.

TABLE 11. Characteristics of project management.

| | Yes | Problems/ Occasionally | No |
|---|---|---|---|
| Instructions for project work | 7 | 2 | 1 |
| Project plans | 10 | (1) | - |
| Project management software | 6 | (3) | 4 |
| Audits and walkthroughs | 7 | 1 | 2 |
| Project evaluation (post mortem) | 4 | 2 | 4 |

Most of the companies had guidelines for project work, though two companies had problems with utilizing them (instructions were out-of-date, or they were not followed). All companies used project plans, but one of them admitted that the plans are often not realistic and had little to do with the actual project.

Audits and walkthroughs were regularly conducted practices in seven companies, while post-mortem audits were regularly carried out only in four companies. Four other companies reported that they did not use them, and two deployed them only on some occasions. The companies analyzing the projects systematically after they were finished had identified methods for estimating the project success. These methods include criteria like profitability of the project, faultlessness of the system, delivery on time, the accuracy of work effort estimation, and customer satisfaction.

Some sort of project management software was used in six companies. Three of them were familiar with project management software while the remaining had only recently begun using them.

Characteristics of risk management activities are given in Table 12. Risk analyses at the start of the project were used systematically in six companies, and occasionally (e.g. if the project is expensive or large) in four companies. In five companies the analysis had no fixed form, whereas the remaining companies always evaluated the probability and the effects of a risk. These two are the elements of Risk exposure (Risk Exposure = Probability * Loss), but only two companies calculated the actual value of risk exposure. The probability and loss were most often estimated numerically (e.g. 1 = low, 2 = medium, 3 = high). The effects of risks were in some companies classified by identifying the areas of negative consequences (exceeding budget or time limits, missing possible profits of the systems etc.). Risks were monitored systematically in five companies.

TABLE 12. Characteristics of risk management.

| | **Systematically** | **Occasionally** |
|---|---|---|
| Risk analysis | 6 | 4 |
| Probability and loss | 5 | (2) |
| Follow up | 5 | 5 |

The IS managers were asked to evaluate the risk management ability of their companies. The frequencies of the grades[10] are illustrated in Figure 16.[11]

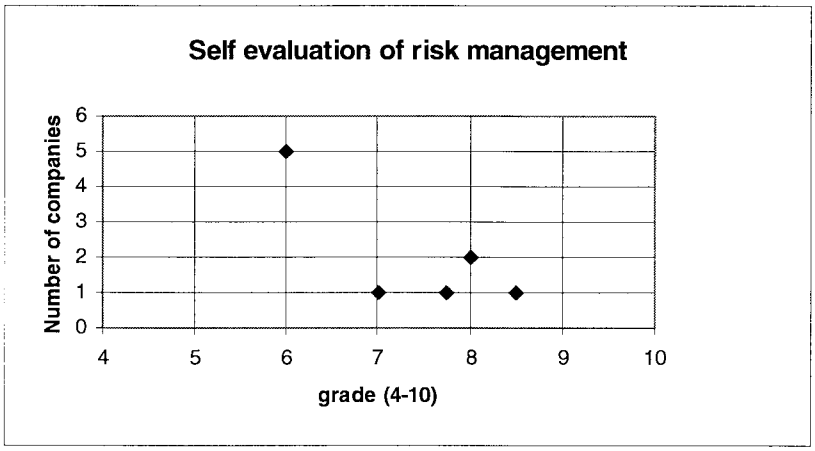



FIGURE 16. Self evaluation of risk management.

[10] The answers were expressed using the grade scale of the Finnish comprehensive schools: 10 and 9 = excellent; 8 = good; 7 = fair; 6 and 5 = adequate; and 4 = poor.

[11] One of the IS managers used a scale of one to five. She assessed the risk management ability as 3.5, which was interpreted as 8-(7.75) in the above scale. Another IS manager's estimation was 'bad', which corresponds to a numeric value of six (6).

This question was not submitted to the IS managers before the interviews. They had to extemporize. No systematic metrics was used, and the answers are thus based on IS managers' subjective intuitions.

## 4.2 Risks

Risks are analyzed in two different ways. First, we present the general topics of risks by introducing risk categories. These groups identify general problematic areas of software development. Next, we examine the individual risk items and present observation frequencies of each risk item.

### 4.2.1 Risk Categories

Besides the fourteen risk categories already identified in the Delphi study, the study helped identify two additional groups. The groups were named as *'potential risk exposure over the system life cycle'* and *'real-time performance shortfalls'*. Overall, we observed 16 groups of risk factors and had 276 identifications of risks items. Risk categories are presented below in descending order (Table 13).

TABLE 13. List of risk categories (modified from Schmidt *et al.* 1999, p. 39).

| # | Risk Category | Description | Observations |
|---|---|---|---|
| 1 | Requirements | Requirements: Inadequate or poor management of system requirements; poor validation of system requirements. | 41 |
| 2 | Sponsorship/Ownership | Mandate: Lack of mandate for the PM to execute the project plan. Lack of trust or poor relationships with the owners of the system. | 38 |
| 3 | Relationship Management | User Relationships: Lack of trust and inadequate user involvement. Unclear roles and expectations among users or other stakeholders. | 31 |
| 4 | Staffing | Staffing: Changes in personnel or staffing levels, unavailability of key personnel resources. | 25 |
| 5 | Technology | Technology: Inadequate understanding of the chosen technology. | 21 |
| 6 | Personnel | Skills: Inadequate personnel skills in development and process management. | 20 |
| 7 | Scope | System Scope: Unclear, changing or partial understanding of the system scope and mission. | 16 |

(cont.)

TABLE 13 (cont.)

| | | | |
|---|---|---|---|
| 8 | Scheduling | Resource control: Poor management of resource consumption and needs. Poor timing. | 14 |
| 9 | External Dependencies | Development environment: Poor management or control over dependencies with external agents. | 14 |
| 10 | Funding | Resource management: Too little or badly estimated resources for SD. | 13 |
| 11 | Project Management | Management: Poor or inefficient management strategy and execution. | 12 |
| 12 | Corporate Environment | Environment: Changes in the business or political environment or poor alignment of the system with the organizational culture. | 8 |
| 13 | Development Process | Process: Inappropriate or lacking process approach. | 7 |
| 14 | Potential Risk Exposure over the System Life Cycle | Failure to balance risks over the system life cycle. | 7 |
| 15 | Planning | Planning: No interest or inadequate skills to plan the project. | 5 |
| 16 | Real-time Performance Shortfalls | Utilization: Problems with the performance of the system. | 4 |

'Requirements' includes such risk items as lack of frozen requirements and misunderstanding the requirements. 'Sponsorship/ownership' corresponds to lack of top management commitment to the project, failure to gain user commitment, and conflicts between user departments. 'Relationship management' represents e.g. failures to manage end user expectations, lack of adequate user involvement, and lack of cooperation from the users.

The original categories are adopted from Schmidt *et al.* (1999). Risk items belonging to each group are introduced in Appendix A. Categories 14 and 16 were not included in the original list. They were added afterwards, since these items were observed by the project managers. Category 14, 'potential risk exposure over the system life cycle', denotes the need to observe risks related to all phases from the very beginning to the termination of the system and making trade-offs between them (e.g. quality vs. cost). For example, the consequences of poor practice in the early phases of system development might have a crucial impact on the maintenance. Category 16, 'real-time performance shortfalls', means facing performance problems during the use of the system.

### 4.2.2 Individual Risk Items

The number of observations concerning each individual risk item are summarized below. Altogether we got 42 different risk factors, of which fourteen were mentioned at least by ten project managers (Top 14 risk items). These factors are shown in Table 14. Column 'Former#' indicates the placing in the Delphi study (see Appendix B). Character '-' corresponds to the factor not being in the former list of the 11 risk items. All the items below belong to the Delphi list of 53 risk items. Columns 'List' and 'Proj.' refer to the two types of project manager's interviews; analyzing a list of risk items and analyzing a single project, respectively. The rest of the identified risks were mentioned by less than ten participants. A complete list of risk factors is presented in Appendix E.

TABLE 14. List of the most common risks.

| # | Former# | Risk Factor | List | Proj. | Total |
|---|---|---|---|---|---|
| 1 | (3) | Misunderstanding the requirements | 10 | 9 | 19 |
| 2 | (8) | Introduction of new technology | 10 | 8 | 18 |
| 3 | (6) | Lack of frozen requirements | 10 | 6 | 16 |
| 4 | (5) | Lack of required knowledge/skills in the project personnel | 9 | 6 | 15 |
| 5 | (9) | Failure to manage end user expectations | 10 | 5 | 15 |
| 6 | (2) | Failure to gain user commitment | 9 | 5 | 14 |
| 7 | (7) | Changing scope/objectives | 10 | 3 | 13 |
| 8 | (10) | Insufficient/inappropriate staffing | 9 | 4 | 13 |
| 9 | (1) | Lack of top management commitment to the project | 9 | 3 | 12 |
| 10 | (11) | Conflict between user departments | 9 | 3 | 12 |
| 11 | (-) | Bad estimation | 4 | 8 | 12 |
| 12 | (-) | Artificial deadlines | 4 | 8 | 12 |
| 13 | (4) | Lack of adequate user involvement | 10 | 1 | 11 |
| 14 | (-) | External dependencies not met | 3 | 7 | 10 |

As can be seen in Table 14, the project managers analyzing a list of risk items identified almost all the risks in the list, but mentioned other risks only occasionally despite the fact that they were encouraged to do so. Altogether they made 141 observations of risks and identified 28 risk items. Only two of these items were not mentioned by the other project managers analyzing a single project. They in turn made 135 observations of 40 risk items. Fourteen of them were not mentioned by project managers analyzing a risk item list. The risks identified by project managers analyzing a single project were arranged by the frequency of observations, and the results are given in Table 15. Column 'Top 14' refers to risk item's position in Table 14.

TABLE 15. Risk items ranked by project managers analyzing a single project.

| | Top 14 | Risk item | Observations |
|---|---|---|---|
| 1. | (1) | Misunderstanding the requirements | 9 |
| 2. | (2) | Introduction of new technology | 8 |
| 3. | (11) | Bad estimation | 8 |
| 4. | (12) | Artificial deadlines | 8 |
| 5. | (14) | External dependencies not met | 7 |
| 6. | (3) | Lack of frozen requirements | 6 |
| 7. | (4) | Lack of required knowledge/skills in the project personnel | 6 |
| 8. | (5) | Failure to manage end user expectations | 5 |
| 9. | (6) | Failure to gain user commitment | 5 |
| 10. | (16) | Potential risk exposure over the system life cycle. | 5 |

Table 15 includes all the risk factors observed by at least half of the project managers analyzing their own project. This list has one item outside the Top 14 list: potential risk exposure over the system life cycle. A corresponding list for the other project managers would consist of the 11 risk items included in the list they analyzed, since no other items were identified by at least half of them.

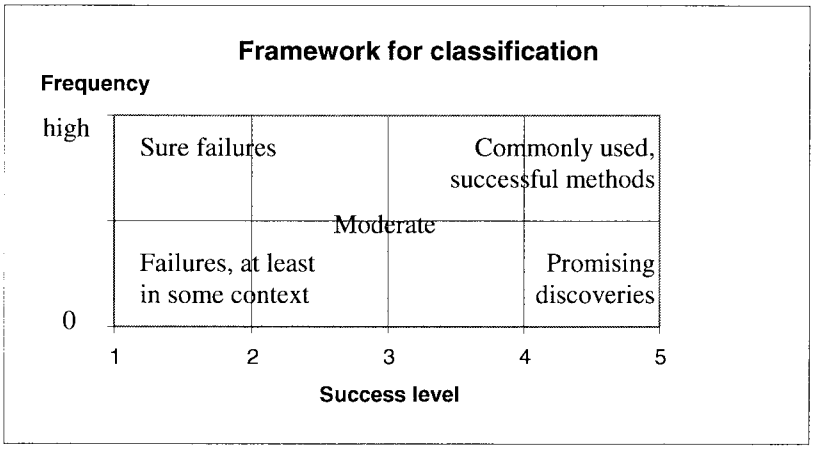
## 4.3 Risk Management Methods

In the analysis of the mitigation strategies we wanted to focus attention on the most important and generally identified problems. Therefore, the analysis was restricted to cover only the 14 risk factors above. The rest of the risks were mentioned solely by a few participants, and their mitigation methods were sparse. Furthermore, we did not separate the methods depending on in which type of interview (analysis of one project or analysis of the risk item list) they were mentioned. Data collection method was supposed to affect risk identification, but it was not assumed to influence which mitigation methods the interviewees explained having used.

The 'relative importance' was used to arrange the methods so that the most successful and frequently used methods were ranked highest. In addition to the list of risk management methods for each risk item, the results were displayed in two-dimensional graphs. A list of risk management methods for the Top 14 risk items and graphs are shown in Appendix F. When reading the Appendix, it should be noticed that groups with a smaller success level than three are considered to be failures.

The graphs were developed to visualize the generality and success level of each method. In order to help interpret the graphs, Table 16 outlines a classification framework for risk management methods. The dimensions used in the graph are success level and frequency. Frequency means the generality of the method *i.e.* how many project managers mentioned it. Methods in the upper right corner are widely used and successful, whereas methods in the upper-left corner – being widely used and unsuccessful – are sure failures. Interesting findings can be made of those methods, which only one project manager had identified, but which are classified as having a relatively high success rate. In the graph these methods are positioned in the lower right corner (frequency = 1 and success rate = 4 or 5). The power of those risk management methods is either highly dependent on the context, *i.e.* they can be applied only in specific cases, or they are useful methods which only a few have utilized so far. Methods placed in the lower-left corner are failures, at least in a particular context, since they have been mentioned by only a few participants. The methods situated in the middle of the graphs are moderately successful. The classification explained here is normative, and its objective is to give an impression of the relative importance of each method. Thus the distinctions between categories are not strict.

TABLE 16. Framework for classification of risk management methods.




The maximum value of frequency dimension is represented by the term 'high' in Table 16, because the actual frequencies varied depending on the risk item. The scale of success level, from one to five, is the same for every risk item.

The majority of identified risk management methods were evaluated as moderate, whereas unsuccessful methods were not mentioned by many participants. As a summary of the results, we present successful methods, promising discoveries, and failures. First, commonly used and successful methods are displayed in Table 17.

TABLE 17. Commonly used, successful methods.

| **Risk Factor** | **Risk Management Methods (commonly used, successful)** |
|---|---|
| Misunderstanding the requirements | • Use a disciplined development approach. |
| Introduction of new technology | • Avoid new technology.<br>• Allocate time for learning when estimating work efforts.<br>• Training and studying.<br>• Obtain knowledge outside the project.<br>• Assure that the technology is functional before choosing it. |
| Lack of frozen requirements | • A disciplined procedure to manage chance requests (change control). |
| Lack of required knowledge/skills in the project personnel | • Identify and obtain required skills in time.<br>• Obtaining help outside the project. |
| Failure to manage end user expectations | • User involvement in the project. |
| Failure to gain user commitment | • Key end-user involvement in the project. |
| Changing scope/objectives | • Keep projects small. |
| Insufficient/inappropriate staffing | • Proper planning and follow up.<br>• Hire and release people to work in the project. |
| Lack of top management commitment to the project | • Increase knowledge of IT among top management.<br>• Make projects important, large, or expensive. |
| Conflict between user departments | • Obtain agreement beforehand.<br>• Communicate and cooperate. |
| Bad estimation | • Tools for workload estimation.<br>• Frequent team meetings; follow up remaining workload. |
| Artificial deadlines | • Frequent team meetings; follow up remaining workload.<br>• Tools for workload estimation. |
| Lack of adequate user involvement | • Agreements on user involvement and workload beforehand.<br>• Management commitment.<br>• Release a user from routine work by reorganizing, or hiring new employees. |
| External dependencies not met | • Document all changes to old systems. |

'Promising discoveries' (Table 18) include methods observed by one project manager and having high success value (4 or 5). 'Moderate' methods having equally high success rate and higher frequency (2 or 3), are worth examining as well, since they are positioned between these two groups (promising discoveries and commonly used, successful methods).

