Dealing with risk : beyond gut feeling : an approach to risk management in software engineering

Eindhoven
University of Technology
The Netherlands

Research Report

FACULTY OF TECHNOLOGY MANAGEMENT

Dealing with risk: beyond gut feeling: an approach to risk management in software engineering

by

F.J. Heemstra, R.J. Kusters, R. Nijhuis, Th.M.J. van Rijn

Report EUT/BDK/86
ISBN 90-386-0415-7
ISSN 0929-8479
Eindhoven 1997

Dealing with risk : beyond gut feeling : an approach to risk management in software engineering

by

F.J. Heemstra, R.J. Kusters, R. Nijhuis, Th.M.J. van Rijn

Report EUT/BDK/86
ISBN 90-386-0415-7
ISSN 0929-8479
Eindhoven 1997

Keywords: RISK-ANALYSIS / SOFTWARE DEVELOPMENT

Eindhoven University of Technology
Faculty of Technology Management
Eindhoven, The Netherlands

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Heemstra, F.J.

Dealing with risk : beyond gut feeling : an approach to risk management in software engineering / by F.J. Heemstra, R.J. Kusters, R. Nijhuis and Th.M.J. van Rijn. - Eindhoven : Technische Universiteit Eindhoven, 1997. -
(Report EUT/BDK, Eindhoven University of Technology, Department of Industrial Engineering and Management Science, ISSN 0929-8479; 86). -
ISBN 90-386-0415-7
NUGI 684
Subject headings: Risk Analysis / Software Development

Dealing with risk: beyond gut feeling

an approach to risk management in software engineering

F.J. Heemstra	The Open University, Heerlen
R.J. Kusters	Eindhoven University of Technology, Eindhoven
R. Nijhuis	NOB, Hilversum
Th.M.J. van Rijn	ORIGIN, Eindhoven

CONTENTS

1. INTRODUCTION 4

1.1. Risk management and software engineering 4

1.2. Objectives and intended audience for this book 5

1.3. The authors' approach to risk management 6

1.4. Background of the study 7

1.5. Acknowledgements 7

1.6. Structure of this book 7

2. DEFINING RISK MANAGEMENT 9

2.1. Risk 9

2.2. Risk exposure 11

2.3. Risk sources 12

2.4. Risk management 14

2.5. Risk behaviour 16

2.6. Risk management as organisational process 16

3. RISK MANAGEMENT ACTIVITIES 20

3.1. Introduction 20

3.2. Identifying risks 23

3.3. Analysing risks 24

3.4. Prioritising risks 25

3.5. Conceiving actions 26

3.6. Choosing actions 28

3.7. Monitoring effect 30

4. CRITICAL FACTORS IN MANAGING SOFTWARE DEVELOPMENT 31

4.1. Introduction 31

4.2. Premises related to an individual 32

4.3. Premises related to a group 34

4.4. Premises related to an organisation 35

4.5. The relationship between risk management and project management 38

4.6. The formal roles in relation to a software project 39

5. EXPLORING THE SOURCES OF RISK 43

5.1. Introduction 43

5.2. Categories of risk sources 43

5.3. The checklist 46

5.4. Assessment of risk sources 54

5.5. Discussion 56

5.6. Conclusion 58

6. AN IMPLEMENTATION OF THE RISK MANAGEMENT APPROACH 60

6.1. Introduction 60

6.2. Risk Management in practice: an example 60

6.3. Introduction, use and control of a risk management procedure 69

6.4. Evaluation of the risk management method 70

7. BIBLIOGRAPHY 74

7.1. Books 74

7.2. Papers 77

1. INTRODUCTION

1.1. Risk management and software engineering

The theme of this book is focused on the issue how to deal with risks as applicable to *software development* processes. It is assumed that these processes are organised in the form of a project. Such a project can be managed and carried out by the organisation requiring the software itself, or by an outside service company with various types and degrees of participation.

In spite of progress made in the field of constructing software during the past decades, it can easily be defended that software engineering is still in its childhood. Although today less projects end in total failure than in the early days of Information Technology, still frequently budgets are over-run, committed delivery dates are violated and users are dissatisfied if not disappointed with the end-result. The bottom-line is that the net returns from such newly developed systems are less than foreseen at the start of the project. The logical question arising from this situation is; what can be done to minimise the probability of these undesirable effects ? This touches the essence of risk management, as risk management in general focuses on ways for reducing the discrepancies between the intended and the actual outcomes of human endeavour.

As such risk management concerns general practices which are embedded in human behaviour. Whether we cross the street, embark on a plane, apply for a job, or start a conversation with an unknown person, some form of risk assessment normally precedes our actual behaviour. Usually we are even not, or only partially aware of these forms of risk management. For our daily activities - like crossing the street - this is quite normal. Assessing the risks has become a routine and therefore an implicit activity, relying on our past experience. For these routine activities this way of handling risks is even a necessity to carry them out efficiently. This allows us to focus our attention on those matters which are less trivial.

Strikingly, in the case of these complex activities, like software development, too often the non-routine nature is not fully appreciated. As a consequence an intuitive and therefore highly subjective and biased approach towards risks is frequently found. Instead of pursuing a rational approach, in which one would consciously identify the things which can go wrong, one hopes for the best. Even the more unfamiliar the situation at hand is, often the greater the tendency is to act on purely intuitive grounds.

1.