TABLE 18. Promising discoveries.

| Risk Factor | Risk Management Methods (promising discoveries) |
|---|---|
| Misunderstanding the requirements | • Good domain knowledge (developers).<br>• Team organization.<br>• Abandon the project, develop a new contract, and a new start. |
| Introduction of new technology | • Use SWOT-analysis when choosing technology.<br>• Project manager's professional skills.<br>• Proper project planning. |
| Lack of frozen requirements | • Good domain knowledge.<br>• Write down also trivially evident tasks. |
| Lack of required knowledge/skills in the project personnel | • Management commitment.<br>• Remove unskilled personnel. |
| Failure to manage end user expectations | • Create realistic expectations in the selling phase.<br>• Changes are deferred for implementation.<br>• Prevent the use of the other systems. |
| Failure to gain user commitment | • Share responsibility with the customer and the supplier.<br>• Make the system critical.<br>• The new system resembles the former one. |
| Changing scope/objectives | • Communication between IT and management.<br>• Technical solutions. |
| Insufficient/inappropriate staffing | • Management commitment. |
| Lack of top management commitment to the project | • Change in the management style. |
| Conflict between user departments | • Have only few departments. |
| Bad estimation | • Team organization.<br>• The customer accepts extensive testing.<br>• Releasing a user from routine work. |
| Artificial deadlines | • The customer accepts extensive testing.<br>• Releasing a user from routine work. |
| Lack of adequate user involvement | • - |
| External dependencies not met | • Allocate time for unexpected delays when estimating work efforts. |

'Lack of adequate user involvement' was the only risk factor which had no risk management methods classified as 'promising discoveries'.

Table 19 includes all methods having a relative success less than 3. Only one of them, making rough definitions in order to manage changing requirements, was categorised as a sure failure. The rest were considered failures at least in some context. Only those risk items which have unsuccessful methods are included in Table 19.

TABLE 19. Failures.

| Risk Factor | Risk Management Methods (failures) |
|---|---|
| Misunderstanding the requirements | • Preparing for capacity problems. |
| Lack of frozen requirements | • Use 'rough' and limited specification.[12] |
| Failure to manage end user expectations | • Off-the-self products are not customized.<br>• Training in the very early stage of the project.<br>• New system does not help user's own work. |
| Failure to gain user commitment | • Letting someone else other than the end user derive the requirements.<br>• Install the system as soon as possible to customer. |
| Insufficient/inappropriate staffing | • Include the max total price in the contract. |
| Lack of top management commitment to the project | • Leave the risk to the customer's responsibility.<br>• Several ongoing large projects. |
| Lack of adequate user involvement | • Participation in seminars and training is made obligatory |

These three tables are presented to highlight interesting findings. A complete list of risk management methods is shown in Appendix F.

## 4.4 Summary of Results

The participating companies can be considered to form a representative sample of Finnish companies. They vary in size, location, and the area of business operation. Overall, the participating companies are bigger than the majority of Finnish companies, but very small firms (*i.e.* less than 20 employees) are presumed not to have invested in their own system development and particularly software project risk management.

Although the companies represent Finnish business relatively successfully, do the project managers represent the majority of their colleagues? Not necessarily the majority, but the most experienced ones. We may assume that risk management is hardly arranged better than what the interviewed project managers explained in the survey, since the participants were particularly experienced. The majority of them had worked several years in information system development and had managed a considerable large number of projects. The projects they had been involved in covered all the main areas of the use of IT.

[12] This was the only method classified as "Sure failure". It was identified by four project managers and its average success rate was 2.25.

A few of the analysed projects, however, were not properly qualified. Some projects were not finished, and all the possible risks of remaining project phases were not necessarily known by the time of the interviews. Changing a project at the beginning of the interview would have caused delays, and the timetables were sometimes too tight to allow the change.

The risks identified in the interviews are presented in two ways: in categories and individually. This method enables both browsing through the most problematic areas of software development and studying the individual risk items in isolation.

The analysis of the risk mitigation method was restricted to cover only the most important risks. Methods were arranged in the order of the indicator value of the 'relative importance'. This value is a product of the average success level given to the method and the number of persons identified using the method. The data collection method did not support identifying unsuccessful countermeasures, since only a few failures were mentioned.

A description of the characteristics of project management, and especially risk management, offer possibilities for benchmarking. Learning from other companies encourages improvements in own practices.

# 5 Discussion

The discussion section is divided into three main areas: assessing the identified risk items, analyzing suggested risk management methods and their success levels, and discussing the general risk management practices of the participating companies.

## 5.1 Assessment of the Identified Risk Items

The importance of risk management is becoming better understood by project managers. The concept of risk was familiar to almost every participating project manager, whereas less than 25% of project managers used risk management in the beginning of the 1990's (Ropponen 1993). This might be due to the depression, financial losses caused by large failure projects, the overall development of project management activity, and the choice of the interviewed companies.

The identified risk items are analyzed in several ways. First, we discuss the items accepted into the Top 14 list (the new list formed in this research), assess the relative order of the items and the observed changes compared to previous lists. We also explain the differences caused by two types of interviews. The risk items have been analyzed using the categories introduced by Cule *et al.* (1999). This classification is presented in the following section. Finally, we evaluate the validity of the sample and observe limitations of the results.

### 5.1.1 Top 14 Risk Factors

Risk items reported by project managers were quite similar with the former studies. When compared with the Top Ten list of software risk items by Boehm (1989, p. 35, see also Table 2 of this thesis), only one risk item was excluded from the new list. This item was 'developing the wrong user interface', and it was neither included in the Delphi list (Keil *et al.* 1998, p. 78). Comparing the identified risk categories with the 53-item list of the Delphi study two additional groups emerged: 'real-time performance shortfalls' and

'potential risk exposure over the system life cycle'. Real-time performance has been mentioned earlier in the software risk management literature by Boehm (1989). Potential risk exposure over the system life cycle has not been reported earlier. It was mainly identified by project managers analyzing one project, and it denotes the need to observe risks related to all phases from the very beginning to the termination of the system and making trade-offs between them (e.g. quality vs. cost). The recognition of this risk is supposed to be caused by the economic depression in the early 1990's in Finland. The tendency in the beginning of the decade was that only necessary software projects were launched. The competition for customers was tough. On the other hand, the customers had gained experience in purchasing software implementations. Being all too familiar with the problems of the past projects, they demanded better project management practices. The suppliers sought to maintain long-term relationships with customers, and they could not afford to leave the customer alone with maintenance problems. All of these reasons explain the appearance of the risk item focusing on risk exposure over total life cycle.

The order of the risk items differed somehow from the previous Delphi list. All the 11 items in the Delphi-list were within the top 14 risk factors of this research. The interviewed project managers found 'requirements' and 'sponsorship/ownership' to be the most problematic areas. Of the individual risks, 'misunderstanding the requirements' and 'introduction of new technology' were ranked first. They were both in a higher position than in the Delphi study.

In the classification introduced in the Delphi study (Keil *et al.* 1998, p. 80, see also Figure 5), both of these items fell into quadrant 2, 'scope and requirements'. Risks in this quadrant were considered to have high relative importance and the project manager has high level of control over them. They form the core of the projects' outcome. Relevant questions connected to that quadrant are: What is the objective of the project? Has the scope of the project been understood correctly? Are the system requirements valid? Keil *et al.* (1998, p. 80) summarized the questions the project manager may ask himself as *'Do I know what I am building and how this might change over time?'*. Placing these risks at the top of the list is supported by their positioning in the quadrant

2 of the Delphi study. The fact that these risks were the most often noted indicates that the project managers have identified these risks and they are considered to be important and within their control. No other patterns arouse when classifying the risks in the categories introduced in the Delphi study.

The other items improving their ranking were: 'lack of frozen requirements', 'lack of adequate knowledge/skill in the project personnel', 'failure to manage end user expectations', 'insufficient/inappropriate staffing', and 'conflict between user departments'. Overall, the project managers regarded people as risk items, not the technology. As one of them said: *'You can always cope with technology, but people are the problem.'* Another continued that *'People skills and negotiation skills are more important than technical skills.'*[13]

Risks concerning funding, scheduling, and external dependencies ('bad estimation', 'artificial deadlines', and 'external dependencies not met') were included in the Delphi study, but not among the Top 11 items. In this research, these three items were most often mentioned by the project managers analyzing one project. These items were the main difference between risk lists formed according to the two types of the interviews. One reason for estimation problems and artificial deadlines might be that the analyzed projects have had excessively tight schedules. Another explanation might be that as those project managers analyzed brand new projects and the other project managers reported all the projects during their career, the importance of these risks has increased.

'Lack of adequate user involvement' and 'changing scope/objectives' are examples of risks which were not problematic in the analyzed projects. These risk items were identified by all project managers who examined the risk list, but hardly any project manager analyzing one project (only one of them mentioned lack of user involvement and three observed change of scope or objectives). Is user involvement really such a big risk as it has been? Our study suggests that user involvement has became quite a standard practice. The managers reported problems with end users, especially with

[13] The actual interviews were conducted in Finnish.

managing their expectations and releasing them from their routine work, but compared to Delphi list our findings indicate positive progress in improving end-user involvement.

Similarly, in the sample projects there were neither risks with conflicting user departments, nor lack of top management commitment. All these risk items have either maintained their position with only minor changes, or they had a considerable lower position when compared to the Delphi list. The former top of the list, 'lack of top management commitment', was now at the 9th place. This is probably a result of several factors. Possible reasons for this are that the managers might be more sophisticated in IT, the recession has focused the development activities on subjects supporting the actual business, and the overall importance of IT has increased, or due to the choice of the companies.

### 5.1.2 Classification of the Risk Items

Cule *et al.* (1999) have created a classification of risk items from the project managers' point of view. This classification is based on four sources of risks: *self, task, customer,* and *environment* (see also Figure 6). We applied this classification to the list of individual risk items by separating the Top 14 risks from the remaining risks. The results are given in Table 20, and the coding is provided in Appendix G.

TABLE 20. Risk catorization.

| Dimension | Category | Top 14 | Others | Total/ Category | Total/ Dimension |
|---|---|---|---|---|---|
| Inside | Self | 1 | 6 | 7 | 21 |
| | Task | 4 | 10 | 14 | |
| Outside | Client | 7 | 5 | 12 | 21 |
| | Environment | 2 | 7 | 9 | |
| Total | All | 14 | 28 | 42 | |

The total number of risks, 42 items, is divided evenly between inside risks (*self* and *task*) and outside risks (*client* and *environment*). The majority of the Top 14 risks, however, are *outside risks*, and, are particularly concerned with *client.*

The category of *self* had the smallest number of risks. The participating project managers were the most experienced, which may explain the small number of risks caused by the characteristics of the project managers themselves. It is also common that managers do not recognize risks related to their own professional skills (people are poor to understand their own weaknesses). The only risk item falling in this category among the Top 14 risks was the most important one: 'misunderstanding the requirements'.

The majority of the risks were classified into the category *task*. However, most of these risks were not among the Top 14. This might be a consequence of experienced project managers' ability to control and direct the project's work. So, despite the large number of task related risk they were not considered the most critical ones, or they were under the active control of the project manager.

*Client* category had 50% of the Top 14 risks. The name of the category refers to the client(s) of the project manager, including customers, top managers, end users etc. These risks are difficult, since project managers' ability to control them is weak. This category corresponds quite well to the quadrant 1, Customer mandate, in the Delphi study (Keil *et al.* 1998, p. 80).

The remaining risks belong to the *environment* category. Among the Top 14 were two environmental risks: 'changing scope/objectives' and 'external dependencies not met'. These were not many, and a project manager can only prepare for them. They can be neither controlled, nor influenced by the project manager (Cule *et al.* 1999, p. 9).

## 5.2 Risk Management Methods and Their Success

Mitigation methods used by project managers were reported to be relatively successful. However, they were not very sophisticated. The use of common sense was usually mentioned to be more effective than complex and refined theories. Still, utilizing some of these theories might help avoid disasters and keep better control over projects. Overall, risk management is not a silver bullet, but it offers a mechanism to increase risk awareness and to better prepare for threats. Several project managers noticed that

identifying the risk is not a problem, but management actions were the real challenge. The project managers' attitude toward risk management was positive: doubtful, but curious. Some project managers were eager to learn new techniques to control their projects.

### 5.2.1 Highlights

When compared to Alter and Ginzberg's (1978) list of risk-reducing strategies, the use of the evolutionary or modular approach was not included in our list. Neither was hiding complexity, permitting voluntary use, relying on diffusion and exposure, or tailoring the system to people's capabilities identified. Alter and Ginzberg commented on the method of 'insist on mandatory use' as being *'a strategy of last resort, violating the basic tenents of participative model'* (Alter and Ginzberg 1978, p. 29). In our study a comparable method, 'participation in seminars and training is made obligatory', was classified as a failure. Obligating the use of a system or participation in a project can thus be considered an ineffective, or even harmful, action. The rest of Alter and Ginzberg's risk-reducing strategies were included in our study as successful, or moderately successful methods.

Risk management techniques introduced by Boehm (1991, p. 35) included several formal analyses (e.g. organization analysis, mission analysis, off-nominal performance analysis, quality-factor analysis, cost-benefit analysis). Our subjects did not mention having used these methods. The interviewed project managers identified 'training in the very early stage of the project' as a failure, whereas Boehm suggests 'early user's manuals' as a mitigation method for risk of developing wrong functions and properties. These items are not necessarily conflicting, since user's manuals can be considered as a means to describe the resulting system in a concrete way to the end user (and project manager).

Most risk management methods observed in our study have not been reported earlier in the literature. The majority of these 'methods' are common sense management actions, and some of the interviewed project management did not even notice having followed a

method; they just described their ordinary work. Earlier literature, Boehm (1991) in particular, has focused on formal tools and techniques, which are still unfamiliar among many practitioners.

As a summary of our findings, we discuss a number of aspects of risk management observed in the study. First, the participating project managers emphasized planning and preparing. Projects should be planned and followed up properly, agreements on user participation should be made before a project starts, and a company should take care of obtaining required skills in advance. Besides planning and following up individual projects, a company should arrange the timing of different projects carefully.

Secondly, the use of a disciplined development approach was reported to eliminate several risk items. The requirement determination methods used in projects should be easily understood by end users. Users participating in the project team should be innovative and capable of understanding system development. The size of the projects should be small. Projects should be maintained according to schedule, and documentation should not be neglected.

Human aspects or 'people skills' were noticed to be often more important than technical abilities. Communication and cooperation are useful strategies in managing human relations. The advantages of team organization were admittedly difficult to achieve, since a spirit of togetherness cannot be created by management fiat. Furthermore, people do not act as 'rational machines' and thus requirement determination should be allowed to take time. Things often come to mind suddenly – after meetings. This manner of processing things should be taken into account when estimating schedules.

In addition, active interaction between management and IT department is essential in order to make IT benefit the business. Top management was observed to be now better aware of IT issues, and underestimating IT departments was declining. Management interest was increased as well due to the introduction of business-oriented software projects.

The selling phase of a new system has been found to be critical. This phase has previously been, or is also today, a separate part conducted by a professional seller. If the project team, or project manager, is responsible for the whole life cycle of a project, including the selling phase, several risks can be avoided or mitigated. The selling phase is also essential in managing users' expectations by creating a realistic image of the resulting system.

Furthermore, the most general method for managing technological change was avoiding new technology as long as possible. The IT area is developing rapidly, and new tools are usually considered unreliable and unnecessary. This strategy is naturally not shared by companies who base their business on technological innovations and leading edge solutions.

Finally, one of the identified methods had a sense of humor: conflicts between user departments can be avoided by decreasing the number of departments (e.g. by joining or by reorganizing).

### 5.2.2 Discrepancies

The material also included discrepancies. The project manager's opinions of the effectiveness of the mitigation strategies differed in some cases. These differences are understandable, since the organizational context and area of operations varied.

In 'introduction of new technology', 'leaving time for learning when estimating work efforts' was considered to be successful by seven project managers. However, two project managers assumed it to be a failure. Their argument was that the schedule should consciously be set as tight as possible and should be accepted by the developers. This motivates the team to work hard and to strive toward the deadline. If there is extra time in the schedule, it all will be consumed, they said.

In 'lack of required knowledge/skills in the project personnel' three project managers mentioned a method of 'staffing with capable users and releasing them to the project'.

Two of them were in favor of it, and one was against. The cause was that arranging the routine work is not necessarily easy. Hiring someone else to do the daily work of the end user participating in a project might demand a lot of additional work from the end user. How can you find a capable substitute to replace a professional end user? Some companies had prepared for these situations with job rotation, team organization etc., but otherwise this is undoubtedly a poor strategy.

'Keeping the management informed of project's situation' was one mitigation strategy for 'lack of top management commitment to the project'. According to one of the project managers, this is not a successful method, unless you do it properly. Sending e-mail or minutes is not enough. The content of this statement is clear: management should be kept informed at 'arms-length' distance, and face-to-face meeting is the most effective way of communicating such knowledge.

With regard to 'bad estimation' and 'artificial deadlines' we observed a similar discrepancy. 'Proper project planning' was categorised both as a success and as a failure. One manager assessing it as a failure argued that *'Proper planning includes defining the requirements before fixing the schedule. This is always impossible.'*[14] Actually, he did not deny the positive effect of proper planning, but he found it difficult to implement.

'Lack of adequate user involvement' included three methods which divided opinions. All these, however, were noticed by the same person. 'Top management commitment' is valued mainly as a successful strategy, but he noted it to be insufficient alone. *'A manager's commitment is for nothing, if he does not also arrange opportunities for participation.'*[15] Another strategy, 'releasing user from routine work by reorganizing or hiring new employees' was argued with the same reasons as in 'lack of required knowledge/skills in the project personnel'. 'Inviting and persuading' the customers or end-users to participate in the project was also one of the discrepancies. According to one of the project managers, this is insufficient. The argument was the same for top management commitment: the lack of guaranteed opportunities to participate.

---

[14] The actual interview was conducted in Finnish.

'Documenting all changes to old systems' in 'external dependencies not met' was observed to be difficult. Since the strategy itself is useful, one project manager had a strong distrust of the suppliers either in documenting the changes, or maintaining the old systems unaltered.

### 5.2.3 Inhibiting and Compensating Strategies

The mitigation methods were classified into compensating or inhibiting strategies (Alter and Ginzberg 1978, p. 28, see Appendix H). As mentioned earlier, compensating strategies are used to correct a previous error or problem after it has occurred, and inhibiting strategies are invoked to avoid a risk beforehand (ibid.). From all the 164 mitigation methods, 99 were inhibiting and 65 compensating. Correspondingly, from the 27 'promising discoveries' (*i.e.* low frequency and high success level), 17 were inhibiting and 10 were compensating. In both cases the portions are equal: about 60% are inhibiting strategies and about 40% are compensating. If the same analysis is made, but only the first ranked mitigation methods are included, the relation changes. Out of these 14 methods 11 are inhibiting and 3 compensating. The shares are 79% and 21%, respectively. The figures are shown in Table 21 as well.

TABLE 21. Classification of Inhibiting/Compensating.

| | **Inhibiting** | **Compensating** | **Total** |
|---|---|---|---|
| All | 99 (60%) | 65 (40%) | 164 |
| 'Promising discoveries' | 17 (63%) | 10 (37%) | 27 |
| First ranked methods | 11 (79%) | 3 (21%) | 14 |

These calculations give us an impression of inhibiting strategies being slightly more general, and particularly more effective. Naturally, avoiding a risk is preferred to correcting its consequences. The inhibiting strategies were evaluated as more successful (mean 3.6) than compensating strategies (mean 3.3).

---

[15] The actual interview was conducted in Finnish.

### 5.2.4 Socio-technical Features of Mitigation Methods

We coded both the risk items and the resolution techniques according to Leavitt's (1964) socio-technical model for organizational change. The summary of the mapping is provided in Table 22. A complete list of categorization is given in Appendix H. The majority of the risks were related to *structure*, while *technology* was the reason for only one of the Top 14 risks. The methods as well were mainly concerned with the *structure*.

TABLE 22. Socio-technical categories.

| Category | Risk items | Methods |
|---|---|---|
| Task | 4 | 28 |
| Structure | 6 | 68 |
| Technology | 1 | 27 |
| Actor | 3 | 41 |
| Total | 14 | 164 |

None of the four approaches analyzed by Lyytinen *et al.* (1998) focused on structural risk items. They were either balanced or focused on the three other categories. However, *structure* was one of the main categories of the risk resolution techniques of these four approaches. Their focus on resolution techniques was either balanced or twofold, but always included the *structure*. Compared to the analysis by Lyytinen *et al.* (1998), none of the four approaches coincides with our findings. However, Ropponen (1999) has performed a corresponding categorization of a very large project, and his findings are similar to ours. The majority of the risk items of that particular project were structure-related (Ropponen 1999, p. 165), and the number of technology-related risk was surprisingly low. Furthermore, the vast majority of management inventions concerned structural issues (Ropponen 1999, p. 176). *Technology*, again, was the least observed focus of interventions out of the four categories. Overall, our results are in line with Ropponen's findings. As both of these studies are empirical, we are encouraged to draw a conclusion that structure-related risks and resolution techniques tend to be the most commonly used in practice.

We continued the analysis by exploring which are the best methods suggested for each category of risk, *i.e.* are the *task* risks compensated by task-related methods. Table 23 presents the results.

TABLE 23. Socio-technical categories of risk management methods.

| Category of risk item | Method: Task | Method: Structure | Method: Technology | Method: Actor | Total |
|---|---|---|---|---|---|
| Task | 11 | 24 | 9 | 10 | 54 |
| Structure | 8 | 27 | 9 | 17 | 61 |
| Technology | 2 | 4 | 4 | 3 | 13 |
| Actor | 7 | 13 | 5 | 11 | 36 |
| Total | 28 | 68 | 27 | 41 | 164 |

The risks related to *task* are mainly controlled by changing the *structure*. The *structure* risks were also managed by actions concerning the *structure*. For technological risks the resolution techniques were balanced, the main methods being technological or structural. Risk of *actors* were managed with a method related to *actors* and *structure*. Overall, the *structure* was the most popular category for methods concerning all types of risks. In other words, structural changes are used for all types of risks, not only for structure-related ones. This phenomenon is understandable, hence – in addition to scrubbing requirements – structural issues are the only ones a project manager can affect in a short-range perspective. *Actors* and *technology* are often beyond the control of a project manager. To summarize, projects should be planned well in advance. Particular attention should be paid to requirements, personnel, and technology and adjusting the structure to meet these conditions.

## 5.3 Risk Management Practices

Basic risk management methods, such as risk analysis and checklists, were used in almost every participating company – especially in IT companies. Overall, the software companies had a better grip over risk management. Most of them had a disciplined development method which alone is handy in mitigating several risks. Risk management is basically about managing uncertainty and complexity (see also Mathiassen & Stage

1990), and all activities reducing them mitigate risks as well. Planning the projects properly and following them actively are simple, but effective, means of project management. A difference between software houses and other companies was in garnering organizational knowledge. IT companies systematically audited finished projects and they had a clear objective to learn from their past experiences. In most of the other companies this was considered 'politically incorrect'. One comment was that *'This is a management problem. We cannot talk about projects without talking about the people involved. The mistakes and criticism are taken personally.'*[16] Boehm and Ross (1989) have advised separating people from the problem, but it is still difficult in practice.

Another difference between software houses and the remaining companies was in monitoring risks during the project. IT companies had usually arranged a systematic way of following the development of risks, e.g. including risk management as a standard issue on the agenda of the steering group[17] meetings. Risk analysis at the beginning of a project was also generally more precise and concrete. The use of project management software did not make any difference in the ability to manage risks.

When evaluating their ability in risk management, IT companies were slightly better than the others. In our opinion[18], their evaluations were also more realistic in relation to other companies, while some companies largely overestimated their capabilities. The company having the highest grade indeed earned it, since they had worked on risk management successfully over the last six years and had considerable experience in it.

Reasons for IT companies' better performance might due to the fact that software development is closer to the core business in software houses. They have developed systematically their skills on project management as well as on risk management. Software houses were good at collecting information about finished projects in order to increase their organizational knowledge and to avoid disasters. They had also experienced employees whose work task included project management improvement.

---

[16] The actual interview was conducted in Finnish.
[17] Other names like control group and advisory group were used as well.
[18] All these evaluations are subjective and they can be possibly biased.

The project managers had in general a good education, and the companies trained them constantly. They were also able to increase their ability to manage risks. As a whole, software houses were better prepared for risks. However, the difference between software houses and other companies was not very significant.

Two of the IS managers held opposite opinions. According to their experience, the IT companies have not proved themselves capable of managing risks properly. As one them said: *'They [the IT companies] do not assess risks. Actually, I am disappointed since I know that one of our suppliers has a quality handbook, but I have never seen any projects where they had used it.'*[19]

Overall, IS managers thought their risk management to be well under control, whereas their project managers did not agree. A clear example of this is when the IS manager stated that the company had quality assurance guidelines, which were followed. One of the project managers of the same company argued that *'No project has ever been documented following the guidelines.'*[20] The IS manager was obviously not aware of the situation, or maybe he just wanted to give a better image of his company by embellishing facts they know. In other aspects, IS managers were well aware of the reality of the project management.

A common comment from the project managers was that the risks are identified relatively easily, but orchestrating the management actions is difficult. Weaknesses were tolerated, although action should have been taken immediately. Besides recognizing the shortcomings in the management, practically all participating companies identified improvements they are going to accomplish in the near future. These improvements include for example use of check lists, developing risk categorization, project evaluations, and improvements in risk analysis.

In general, the lack of disciplined development practices increases risks related to estimating work-effort, fixing the schedule, and managing requirement changes. These

---

[19] The actual interviews were conducted in Finnish.
[20] The actual interviews were conducted in Finnish.

risks, however, materialize differently in each project. Overall risk reduction techniques have to be adjusted for a particular situation.

## 5.4 Comments and limitations

Our research results have been communicated to several software professionals. The identified risk lists has been found by these practitioners to correspond their experiences. Some risks are not relevant in all companies; for example, a company investing huge sums of money in new technology does not consider 'introduction of new technology' as a risk, but as an opportunity. In the same vein, the estimation of the importance of a risk items depends on several factors: time perspective, special system requirements (e.g. reliability is critical), the organizational context, the evaluator's role (a supplier or a customer), etc. In addition, we asked a risk management trainer to evaluate our list. A representative of Software Technology Transfer Finland Oy, Pekka Forselius, contended that their experience supports our results (personal communication, June 3rd, 1999). Mr Forselius has trained risk management as part of software project management since January 1997. The trainees have ranked risk items (23 risk items from the Finnish panel of the Delphi study) as an exercise. After about 50 courses and over 450 trainees, he stated that the his results were astonishingly similar to our findings. All these comments permit us to argue that the sample is representative and the results reflect general risk factors.

There are some limitations in the derived risk list. All analyzed projects were not finished, and some of them were not 'genuine' software projects. There is also a possibility that these projects did not include risks the project managers are often facing in other projects. This risk could be avoided by using two types of interviewing methods, where half of the project managers analyzed the 11 item list of risk factors.

Despite promising findings concerning risk mitigation methods the limitations of the study have to be recognized. The numerical values connected to each mitigation strategy were given solely by the researcher, not by the participant. The values are based on the 'reading' of the interview transcripts. The values might include bias which results from

our inability to read the opinions of the participants, or misinterpret their statements. The probability of such bias, however, has been reduced by basing coding on a careful deliberation and standard classification instructions. Using two people to form the categorization and compare the results was unfortunately not possible, since there was only one researcher in the study. In the same vein, the classification of risk items and mitigation methods of socio-technical categories might be incorrect. All these classifications, however, have been checked carefully several times with the mentor of the study.

The sample is small, and no statistical generalizations can be drawn from the findings. However, the analysis of the risk resolution techniques covers a total number of 164 methods, which allows the use of quantitative methods in future.

## 5.5 Summary

The risk items identified by project managers were as expected. When compared to the previous checklists there were only minor changes in the content and in the relative order of the risk item list.

The risk items observed most often were 'misunderstanding the requirements' and 'introduction of new technology'. Using the classification method of Keil *et al.* (1998, p. 80), both of these items fell into quadrant 2, *scope and requirements*. Risks in this quadrant are considered to have high relative importance, and the project manager has a high level of control over them. Positioning at the top of the list implies that project managers have recognized the risk as items they can and must control.

Mapping the risks according to Cule *et al.* (1999) reveals the principal area of risks being *client*, *i.e.* client(s) of the project manager. This classification is based on a behavioural model, and risks are viewed from the project managers' perspective. These results seem warranted, since the participants were experienced. Relationship management can be more difficult for technology oriented IT professionals.

The risk resolution techniques noted by the project managers were simple and relatively successful. No complex theories were utilized. The majority of the mitigation methods relate to changes in the *structure*, *i.e.* organizational and institutional aspects of system development, like arrangements of communication, authority, and work flow. Examples of structural actions are changes in project schedule, or personnel. The most effective methods were inhibiting strategies. Compensating strategies were almost equally common, but they were not among the 'best' mitigation strategies.

The software houses differed from other companies when we analyzed the risk management practices. Overall, software houses had better competence in risk management. However, all the participating companies had room for improvement, and they had noticed their limitations in risk management.

The theory and the practice of software project risk management have not yet coalesced. Even experienced project managers are missing up-to-date training on risk management. Yet, previous failures and recent economic depression have taught them to take risk management seriously. Even though risk management has been discussed in the IS literature for over 20 years, the practitioners' interest has been raised only recently.

# 6 Conclusion

In this thesis we examined risk management in software development. The principal focus has been on exploring the use of risk mitigation strategies and their success rates. The research subjects covered ten companies. The research problem was: *What methods, if any, are used to control and mitigate risk items in software development projects in Finland today, and how successful are these methods?* In order to answer these questions we had to clarify which risk factors were considered most important, and how the risk management activity was organized in Finnish enterprises.

We began with exploring what has been reported previously in the risk management field. An overview of the risk management literature formed a basis for our later analysis. We compared several definitions of the concept of risk, and introduced typical risks in information system development. The examination of the suggested risk management methods concentrated on software development project risks. As a result of the literature review, we noticed a lack of empirical research on the use of risk management methods.

Next, we presented the methodology of the research. The study was a qualitative field study in that the empirical data was collected using semi-structured interviews. Ten companies and altogether 31 persons participated the study. The interviews were classified into three categories: an IS manager, a project manager analyzing his own project, and a project manager going through a list of risk items. Two different types of project managers' interviews were used in order to increase the reliability and the validity of the study, and as it is a part of an international research we had to apply similar data collection method as in other participating countries. The background of the company, general information of risk management methods, and self evaluation of the risk management ability were the topics of the IS managers' interviews. Half of the project managers analyzed one of their earlier (or current) projects from the risk management perspective ('project' interviews). The other half of them went down a 11-item list of risk factors and recalled if they had encountered them in any project during their career ('list' interviews). The aim of the project managers' interviews was to

collect experiences of risk items, mitigation methods, and their success rates. The research has a practical focus: what risks really have occurred, how they have been managed, and what has been the result of the management interventions.

We continued with analyzing the interview data. The research involves three kinds of results: a list of the most important risk items, a list of identified mitigation methods for these items, and general information on risk management practices.

Risk items reported by project managers were similar to former studies. When compared with the earlier Delphi study, two additional groups were recognized: 'real-time performance shortfalls' and 'potential risk exposure over the system life cycle'. Real-time performance has been mentioned earlier in the software risk management literature, but potential risk exposure over the system life cycle is new. The appearance of this risk is noteworthy, since it denotes changes in attitudes. Software development is now considered more holistically. A development project is seen as just a part of a long lasting cooperation. The tendency is that the suppliers will extend their responsibility to cover all phases of the system life cycle, and shortsighted solutions in the development phase are beginning to give way to more foresighted decisions balancing risks over the system's whole life cycle. This change is encouraging.

Risk mitigation methods used by project managers were reported to be relatively successful, but not very sophisticated. The use of common sense was mentioned to be more effective than complex and refined theories. The mitigation strategies for each risk items were reasonable and practical. They have been used successfully, though some of them were not considered effective. Furthermore, the impact of these strategies is often related to a particular context of the project. The participants mentioned several mitigation methods, which can be considered as 'potential discoveries'. Those methods were estimated to be highly successful, but by a single participant only. These methods were worth examining, though some of them cannot be adapted into different organizational context.

The problems of software risk management are in risk monitoring and execution. Identification, analysis, and prioritization, *i.e.* risk assessment, were considered to be better understood. These actions are now standard practice in several companies. However, several project managers lacked knowledge of risk management methods, and they were not familiar with risk management tools and techniques. As practitioners have noticed shortcomings in their ability to manage software project risk, they are willing to increase their education. Risk management is popular today and it has the potential to became a common practice. This is an opportunity that has to be taken seriously. Otherwise the whole idea will be forgotten as new trends appear.

Risk management should be included in every day project management as one ingrained element. If it requires considerable additional work and the benefits remain unclear, the busy project managers drop it. Basic risk management methods as risk analysis and checklists were used in nearly every company – especially in IT companies. Obtaining at least these methods as permanent procedures within a disciplined project management practice would increase the risk management ability of several companies.

Overall, the software companies had better competence in risk management. They were better prepared, they had enough resources, and they appreciated risk management more than other companies. The closer information technology is to the strategic core of the company, the better risk management is organized. The difference between software houses and other companies, however, was not considerable, and bold conclusions cannot be made based on this research.

Because the sample included only ten companies and twenty project managers, the results cannot be statistically generalized. Obtaining a generalizable understanding of risk management demands more companies and interviews. The scope of this research was limited to cover only ten companies due to limited resource. However, the participating companies can be regarded as a representative sample of Finnish enterprises. The findings have strong support from previous surveys, which allows us to argue that the interviewed managers and analyzed projects illustrate the general state of the art in risk management.

In future, also other project team members, like end users and maintainers, should be included in the interviews to obtain a richer understanding of risk management practices. This would support the idea of regarding risks as associated with a stakeholder. One way of examining the success of the mitigation methods is to arrange a field study. Special attention would be directed to risk management in a few projects, say from one to three projects. The analysis should cover the total life cycle of the project and a check point after one year's use of the system. Such an analysis has already been conducted by Ropponen (1999). Another alternative is to train an experienced project manager to act as an internal consult on risk issues. He would collect risk-related data from all (or several) ongoing projects of the company, guide and help the project managers in risk management, and finally, form guidelines of risk management for his own organization. Useful findings would be done as well by analyzing the relations of risk items. Risks are interrelated, and eliminating some of them will remove some others (e.g. 'Management commitment' is a countermeasure for personnel and staffing risks). Moreover, a quantitative statistical analysis would be arranged to evaluate our findings. The resulting lists should be sent to several practitioners, who would assess the lists compared to their experience. Finally, our results will be included in an international comparison of risk management methods and their success rates.

## References

Alter S. & Ginzberg M. (1978), Managing Uncertainty in MIS Implementation. Sloan Management Review, Vol. 20, No. 1, Fall, 23-31.

Barki H., Rivard S., & Talbot J. (1993), Toward an Assessment of Software Development Risk. Journal of Management Information Systems, Vol. 10, No 2, 203-225.

Benbasat I. (1984), An Analysis of Research Methodologies. In McFarlan F. W. (ed.), The Information Systems Research Challenge. HBS, Boston.

Boehm B. W. (1988), A Spiral Model of Software Development and Enhancement. IEEE Computer, May, 61-72.

Boehm B. W. (1989), Software Risk Management Tutorial. IEEE Computer Society Press.

Boehm B. W. (1991), Software Risk Management: Principles and Practices. IEEE Software, Jan, 32-41.

Boehm B. W. & Ross R. (1989), Theory-W Software Project Management: Principles and Examples. IEEE Transactions on Software Engineering, Vol. 15, No. 7, July, 902-916.

Carr M. J. (1997), Risk Management May Not Be for Everyone. Counterpoint, IEEE Software, May/June, 21, 24.

Charette R. N. (1989), Software Engineering Risk Analysis and Management. Intertext Publications McGraw-Hill Book Company.

Collins Cobuild (1995), English Dictionary. HarperCollins Publishers, London.

Conrow E. H. & Shishido P. S. (1997), Implementing Risk Management on Software Intensive Projects. IEEE Software, May/June, 83-89.

Crockford N. (1980), An Introduction to Risk Management, Woodhead-Faulkner, Cambridge, England.

Cule P., Schmidt R., Lyytinen K., & Keil M. (1999), Heading Off Failures: Strategies for Preempting IS Project Risk. Working paper.

Davis G. B. (1982), Strategies for Information Requirements Determination. IBM Systems Journal, vol. 21, No. 1, 4-30.

Davis G. B. & Olson M. H. (1985), Management Information Systems - Conceptual Foundations, Structure and Development. McGraw-Hill Book Company.

Duncan W. R. (ed.) (1996), A Guide to the Project Management Body of Knowledge. Project Management Institute, Upper Darby, USA.

Fairley R. (1994), Risk Management for Software Projects. IEEE Software, May, Vol. 11, No. 3, 57-67.

Galliers R. (ed) (1992), Information Systems Research. Issues, Methods and Practical Guidelines. Alfred Waller Ltd, Oxfordshire, Great Britain.

Genuchten M. van (1991), Why is Software Late? An Empirical Study of Reasons For Delay in Software Development. IEEE Transactions on Software Engineering, Vol. 17, No. 6, June, 582-590.

Gönfors M. (1982), Kvalitatiiviset kenttätyömenetelmät. WSOY, Porvoo.

Hirsjärvi S., Remes P., & Sajavaara P. (1997), Tutki ja kirjoita. Kirjayhtymä Oy, Helsinki.

Jenkins A. M. (1985), Research Methodologies and MIS Research. In Mumford E. *et al.*, Research Mehods in Information Systems. North-Holland, Amsterdam.

Karolak D. W. (1996), Software Engineering Risk Management. IEEE Computer Society Press, Los Alamitos, California.

Keil M., Cule P., Lyytinen K., & Schmidt R. (1998), A Framework for Identifying Software Project Risks. Communications of the ACM, November, Vol. 41, No. 11, 76-83.

Kontio J. (1997), The Riskit Method for Software Risk Management, version 1.00. CS-TR-3782. Computer Science Technical Reports. University of Maryland. College Park, MD.

Kontio J., Englund H., & Basili V.R. (1996), Experiences from an Exploratory Case Study with a Software Risk Management Method. CS-TR-3705, UMIACS-TR-96-75. University of Maryland Technical Reports. College Park, Maryland.

Kontio J., Getto G., & Landes D. (1998), Experiences in Improving Risk Management Processes Using the Concepts of the Riskit Method. Proceedings of the Proceedings of the Sixth International Symposium on the Foundations of Software Engineering (FSE-6).

Leavitt H. J. (1964), Applied organization change in industry: structural, technical and human approaches in Leavitt H. J. (ed), New Perspectives in Organizational Research. John Wiley, Chichester, 55- 71.

Lister T. (1997), Risk Management is Project Management for Adults. POINT, IEEE Software, May/June, 20, 22.

Lyytinen K. & Hirschheim R. (1987), Information systems failures - a survey and classification of the empirical literature. Oxford Surveys in Information Technology, Oxford University Press, England, Vol. 4, 257-309.

Lyytinen K., Mathiassen L., & Ropponen J. (1996), A Framework for Software Risk Management. Journal of Information Technology, Vol. 11, No. 4, 275-285.

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, September, 233-255.

MacCrimmon K. R. & Wehrung D. A. (1986), Taking Risks, Free Press, New York.

Mathiassen L. & Stage J. (1990), Complexity and Uncertainty in Software Design. Procseedings of CompEuro, 1-8.

McFarlan W. (1982), Portfolio Approach to Information Systems. Journal of Systems Management, January, 12-19.

Moynihan T. (1997), How Experienced Project Managers Assess Risk. IEEE Software, Vol. 14, No. 3: May/June, 35-41.

Mäkelä K. (ed) (1987), Kvalitatiivisen aineiston analyysi ja tulkinta. Gaudeamus, Helsinki.

Neo B. S. & Leong K. S. (1994), Managing Risks in Information Technology projects: A Case Study of TradeNet. Journal of Information Technology Management, Vol. 15, No. 3, May/June, 35-41.

Pelin R. (1990), Projektin suunnittelun ja valvonnan menetelmät. 2. uudistettu painos. Gummerus Kirjapaino Oy, Jyväskylä.

Ropponen J. (1993), Risk Management in Information System development. Licentiate thesis, Computer Science Reports, Technical Reports TR-3, University of Jyväskylä, Finland.

Ropponen J. (1999), Software Risk Management - Foundations, Principles and Empirical Findings. Jyväskylä Studies in Computing 1, University of Jyväskylä, Finland.

Ropponen J. & Lyytinen K. (1997), Can Software Risk Management Improve System Development: An Exploratory Study. European Journal of Information Systems, 6, 41-50.

Ropponen J. & Lyytinen K. (1999), Components of Software Development Risk: How to address them? A project manager survey. IEEE Transactions on Software Engineering, 1999, Vol. 25, Forthcoming issue.

Schmidt R., Lyytinen K., Keil M., & Cule P. (1998), Identifying Software Project Risks: An International Delphi Study. Submitted for publication in Journal of Management Information.

Silverman D. (ed) (1997), Qualitative Research. Theory, method and practise. Sage, London.

Williams R. C., Walker J. A., & Dorofee A. J. (1997), Putting Risk Management into Practice. IEEE Software, Vol. 14, Number 3, May/June, 75-82.

Yin R. K. (1984), Case Study Research: Design and Methods. Sage, Beverly Hills, CA.

Yin R. K. (1989), Research Desing Issues in Using the Case Study Method to Study Management Information Systems. In Cash J. I. & Nunamaker J. F. jr (Eds.), The Information Systems Research Challenge: Qualitative Research Methods. Vol. 1, Harvard Business School Research Colloquium, Boston, 1-6.

Appendix A: A complete list of risk items (Delphi 53-list)

**Table 3. Full List of Risk Factors**

| | |
|---|---|
| **1.** | **Corporate Environment** |
| 1.1 | *A climate of change in the business and organizational environment that creates instability in the project.* |
| 1.2 | *Mismatch between company culture and required business process changes needed for new system.* A mismatch between the corporate culture and the changes required by the new system. |
| 1.3 | *Projects That Are Intended to Fail*: Projects started for political reasons which carry no clear business value but serve to divert the organization's focus from actual needed change. Such projects are under-funded, not supported and are not intended to succeed. Projects have no business value and are used as diversionary tactics to avoid facing the real change needs. |
| 1.4 | *Unstable Corporate Environment:* Competitive pressures radically alter user requirements, sometimes making the entire project obsolete. |
| 1.5 | *Change in Ownership or Senior Management:* New owners and/or managers set new business direction that causes mismatch between corporate needs and project objectives. |
| **2.** | **Sponsorship/Ownership** |
| 2.1 | *Lack of Top Management Commitment to the Project.* This includes oversight by executives and visibility of their commitment, committing required resources, changing policies as needed. |
| 2.2 | *Lack of client responsibility, ownership, and buy-in of the project and its delivered system(s).* |
| 2.3 | *Failure to gain user commitment*: Laying blame for "lack of client responsibility" on the project leader rather than on the users. |
| 2.4 | *Conflict Between User Departments*: Serious differences in project goals, deliverables, design, etc., calls into question concept of shared ownership. |
| 2.5 | *Failure to get project plan approval from all parties* |
| **3.** | **Relationship Management** |
| 3.1 | *Failure to Manage End User Expectations*: Expectations determine the actual success or failure of a project. Expectations mismatched with deliverable - too high or too low - cause problems. Expectations must be correctly identified and constantly reinforced in order to avoid failure. |
| 3.2 | *Lack of Adequate User Involvement*: Functional users must actively participate in the project team and commit to their deliverables and responsibilities. User time must be dedicated to the goals of the project. |
| 3.3 | *Lack of Cooperation from Users*: Users refuse to provide requirements and/or refuse to do acceptance testing. |
| 3.4 | *Failure to Identify All Stakeholders*: Tunnel vision leads project management to ignore some key stakeholders in the project, affecting requirements definition, implementation, etc. |
| 3.5 | *Growing Sophistication of Users Leads to Higher Expectations*: Users are more knowledgeable, have seen sophisticated applications, apply previous observations to existing project. |
| 3.6 | *Managing Multiple Relationships with Stakeholders*: Some "clients" are also "partners" in producing deliverables in other projects. Leads to confusion of roles and responsibilities. |
| 3.7 | *Lack of appropriate experience of the user representatives*: Users assigned who lack necessary knowledge of the application or the organization |
| **4.** | **Project Management** |
| 4.1 | *Not Managing Change Properly:* Each project needs a process to manage change so that scope and budget are controlled. Scope creep is a function of ineffective change management and of not clearly identifying what equals success. |
| 4.2 | *Lack of Effective Project Management Skills*: Project teams are formed and the project manager does not have the power or skills to succeed. Project administration must be properly addressed. |
| 4.3 | *Lack of Effective Project Management Methodology*: The team employs no change control, no project planning or other necessary skills or processes. |
| 4.4 | *Improper Definition of Roles and Responsibilities*: Members of the project team and the organization are unclear as to their roles and responsibilities. This includes outsourcers and consultants. |
| 4.5 | *Poor or Non-Existent Control*: No sign-offs, no project tracking methodology, unaware of overall project status, "lost in the woods". |
| 4.6 | *Poor Risk Management*: Countering the wrong risks. |
| 4.7 | *Choosing the Wrong Development Strategy*: e.g. waterfall, prototyping, etc. |
| **5.** | **Scope** |
| 5.1 | *Unclear/Misunderstood Scope/Objectives.* It is impossible to pin down the real scope or objectives due to differences or fuzziness in the user community. |
| 5.2 | *Changing Scope/Objectives*: Business changes or reorganizes part way through the project. |
| 5.3 | Scope Creep: Not thoroughly defining the scope of the new system and the requirements before starting, consequently not understanding the true work effort, skill sets and technology required to complete the project. |
| 5.4 | *Project Not Based on Sound Business Case*: Users and developers ignore business requirements, develop system for sake of technology. |
| 5.5 | *Number of organizational units involved*: increased number of lines of communication and conflict potential expands the scope of the system. |

Appendix A: A complete list of risk items (Delphi 53-list)

| | |
|---|---|
| **6.** | **Requirements** |
| 6.1 | *Lack of Frozen Requirements.* Because the needs of the users change, the requirements change. Consequently the system will never be moved into production because none of the requirements are ever completed. Alternatively, freezing a subset of the functionality and delivering allows for the completion of the system and update releases as required. |
| 6.2 | *Misunderstanding the Requirements.* Not thoroughly defining the requirements of the new system before starting, consequently not understanding the true work effort, skill sets and technology required to complete the project. |
| 6.3 | *New and/or Unfamiliar Subject Matter for Both Users and Developers:* Lack of domain knowledge leads to poor requirements definition. |
| **7.** | **Funding** |
| 7.1 | *Under Funding of Development*: Setting the budget for a development effort before the scope and requirements are defined or without regard to them (i.e., picking a number out of the air). |
| 7.2 | *Under Funding of Maintenance*: Support for products in the maintenance phase. If the customer is unprepared or does not budget for this, the project can be judged a failure even if successful in all other aspects. |
| 7.3 | *Bad Estimation*: Lack of effective tools or structured techniques to properly estimate scope of work. Unrealistic cost estimates cause illogical or sub-optimal planning, strategy, and decisions. |
| 7.4 | *"All or Nothing":* Requires budgeting entire project at the outset, leading to under funding in later years of project. |
| **8.** | **Scheduling** |
| 8.1 | *Artificial Deadlines.* Presence of unrealistic deadlines or functionality expectations in given time period. - 'crash projects' in which test time or training time is reduced – using something other than work effort required to determine when the new system should move into production. |
| 8.2 | *"Preemption" of Project by Higher Priority Project*: Management unable to resolve conflicting schedule demands. |
| **9.** | **Development Process** |
| 9.1 | *Lack of Effective Development Process/Methodology:* Leading to quality problems - Documentation, Software and Testing—poor estimating -- insufficient time for up-front work, e.g., design—little flexibility for change—insufficient testing. |
| 9.2 | *Trying New Development Method/Technology During Important Project* |
| **10.** | **Personnel** |
| 10.1 | *Lack of Required Knowledge/Skills in the Project Personnel*: e.g., technology, business knowledge and experience. |
| 10.2 | *Lack of "People Skills" in Project Leadership*: PM tries to "manage" schedules, technology, requirements, etc., ignoring that management is dealing with people on the team. |
| 10.3 | *Poor Team Relationships*: Strains existing in the team due to such things as burnout or conflicting egos and attitudes. |
| **11** | **Staffing** |
| 11.1 | *Insufficient/Inappropriate Staffing*: Not enough people or people with wrong skills/insufficient skills assigned to project, regardless of availability. |
| 11.2 | *Staffing Volatility*: At some point in the project, losing the key project manager, analysts or technicians (especially in new technology). |
| 11.3 | *Excessive Use of Outside Consultants*: Can lead to a conflict of interest, e.g., billable hours vs. budget, or resulting in the internal staff not having significant involvement |
| 11.4 | *Lack of Available Skilled Personnel*: People with the right skills are not available when you need them. |
| **12.** | **Technology** |
| 12.1 | *Introduction of New Technology:* Using new, or 'bleeding edge', technology that has not been used successfully at other companies, or major technological shift occurs during the project. |
| 12.2 | *Stability of Technical Architecture* – Has to be done before comparable applications. |
| **13.** | **External Dependencies** |
| 13.1 | *External Dependencies Not Met*: The project's consultants or vendors do not deliver, go out of business, or are unclear as to their roles and responsibilities. |
| 13.2 | *Multi-Vendor Projects Complicate Dependencies*: Integration of packages from multiple vendors hampered by incompatibilities and lack of cooperation between vendors. |
| 13.3 | *Lack of Control Over Consultants, Vendors, and Sub-contractors*: Schedule or quality problems beyond control of project manager. No legal recourse due to poor contract specification. |
| **14.** | **Planning** |
| 14.1 | *No Planning or Inadequate Planning*: Attitude that planning is unimportant or impractical. |

Note: Items are grouped by category. Shaded items represent risk factors not observed in earlier lists (i.e., Boehm 1989, Barki, et al. 1993).

## Appendix B: A list of the most important risk items (Delphi 11-list)

List of Risk Factors

1. Lack of top management commitment to the project

   This includes oversight by executives and visibility of their commitment, committing required resources, changing policies as needed.

2. Failure to gain user commitment

   Laying blame for "lack of client responsibility" on the project leader rather than on the users.

3. Misunderstanding the requirements

   Not thoroughly defining the requirements of the new system before starting, consequently not understanding the true work effort, skill sets and technology required to complete the project.

4. Lack of adequate user involvement

   Functional users must actively participate in the project team and commit to their deliverables and responsibilities. User time must be dedicated to the goals of the project.

5. Lack of required knowledge/skills in the project personnel

   e.g. technology, business knowledge, and experience

6. Lack of frozen requirements

   Because the needs of the users change, the requirements change. Consequently the system will never be moved into production because none of the requirements are ever completed. Alternatively, freezing a subset of the functionality and delivering allows for the completion of the system and update releases as required.

7. Changing scope/objectives

   Business changes or reorganizes part way through the project.

8. Introduction of new technology

   Using new, or 'bleeding edge', technology that has been used successfully at other companies, or major technological shift occurs during the project.

9. Failure to manage end user expectations

   Expectations determine the actual success or failure of a project. Expectations mismatched with deliverable – too high or too low - cause problems. Expectations must be correctly identified and constantly reinforced in order to avoid failure.

10. Insufficient/inappropriate staffing

    Not enough people or people with wrong skills/insufficient skills assigned to project, regardless of availability.

11. Conflict between user departments

    Serious differences in project goals, deliverables, design, etc., calls into question concept of shared ownership.

Appendix C: A letter introducing the study

Kalle Lyytinen
Jyväskylän yliopisto
Tietojenkäsittelytieteiden laitos
PL 35, 40351 JYVÄSKYLÄ
puh. (014) 603 025
e-mail: kalle@jytko.jyu.fi

KIRJE 13.8.1998

**Arvoisa vastaanottaja,**

Jyväskylän yliopistossa on käynnissä tutkimus **tietojärjestelmäprojektien riskien hallinnasta**. Tutkimuksen tarkoituksena on selvittää, miten riskejä hallitaan ja kuinka toimivia erilaiset riskienhallinnan menetelmät ovat.

Aineisto tutkimusta varten kerätään **haastattelemalla kokeneita** tietojärjestelmäprojekteihin osallistuneita henkilöitä. Kaikki haastattelutieto käsitellään luottamuksellisesti, jolloin vastaajien henkilöllisyys ja organisaatio jäävät pelkästään tutkijoiden tietoon. Osallistuminen tutkimukseen tarjoaa oivan mahdollisuuden **arvioida yrityksen omaa riskienhallintatapaa** ja saada käyttöönsä tiivistettyä **tietoa alan viimeisestä kehityksestä**. Kaikille tutkimukseen osallistuneille lähetään lyhyt **yhteenveto** haastattelujen tuloksista ja riskienhallinnan "best practices" lista.

Tutkimustyön tulokset tullaan julkaisemaan tietojärjestelmäprojektien riskienhallintaa ja onnistumista koskevassa Pro Gradu -työssä. Työ sisältää myös katsauksen aiheeseen liittyvään kirjallisuuteen ja aiempiin tutkimuksiin. Työ on osa kansainvälistä riskienhallinnan tutkimushanketta. Vastaava kysely toteutetaan parhaillaan myös USA:ssa ja Hong Kongissa.

**Haastattelupyyntö**

Pyydämme Teitä ystävällisesti käyttämään hetken aikaa tieteellisen tutkimuksen ja käytännön työn yhteensovittamiseen osallistumalla haastatteluun. Kustakin yrityksestä haastattelemme **tietohallinnosta vastaavaa henkilöä** ja **kahta kokenutta projektipäällikköä**. Haastattelut kestävät 1-2 tuntia ja ne toteutetaan syys- ja lokakuun aikana.

Toivomme Teidän ilmoittavan **osallistumisestanne** tutkimukseen joko Kalle Lyytiselle (yhteystiedot yllä) tai Mari Honkoselle (puh. 014-602 549, e-mail: mari@kanto.jyu.fi). Haastatteluaikojen sopimiseksi otamme myöhemmin yhteyttä.

Odotamme vastauksianne **elokuun loppuun mennessä**.

Ystävällisesti,

Kalle Lyytinen
professori

Mari Honkonen
tutkija

Appendix D: Instructions for preparing for the interviews

**Tietojärjestelmäprojektien riskienhallinnan tutkimus 3.9.1998**

Jyväskylän yliopistossa on käynnissä tutkimus tietojärjestelmäprojektien riskien hallinnasta. Tutkimuksen tarkoituksena on selvittää, miten riskejä hallitaan ja kuinka toimivia erilaiset riskienhallinnan menetelmät ovat.

Tutkimuksen aineisto kerätään haastattelemalla kokeneita tietojärjestelmäprojekteihin osallistuneita henkilöitä. Kaikki haastattelutieto käsitellään luottamuksellisesti, jolloin vastaajien henkilöllisyys ja organisaatio jäävät pelkästään tutkijoiden tietoon. Kaikille tutkimukseen osallistuneille lähetään lyhyt yhteenveto haastattelujen tuloksista ja riskienhallinnan "best practices" lista.

Tutkimustyön tulokset tullaan julkaisemaan tietojärjestelmäprojektien riskienhallintaa ja onnistumista koskevassa Pro Gradu -työssä. Työ sisältää myös katsauksen aiheeseen liittyvään kirjallisuuteen ja aiempiin tutkimuksiin. Työ on osa kansainvälistä riskienhallinnan tutkimushanketta. Vastaava kysely toteutetaan parhaillaan myös USA:ssa ja Hong Kongissa.

**Haastatteluihin valmistautuminen**

Jokaisessa yrityksessä tulemme haastattelemaan kolmea henkilöä ja kukin näistä haastatteluista on luonteeltaan erilainen. Tietohallintojohtajan kanssa käymme läpi yleisiä tietoja yrityksestä ja sen tavoista hallita riskejä, ja projektipäälliköiden kanssa tarkastelemme yksityiskohtaisemmin erilaisia riskejä ja niiden poistamiseksi käytettyjä menetelmiä. Projektipäälliköiden haastattelut poikkeavat toisistaan siten, että toisen päällikön kanssa käsitellään yksi tietty projekti ja analysoidaan siinä esiintulleita riskejä, kun taas toisen kanssa käydään läpi luettelo riskeistä ja keskustellaan niiden ilmenemisestä eri projekteissa.

**Pyydämme teitä ystävällisesti valmistautumaan seuraavasti:**

***Tietohallintojohtaja***

Tässä haastattelussa selvitämme yrityksen taustatietoja ja yleistä riskienhallintamenettelyä. Tulemme kysymään tietoja mm. yrityksen ja sen tietohallinto-osaston koosta (mk, henkilöä), organisaatiorakenteesta, käytettävistä työkaluista ja -menetelmistä, henkilöstön koulutuksesta ja toteutettujen projektien koosta ja toimialasta. Lisäksi keskustelemme yrityksessä käytettävistä riskienhallinnan ohjausmenetelmistä, yrityskulttuurista ja järjestelmäkehityksen yleisistä ongelmakohdista.

***Projektipäällikkö, käsiteltävänä toteutunut projekti***

Haastattelun tavoitteena on analysoida yksi projekti riskienhallinan kannalta. Projektipäällikköä pyydetään valitsemaan etukäteen jokin projekti käsiteltäväksi. Projektin olisi hyvä olla hiljattain päättynyt, käsittelyn helpottamiseksi projektipäällikkö

voi varata mukaansa projektin dokumentaatiota. Haastattelussa pyydämme projektipäällikköä käymään yksityiskohtaisesti läpi kyseisen projektin eri vaiheet ja palauttamaan mieleensä esiintulleita ongelmia ja riskejä sekä niiden ratkaisutapoja.

***Projektipäällikkö, luettelo riskeistä***

Projektipäällikköä pyydetään käymään läpi luettelo riskitekijöistä ja kertomaan omia kokemuksiaan vuosien varrelta kustakin tekijästä. Olemme kiinnostuneita kuulemaan onko projektipäällikkö kohdannut työssään kyseisiä riskejä, millaisissa projekteissa ne ovat ilmenneet, miten niitä on käsitelty ja onko menettely ollut onnistunut.

Luettelon riskitekijöistä lähetämme haastateltavalle etukäteen.

**Kaikkia haastateltavia koskevia tietoja**

Kaikilta haastateltavilta kysymme lisäksi heidän omaa koulutustaustaansa ja projektityökokemustaan. Vaikka haastattelut on luokiteltu erilaisten otsikkojen alle, niin missään haastattelussa keskustelua ei ole kuitenkaan tiukasti rajattu juuri kyseiseen aiheeseen, vaan voitte vapaasti jakaa kokemuksianne riskienhallinnasta.

Nauhoitamme haastattelut, mikäli haastateltavilla ei ole mitään sitä vastaan. Nauhat kirjoitetaan puhtaaksi ja teksti lähetetään haastateltavalle tarkistettavaksi. Tällä menettelyllä pienennetään väärinkäsityksistä, puutteellisista muistiinpanoista tai haastattelijan virheellisistä muistikuvista johtuvia riskiä. Nauhat säilytetään lukkojen takana ja ne tuhotaan tekstin tarkistuksen jälkeen. Kaikki tutkimuksen aineisto kerätään tarkistetuista teksteistä, ei suoraan nauhoista.

Haastattelijana toimii tutkija Mari Honkonen. Professori Kalle Lyytinen osallistuu mahdollisuuksien mukaan tietohallintojohtajien haastatteluihin.

**Yhteystiedot**

Professori Kalle Lyytinen
Email: kalle@ jytko.jyu.fi

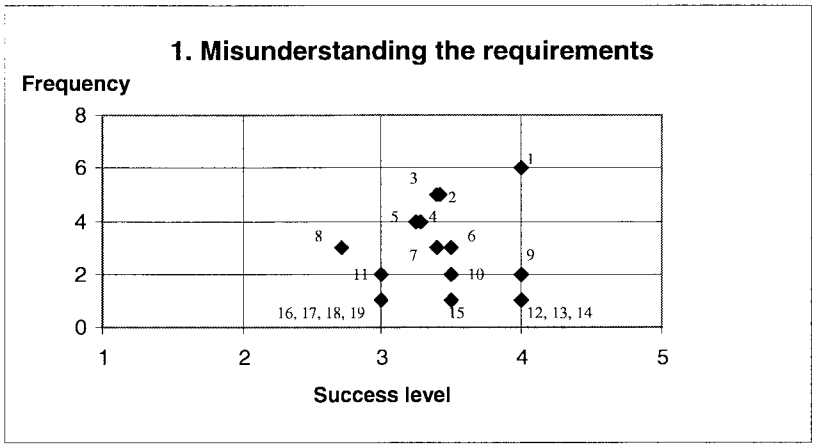
Jyväskylän yliopisto
Tietojenkäsittelytieteiden laitos
PL 35
40351 JYVÄSKYLÄ
puh. (014) 603 025
fax (014) 603 011

Tutkija Mari Honkonen
Jyväskylän yliopisto
Tietojenkäsittelytieteiden laitos
PL 35 (Kolmikulma)
40351 JYVÄSKYLÄ
Email: mari@kanto.jyu.fi
puh. (014) 602 549
fax (014) 602 544

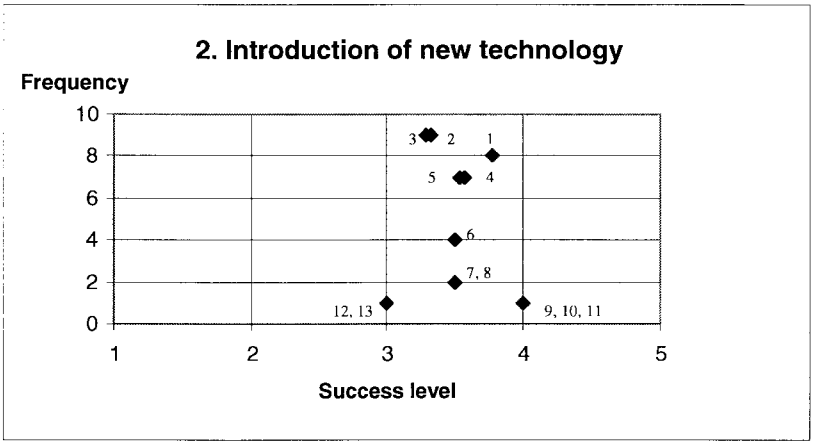
Appendix E: Identified risk items

| | ***Identified Risk Items*** | Observations |
|---|---|---|
| 1 | Misunderstanding the requirements | 19 |
| 2 | Introduction of new technology | 18 |
| 3 | Lack of frozen requirements | 16 |
| 4 | Lack of required knowledge/skills in the project personnel | 15 |
| 5 | Failure to manage end user expectations | 15 |
| 6 | Failure to gain user commitment | 14 |
| 7 | Changing scope/objectives | 13 |
| 8 | Insufficient/inappropriate staffing | 13 |
| 9 | Lack of Top Management Commitment to the project | 12 |
| 10 | Conflict between user departments | 12 |
| 11 | Bad estimation | 12 |
| 12 | Artificial deadlines | 12 |
| 13 | Lack of adequate user involvement | 11 |
| 14 | External dependencies not met | 10 |
| 15 | Lack of effective project management methodology | 8 |
| 16 | Potential risk exposure over the system life cycle (New!) | 7 |
| 17 | New and/or unfamiliar subject matter for both users and developers. | 6 |
| 18 | Lack of effective development process/methodology | 6 |
| 19 | Staffing volatility | 5 |
| 20 | No planning or inadequate planning | 5 |
| 21 | Lack of cooperation from the users | 4 |
| 22 | Lack of available skilled personnel | 4 |
| 23 | Real-time performance shortfalls | 4 |
| 24 | Mismatch between company culture and required business process changes needed for new system. | 3 |
| 25 | Lack of "people skills" in project leadership | 3 |
| 26 | Excessive use of outside consultants | 3 |
| 27 | Stability of technical architecture | 3 |
| 28 | Multi-vendor projects complicate dependencies | 3 |
| 29 | A climate of change in the business and organizational environment that creates instability in the project. | 2 |
| 30 | Change in ownership or senior management | 2 |
| 31 | Lack of effective project management skills | 2 |
| 32 | Number of organizational units involved | 2 |
| 33 | "Preemption" of project by higher priority project | 2 |
| 34 | Poor team relationships | 2 |
| 35 | Unstable corporate environment | 1 |
| 36 | Growing sophistication of users leads to higher expectations. | 1 |
| 37 | Improper definition of roles and responsibilities | 1 |
| 38 | Poor risk management | 1 |
| 39 | Scope creep | 1 |
| 40 | Under funding of development | 1 |
| 41 | Trying new development method/technology during important project. | 1 |
| 42 | Lack of control over consultants, vendors, and sub-contractors. | 1 |
| | | 276 |

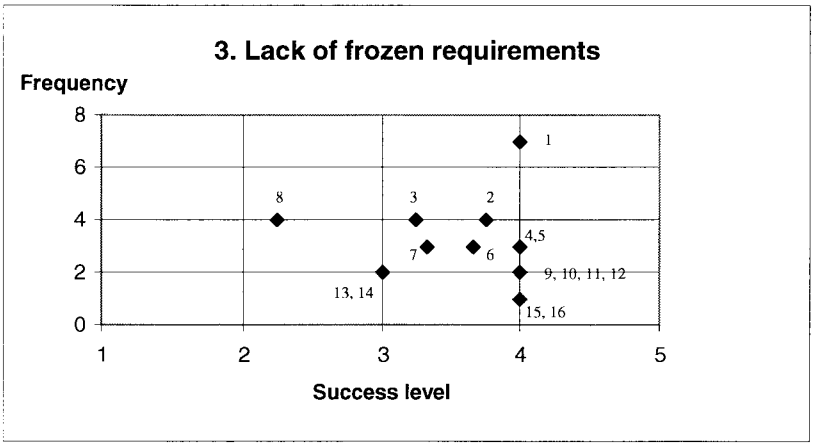
| 1 | Misunderstanding the requirements | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *Use a disciplined development approach.* | 4 | 6 | 24 |
| 2 | *Audit the tender; walkthroughs.* | 3,43 | 5 | 17,14 |
| 3 | *Review the requirements carefully with users.* | 3,4 | 5 | 17 |
| 4 | *Visualize the system and process using pilots, prototypes, etc.* | 3,29 | 4 | 13,14 |
| 5 | *Test the validity of requirements.* | 3,25 | 4 | 13 |
| 6 | *Careful and thorough investigation of the subject domain and requirements.* | 3,5 | 3 | 10,50 |
| 7 | *Capable users.* | 3,4 | 3 | 10,20 |
| 8 | *Preparing for capacity problems.* | 2,71 | 3 | 8,14 |
| 9 | *Keep the system simple; reduce functionality.* | 4 | 2 | 8 |
| 10 | *Increased experience in specifying requirements.* | 3,5 | 2 | 7 |
| 11 | *Requirements are defined by sole person (project manager).* | 3 | 2 | 6 |
| 12 | *Good domain knowledge (developers).* | 4 | 1 | 4 |
| 13 | *Team organization.* | 4 | 1 | 4 |
| 14 | *Abandon the project, develop a new contract, and new start.* | 4 | 1 | 4 |
| 15 | *Tools for workload estimation.* | 3,5 | 1 | 3,50 |
| 16 | *Preparing for an unsuccessful project.* | 3 | 1 | 3 |
| 17 | *Change management/control.* | 3 | 1 | 3 |
| 18 | *Postpone the installation.* | 3 | 1 | 3 |
| 19 | *Create realistic expectations in selling/marketing phase.* | 3 | 1 | 3 |



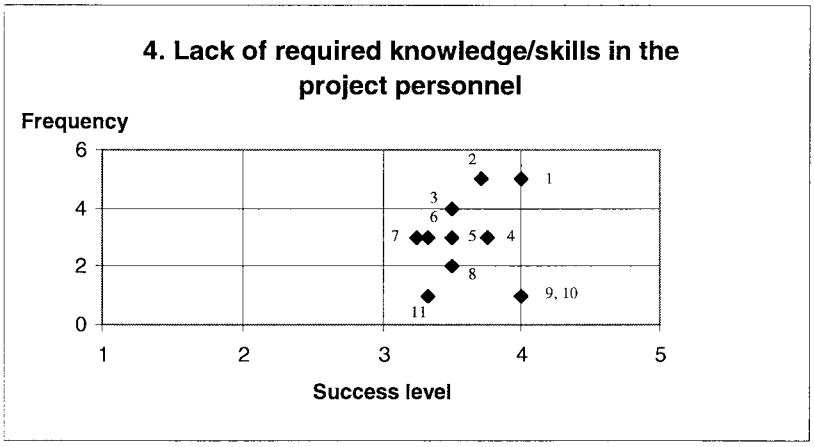
| 2 | Introduction of new technology | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *Avoid new technology.* | 3,78 | 8 | 30,22 |
| 2 | *Allocate time for learning when estimating work efforts.* | 3,33 | 9 | 30 |
| 3 | *Training and studying.* | 3,3 | 9 | 29,70 |
| 4 | *Obtain knowledge outside the project.* | 3,57 | 7 | 25 |
| 5 | *Assure that the technology is functional before choosing it.* | 3,55 | 7 | 24,82 |
| 6 | *Trusted suppliers and detailed contracts.* | 3,5 | 4 | 14 |
| 7 | *Prepare for problems: reserve money and develop alternative plans.* | 3,5 | 2 | 7 |
| 8 | *Reduce system features (requirements scrubbing).* | 3,5 | 2 | 7 |
| 9 | *Use SWOT-analysis when choosing technology.* | 4 | 1 | 4 |
| 10 | *Project manager's professional skills.* | 4 | 1 | 4 |
| 11 | *Proper project planning.* | 4 | 1 | 4 |
| 12 | *Separate innovations and research from development projects.* | 3 | 1 | 3 |
| 13 | *Develop other systems by using the same new technology.* | 3 | 1 | 3 |



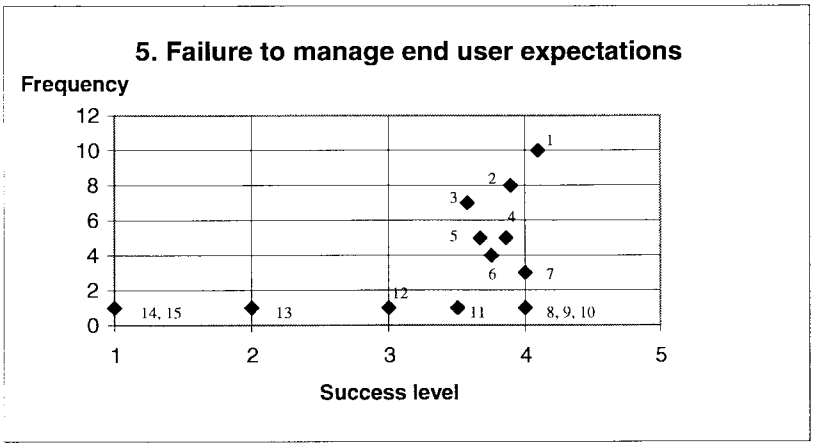
| 3 Lack of frozen requirements | | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *A disciplined procedure to manage chance requests (change control).* | 4 | 7 | 28 |
| 2 | *Users must understand system development methods and the value of requirements.* | 3,75 | 4 | 15 |
| 3 | *Experience and skill of a project team member, especially the project manager.* | 3,25 | 4 | 13 |
| 4 | *Changes are deferred for implementation later.* | 4 | 3 | 12 |
| 5 | *A sound project management method (frozen requirements).* | 4 | 3 | 12 |
| 6 | *Defining the requirements before starting implementation.* | 3,67 | 3 | 11 |
| 7 | *Keep the project size small and design simple.* | 3,33 | 3 | 10 |
| 8 | *Use 'rough' and limited specification.* | 2,25 | 4 | 9 |
| 9 | *Audits with customers and suppliers.* | 4 | 2 | 8 |
| 10 | *Fixed price contracting.* | 4 | 2 | 8 |
| 11 | *Effective project management methodology. Increasing PM's authority.* | 4 | 2 | 8 |
| 12 | *Allow the requirements determination phase to take time.* | 4 | 2 | 8 |
| 13 | *Provide stock money for unforeseen changes.* | 3 | 2 | 6 |
| 14 | *Prototypes and other mockups.* | 3 | 2 | 6 |
| 15 | *Good domain knowledge.* | 4 | 1 | 4 |
| 16 | *Write down also trivially evident tasks.* | 4 | 1 | 4 |



| 4 | Lack of required knowledge/skills in the project personnel | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *Identify and obtain required skills in time.* | 4 | 5 | 20 |
| 2 | *Obtaining help outside the project.* | 3,71 | 5 | 18,57 |
| 3 | *Taking the lack into account when estimating work effort.* | 3,5 | 4 | 14 |
| 4 | *Creating expectations of a success to the project: proper timing and resources.* | 3,75 | 3 | 11,25 |
| 5 | *Project specific training.* | 3,5 | 3 | 10,50 |
| 6 | *Staffing with capable users and releasing them to project.* | 3,33 | 3 | 10 |
| 7 | *Avoid new technology.* | 3,25 | 3 | 9,75 |
| 8 | *Good team relationships.* | 3,5 | 2 | 7 |
| 9 | *Management commitment.* | 4 | 1 | 4 |
| 10 | *Remove unskilled personnel.* | 4 | 1 | 4 |
| 11 | *Making it easy for those who can not ..* | 3,33 | 1 | 3,33 |

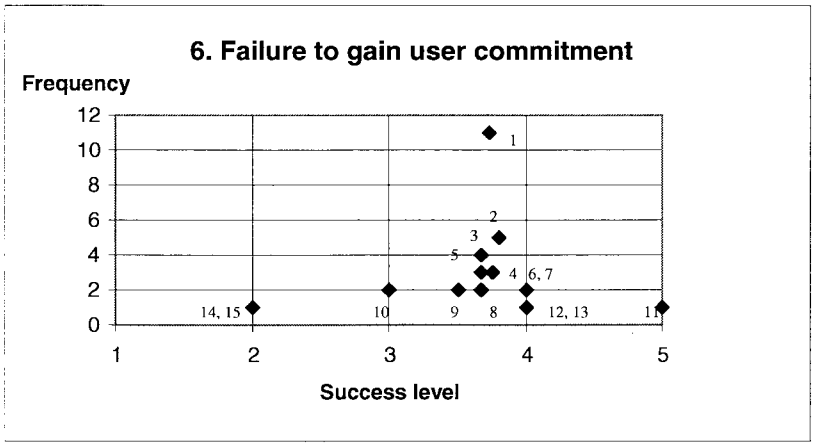


| 5 Failure to manage end user expectations | | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *User involvement in the project.* | 4,1 | 10 | 41 |
| 2 | *Joint seminars and workshops.* | 3,89 | 8 | 31,11 |
| 3 | *Inform users in the beginning about the new system.* | 3,57 | 7 | 25 |
| 4 | *Defining end user requirements in the beginning.* | 3,86 | 5 | 19,29 |
| 5 | *Prototypes, demos etc.* | 3,67 | 5 | 18,33 |
| 6 | *Training both during introduction and later while having used the system for some time.* | 3,75 | 4 | 15 |
| 7 | *Skilled users and developers.* | 4 | 3 | 12 |
| 8 | *Create realistic expectations in the selling phase.* | 4 | 1 | 4 |
| 9 | *Changes are deferred for implementation.* | 4 | 1 | 4 |
| 10 | *Prevent the use of the other systems.* | 4 | 1 | 4 |
| 11 | *The new system resembles the former one.* | 3,5 | 1 | 3,50 |
| 12 | *Selecting a suitable name for the project.* | 3 | 1 | 3 |
| 13 | *Off-the-self products are not customized.* | 2 | 1 | 2 |
| 14 | *Training in the very early stage of the project.* | 1 | 1 | 1 |
| 15 | *New system does not help users' own work.* | 1 | 1 | 1 |

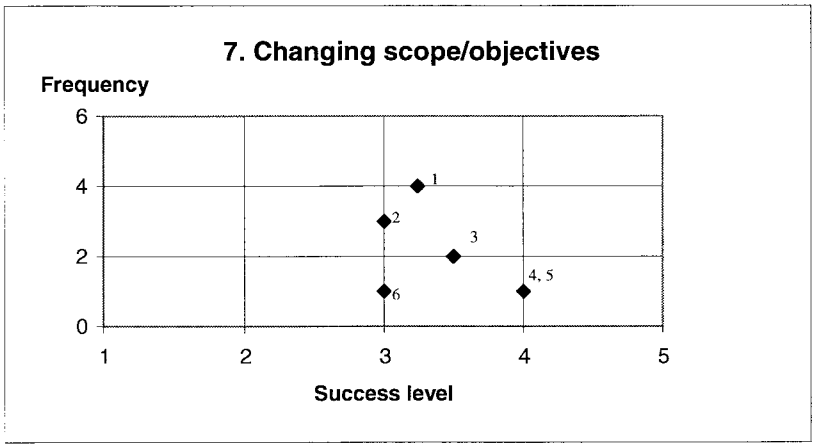


Appendix F: Risk management methods

| 6 | Failure to gain user commitment | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *Key end-user involvement in the project.* | 3,73 | 11 | 41,07 |
| 2 | *Inform users of new system features. Market the system.* | 3,8 | 5 | 19 |
| 3 | *Using a method which is easily understood by users.* | 3,67 | 4 | 14,67 |
| 4 | *Project manager's personality. Nominating the PM among users.* | 3,75 | 3 | 11,25 |
| 5 | *Training.* | 3,67 | 3 | 11 |
| 6 | *Choose innovative users who understand data models.* | 4 | 2 | 8 |
| 7 | *Growing sophistication of users* | 4 | 2 | 8 |
| 8 | *Projects based on business needs.* | 3,67 | 2 | 7,33 |
| 9 | *Evaluating user needs and project risks in advance.* | 3,5 | 2 | 7 |
| 10 | *Change management.* | 3 | 2 | 6 |
| 11 | *Share responsibility with the customer and the supplier.* | 5 | 1 | 5 |
| 12 | *Make the system critical.* | 4 | 1 | 4 |
| 13 | *The new system resembles the former one.* | 4 | 1 | 4 |
| 14 | *Letting someone else than the end user derive the requirements.* | 2 | 1 | 2 |
| 15 | *Install the system as soon as possible to customer.* | 2 | 1 | 2 |

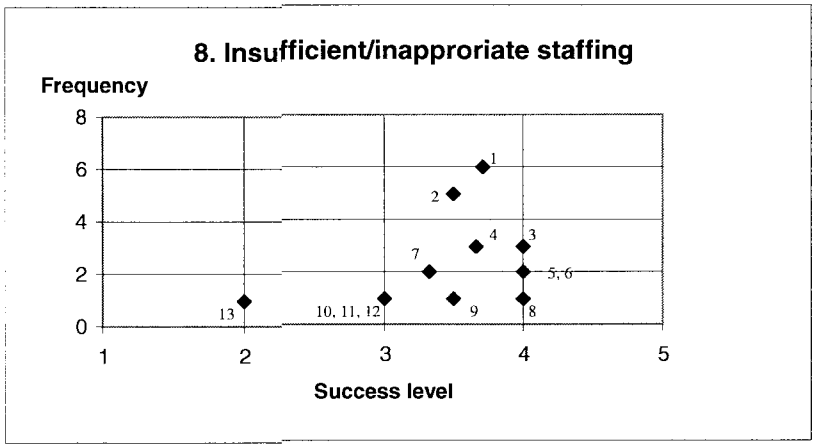


| 7 Changing scope/objectives | | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *Keep projects small.* | 3,25 | 4 | 13 |
| 2 | *Prepare for consequences, develop scenarios.* | 3 | 3 | 9 |
| 3 | *Disciplined project management method.* | 3,5 | 2 | 7 |
| 4 | *Communication between IT and management.* | 4 | 1 | 4 |
| 5 | *Technical solutions: new version may allow more users.* | 4 | 1 | 4 |
| 6 | *Experience.* | 3 | 1 | 3 |

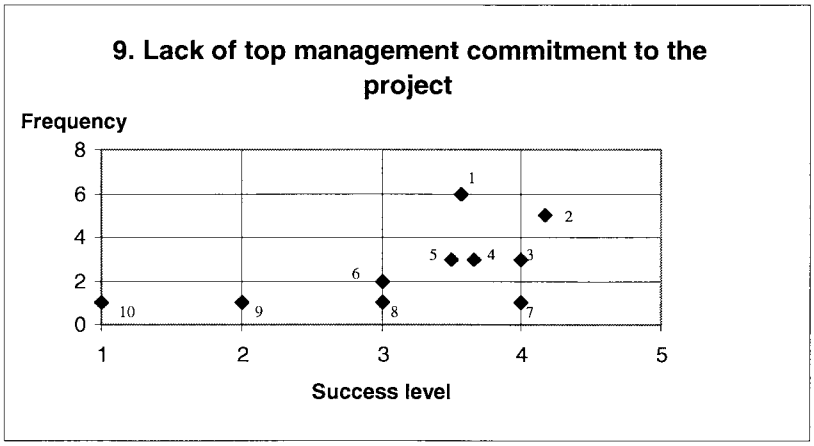


Appendix F: Risk management methods

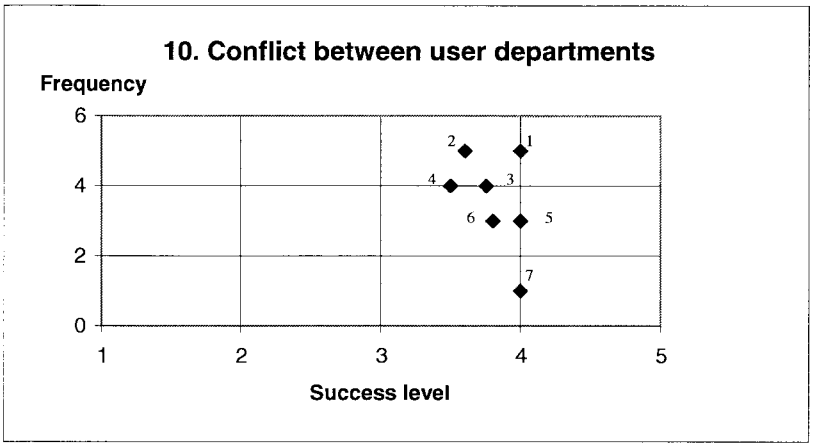
| 8 Insufficient/inappropriate staffing | | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *Proper planning and follow up.* | 3,71 | 6 | 22,29 |
| 2 | *Hire and release people to work in the project.* | 3,5 | 5 | 17,50 |
| 3 | *Keep projects in time.* | 4 | 3 | 12 |
| 4 | *Training.* | 3,67 | 3 | 11 |
| 5 | *Avoid new technology.* | 4 | 2 | 8 |
| 6 | *Collect a database of personnel's skills.* | 4 | 2 | 8 |
| 7 | *Optimizing both people and time.* | 3,33 | 2 | 6,67 |
| 8 | *Management commitment.* | 4 | 1 | 4 |
| 9 | *Hiring a good project manager.* | 3,5 | 1 | 3,50 |
| 10 | *Reduce system features.* | 3 | 1 | 3 |
| 11 | *Use methods enabling users to work independently.* | 3 | 1 | 3 |
| 12 | *Complain and renegotiate with the supplier.* | 3 | 1 | 3 |
| 13 | *Include the max total price in the contract.* | 2 | 1 | 2 |



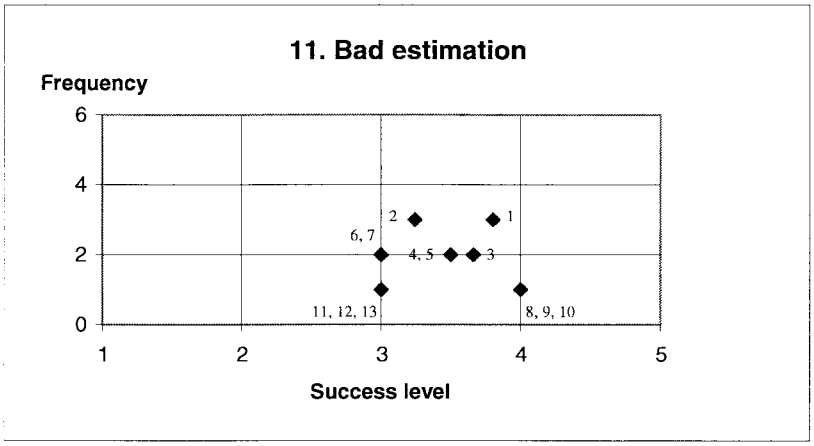
| 9 | Lack of top management commitment to the project | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *Increase knowledge of IT among top management.* | 3,57 | 6 | 21,43 |
| 2 | *Make projects important, large, or expensive.* | 4,17 | 5 | 20,83 |
| 3 | *Projects are based on business needs.* | 4 | 3 | 12 |
| 4 | *Communication between IT and management.* | 3,67 | 3 | 11 |
| 5 | *Mutual contribution to steering group.* | 3,5 | 3 | 10,50 |
| 6 | *Keep the management informed of the project situation.* | 3 | 2 | 6 |
| 7 | *Change in the management style.* | 4 | 1 | 4 |
| 8 | *Prepare for additional cost, if the management is not committed.* | 3 | 1 | 3 |
| 9 | *Leave the risk to the customer's responsibility.* | 2 | 1 | 2 |
| 10 | *Several ongoing large projects.* | 1 | 1 | 1 |



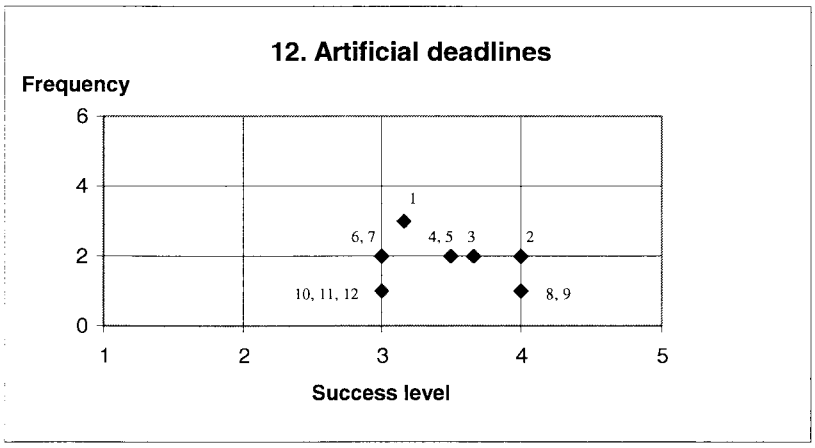
| 10 Conflict between user departments | | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *Obtain agreement beforehand.* | 4 | 5 | 20 |
| 2 | *Communicate and cooperate.* | 3,6 | 5 | 18 |
| 3 | *Take all parties into the project.* | 3,75 | 4 | 15 |
| 4 | *Make the system generic and user-friendly.* | 3,5 | 4 | 14 |
| 5 | *Define ownership.* | 4 | 3 | 12 |
| 6 | *Develop people skills.* | 3,8 | 3 | 11,40 |
| 7 | *Have only few departments.* | 4 | 1 | 4 |



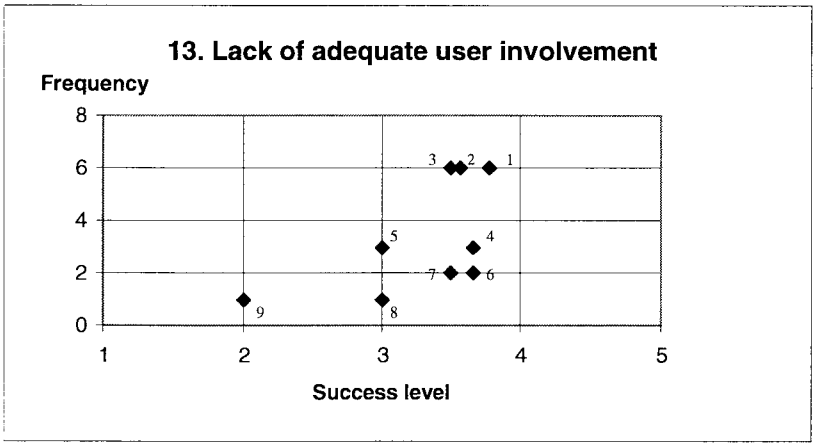
| 11 Bad estimation | | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *Tools for workload estimation.* | 3,8 | 3 | 11,40 |
| 2 | *Frequent team meetings; follow up remaining workload.* | 3,25 | 3 | 9,75 |
| 3 | *Good domain knowledge and technological skills.* | 3,67 | 2 | 7,33 |
| 4 | *Not fixed price contracting.* | 3,5 | 2 | 7 |
| 5 | *Obtaining help outside the project.* | 3,5 | 2 | 7 |
| 6 | *Allocate and coordinate workload effectively.* | 3 | 2 | 6 |
| 7 | *Proper project planning.* | 3 | 2 | 6 |
| 8 | *Team organization.* | 4 | 1 | 4 |
| 9 | *The customer accepts extensive testing.* | 4 | 1 | 4 |
| 10 | *Releasing a user from routine work.* | 4 | 1 | 4 |
| 11 | *Reduce system features.* | 3 | 1 | 3 |
| 12 | *Take into account the professional skills of the team members.* | 3 | 1 | 3 |
| 13 | *Launch the project quickly.* | 3 | 1 | 3 |



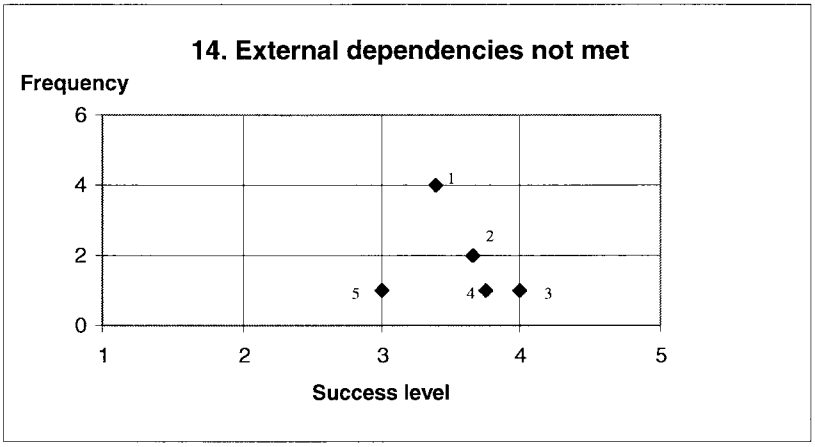
| 12 Artificial deadlines | | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *Frequent team meetings; follow up remaining workload.* | 3,17 | 3 | 9,50 |
| 2 | *Tools for workload estimation.* | 4 | 2 | 8 |
| 3 | *Good domain knowledge and technological skills.* | 3,67 | 2 | 7,33 |
| 4 | *Not fixed price contracting.* | 3,5 | 2 | 7 |
| 5 | *Obtaining help outside the project.* | 3,5 | 2 | 7 |
| 6 | *Allocate and coordinate workload effectively.* | 3 | 2 | 6 |
| 7 | *Proper project planning.* | 3 | 2 | 6 |
| 8 | *The customer accepts extensive testing.* | 4 | 1 | 4 |
| 9 | *Releasing a user from routine work.* | 4 | 1 | 4 |
| 10 | *Reduce system features.* | 3 | 1 | 3 |
| 11 | *Take into account the professional skills of the team members.* | 3 | 1 | 3 |
| 12 | *Launch the project quickly.* | 3 | 1 | 3 |



| 13 Lack of adequate user involvement | | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *Agreements on user involvement and workload beforehand.* | 3,78 | 6 | 22,67 |
| 2 | *Management commitment.* | 3,57 | 6 | 21,43 |
| 3 | *Release a user from routine work by reorganizing, or hiring new employees.* | 3,5 | 6 | 21 |
| 4 | *Inform users.* | 3,67 | 3 | 11 |
| 5 | *Invite and persuade.* | 3 | 3 | 9 |
| 6 | *User participants interest or a possibility of promotion.* | 3,67 | 2 | 7,33 |
| 7 | *Scheduling of the project. Note total workload of users and holidays.* | 3,5 | 2 | 7 |
| 8 | *Informal discussions with users.* | 3 | 1 | 3 |
| 9 | *Participation in seminars and training is made obligatory.* | 2 | 1 | 2 |



| 14 External Dependencies Not Met | | Success Level | Freq. | Relative importance |
|---|---|---|---|---|
| 1 | *Document all changes to old systems.* | 3,4 | 4 | 13,60 |
| 2 | *Test and audit.* | 3,67 | 2 | 7,33 |
| 3 | *Allocate time for unexpected delays when estimating work efforts.* | 4 | 1 | 4 |
| 4 | *Disciplined project management method.* | 3,75 | 1 | 3,75 |
| 5 | *Experienced supplier.* | 3 | 1 | 3 |



## Appendix G: Categorizing risk items according to Cule *et al.* (1999)

Coding: S = Structure, T = Task, C = Client, and E = Environment.

| | Identified Risk Items | Coding |
|---|---|---|
| 1 | Misunderstanding the requirements | S |
| 2 | Introduction of new technology | T |
| 3 | Lack of frozen requirements | C |
| 4 | Lack of required knowledge/skills in the project personnel | T |
| 5 | Failure to manage end user expectations | C |
| 6 | Failure to gain user commitment | C |
| 7 | Changing scope/objectives | E |
| 8 | Insufficient/inappropriate staffing | T |
| 9 | Lack of Top Management Commitment to the project | C |
| 10 | Conflict between user departments | C |
| 11 | Bad estimation | T |
| 12 | Artificial deadlines | C |
| 13 | Lack of adequate user involvement | C |
| 14 | External dependencies not met | E |
| 15 | Lack of effective project management methodology | S |
| 16 | Potential risk exposure over the system life cycle (New!) | T |
| 17 | New and/or unfamiliar subject matter for both users and developers. | C |
| 18 | Lack of effective development process/methodology | T |
| 19 | Staffing volatility | E |
| 20 | No planning or inadequate planning | S |
| 21 | Lack of cooperation from the users | C |
| 22 | Lack of available skilled personnel | T |
| 23 | Real-time performance shortfalls | T |
| 24 | Mismatch between company culture and required business process changes needed for new system. | E |
| 25 | Lack of "people skills" in project leadership | S |
| 26 | Excessive use of outside consultants | T |
| 27 | Stability of technical architecture | T |
| 28 | Multi-vendor projects complicate dependencies | T |
| 29 | A climate of change in the business and organizational environment that creates instability in the project. | E |
| 30 | Change in ownership or senior management | E |
| 31 | Lack of effective project management skills | S |
| 32 | Number of organizational units involved | C |
| 33 | "Preemption" of project by higher priority project | E |
| 34 | Poor team relationships | T |
| 35 | Unstable corporate environment | E |
| 36 | Growing sophistication of users leads to higher expectations. | C |
| 37 | Improper definition of roles and responsibilities | S |
| 38 | Poor risk management | S |
| 39 | Scope creep | T |
| 40 | Under funding of development | C |
| 41 | Trying new development method/technology during important project. | T |
| 42 | Lack of control over consultants, vendors, and sub-contractors. | E |

# Appendix H: Coding of risks and their management methods

Component: A = Actor, T = Task, S = Structure, and Te = Technology.

Strategy: I = Inhibiting, and C = Compensating.

| | | Component | Strategy |
|---|---|---|---|
| 1 | Misunderstanding the requirements | T | |
| *1* | *Use a disciplined development approach.* | Te | I |
| *2* | *Audit the tender; walkthroughs.* | S | I |
| *3* | *Review the requirements carefully with users.* | S | I |
| *4* | *Visualize the system and process using pilots, prototypes, etc.* | Te | I |
| *5* | *Test the validity of requirements.* | T | I |
| *6* | *Careful and thorough investigation of the subject domain and requirements.* | T | I |
| *7* | *Capable users.* | A | I |
| *8* | *Preparing for capacity problems.* | Te | C |
| *9* | *Keep the system simple; reduce functionality.* | T | C |
| *10* | *Increased experience in specifying requirements.* | A | I |
| *11* | *Requirements are defined by sole person (project manager).* | S | I |
| *12* | *Good domain knowledge (developers).* | A | I |
| *13* | *Team organization.* | S | I |
| *14* | *Abandon the project, develop a new contract, and new start.* | S | C |
| *15* | *Tools for workload estimation.* | Te | I |
| *16* | *Preparing for an unsuccessful project.* | S | C |
| *17* | *Change management/control.* | T | C |
| *18* | *Postpone the installation.* | Te | C |
| *19* | *Create realistic expectations in selling/marketing phase.* | T | I |
| 2 | Introduction of new technology | Te | |
| *1* | *Avoid new technology.* | Te | I |
| *2* | *Allocate time for learning when estimating work efforts.* | S | C |
| *3* | *Training and studying.* | A | C |
| *4* | *Obtaining knowledge outside the project.* | S | C |
| *5* | *Assure that the technology is functional before choosing it.* | Te | I |
| *6* | *Trusted suppliers and detailed contracts.* | A | I |
| *7* | *Prepare for problems: reserve money and develop alternative plans.* | S | C |
| *8* | *Reduce system features (requirements scrubbing).* | T | C |
| *9* | *Use SWOT-analysis when choosing technology.* | Te | I |
| *10* | *Project manager's professional skills.* | A | I |
| *11* | *Proper project planning.* | S | I |
| *12* | *Separate innovations and research from development projects.* | T | I |
| *13* | *Develop other systems by using the same new technology.* | Te | C |
| 3 | Lack of frozen requirements | T | |
| *1* | *A disciplined procedure to manage chance requests (change control).* | S | C |
| *2* | *Users must understand system development methods and the value of requirements.* | A | I |
| *3* | *Experience and skill of a project team member, especially the project manager.* | A | I |
| *4* | *Changes are deferred for implementation later.* | T | C |
| *5* | *A sound project management method (frozen requirements).* | S | I |
| *6* | *Defining the requirements before starting implementation.* | T | I |
| *7* | *Keep the project size small and design simple.* | T | I |
| *8* | *Use "rough" definitions.* | T | C |
| *9* | *Audits with customers and suppliers.* | S | I |

| | | | |
|---|---|---|---|
| *10* | *Fixed price contracting.* | S | I |
| *11* | *Effective project management methodology. Increasing PM's authority.* | A | I |
| *12* | *Allow the requirements determination phase to take time.* | S | C |
| *13* | *Provide stock money for unforeseen changes.* | S | C |
| *14* | *Prototypes and other mockups.* | Te | C |
| *15* | *Good domain knowledge.* | A | I |
| *16* | *Write down also trivially evident tasks.* | S | I |
| 4 | Lack of required knowledge/skills in the project personnel | A | |
| *1* | *Training and employees skills in general.* | A | I |
| *2* | *Identify and obtain required skills in time.* | S | C |
| *3* | *Taking the lack into account when estimating work effort.* | S | C |
| *4* | *Creating expectations of a success to the project: proper timing and resources.* | S | I |
| *5* | *Project specific training.* | A | C |
| *6* | *Staffing with capable users and releasing them to project.* | A | I |
| *7* | *Avoid new technology.* | Te | I |
| *8* | *Good team relationships.* | S | I |
| *9* | *Management commitment.* | S | I |
| *10* | *Removing an unskilled person* | A | C |
| *11* | *Making it easy for those who can not ..* | A | C |
| 5 | Failure to manage end user expectations | A | |
| *1* | *User involvement in the project.* | S | I |
| *2* | *Joint seminars and workshops.* | S | I |
| *3* | *Inform users in the beginning about the new system.* | A | I |
| *4* | *Defining end user requirements in the beginning.* | T | I |
| *5* | *Prototypes, demos etc.* | Te | C |
| *6* | *Training both during introduction and later while having used the system for some time.* | A | C |
| *7* | *Skilled users and developers.* | A | I |
| *8* | *Create realistic expectations in the selling phase.* | T | I |
| *9* | *Changes are deferred for implementation later.* | T | C |
| *10* | *Prevent the use of the other system.* | Te | I |
| *11* | *The new system resembles the former one.* | Te | I |
| *12* | *Selecting a suitable name for the project.* | S | I |
| *13* | *Off-the-self products are not customized.* | Te | C |
| *14* | *Training in the very early stage of the project.* | A | C |
| *15* | *New system does not help users' own work.* | T | C |
| 6 | Failure to gain user commitment | S | |
| *1* | *Key end-user involvement in the project.* | S | I |
| *2* | *Inform users of new system features. Market the system.* | A | I |
| *3* | *Using a method which is easily understood by users.* | Te | I |
| *4* | *Project manager's personality. Nominating the PM among users.* | A | I |
| *5* | *Training.* | A | C |
| *6* | *Choose innovative users who understand data models.* | A | I |
| *7* | *Growing sophistication of users* | A | I |
| *8* | *Projects based on business needs.* | T | I |
| *9* | *Evaluating user needs and project risks in advance.* | T | I |
| *10* | *Change management.* | T | C |
| *11* | *Share responsibility with the customer and the supplier.* | S | I |
| *12* | *The developed system is critical.* | T | I |
| *13* | *The new system resembles the former one.* | Te | C |

| | | | |
|---|---|---|---|
| *14* | *Letting someone else than the end user derive the requirements.* | S | I |
| *15* | *Install the system as soon as possible to customer.* | Te | I |
| 7 | Changing scope/objectives | T | |
| *1* | *Keep projects small.* | T | I |
| *2* | *Prepare for consequences, develop scenarios.* | S | C |
| *3* | *Disciplined project management method.* | S | I |
| *4* | *Communication between IT and management.* | S | I |
| *5* | *Technical solutions: new version may allow more users.* | Te | C |
| *6* | *Experience.* | A | I |
| 8 | Insufficient/inappropriate staffing | S | |
| *1* | *Proper planning and follow up.* | S | C |
| *2* | *Hire and release people to work in the project.* | S | I |
| *3* | *Keep projects in time.* | S | I |
| *4* | *Training.* | A | C |
| *5* | *Avoid new technology.* | Te | I |
| *6* | *Collect a database of personnel's skills.* | A | I |
| *7* | *Optimizing both people and time.* | S | I |
| *8* | *Management commitment.* | A | I |
| *9* | *Hiring a good project manager.* | A | I |
| *10* | *Reduce system features.* | T | C |
| *11* | *Use methods enabling users to work independently.* | Te | I |
| *12* | *Complain and renegotiate with the supplier.* | S | C |
| *13* | *Include the max total price in the contract.* | S | I |
| 9 | Lack of Top Management Commitment to the project | A | |
| *1* | *Increasing knowledge of IT among top management.* | A | I |
| *2* | *Make projects important, large, or expensive.* | T | I |
| *3* | *Projects are based on business needs.* | T | I |
| *4* | *Communication between IT and management.* | S | I |
| *5* | *Mutual contribution to steering group.* | S | C |
| *6* | *Keep the management informed of the project situation.* | A | I |
| *7* | *Change in the management style.* | S | I |
| *8* | *Prepare for additional cost, if the management is not committed.* | S | C |
| *9* | *Leave the risk to the customer's responsibility.* | S | I |
| *10* | *Several ongoing large projects.* | T | I |
| 10 | Conflict between user departments | S | |
| *1* | *Obtain agreement beforehand.* | T | I |
| *2* | *Communicate and cooperate.* | S | C |
| *3* | *Take all parties into the project.* | S | I |
| *4* | *Make the system generic and user-friendly.* | Te | C |
| *5* | *Define ownership.* | S | I |
| *6* | *Develop people skills.* | A | I |
| *7* | *Have only few departments.* | S | I |
| 11 | Bad estimation | T | |
| *1* | *Tools for workload estimation.* | Te | I |
| *2* | *Frequent team meetings; follow up remaining workload.* | S | C |
| *3* | *Good domain knowledge and technological skills.* | A | I |
| *4* | *Not fixed price contracting.* | S | I |
| *5* | *Obtaining help outside the project.* | S | C |
| *6* | *Allocate and coordinate workload effectively.* | S | C |
| *7* | *Proper project planning.* | S | I |
| *8* | *Team organization.* | S | I |

| | | | |
|---|---|---|---|
| *9* | *The customer accepts extensive testing.* | Te | C |
| *10* | *Releasing a user from routine work.* | S | C |
| *11* | *Reduce system features.* | T | C |
| *12* | *Take into account the professional skills of the team members.* | A | C |
| *13* | *Launch the project quickly.* | S | C |
| 12 | Artifical deadlines | S | |
| *1* | *Frequent team meetings; follow up remaining workload.* | S | C |
| *2* | *Tools for workload estimation.* | Te | I |
| *3* | *Good domain knowledge and technological skills.* | A | I |
| *4* | *Not fixed price contracting.* | S | I |
| *5* | *Obtaining help outside the project.* | S | C |
| *6* | *Allocate and coordinate workload effectively.* | S | C |
| *7* | *Proper project planning.* | S | I |
| *8* | *The customer accepts extensive testing.* | Te | C |
| *9* | *Releasing a user from routine work.* | S | C |
| *10* | *Reduce system features.* | T | C |
| *11* | *Take into account the professional skills of the team members.* | A | C |
| *12* | *Launch the project quickly.* | S | C |
| 13 | Lack of adequate user involvement | S | |
| *1* | *Agreements on user involvement and workload beforehand.* | S | I |
| *2* | *Management commitment.* | A | I |
| *3* | *Releasing a user from routine work by reorganizing, or hiring new employees.* | S | C |
| *4* | *Inform users.* | A | C |
| *5* | *Invite and persuade.* | S | C |
| *6* | *User participants interest or a possibility of promotion.* | A | I |
| *7* | *Scheduling of the project. Note total workload of users and holidays.* | S | I |
| *8* | *Informal discussions with users.* | A | C |
| *9* | *Participation in seminars and training is made obligatory.* | S | C |
| 14 | External dependencies not met | S | |
| *1* | *Document all changes to old systems.* | T | I |
| *2* | *Test and audit.* | Te | C |
| *3* | *Allocate time for unexpected delays when estimating work efforts.* | S | C |
| *4* | *Disciplined project management method.* | S | I |
| *5* | *Experienced supplier.* | A | I |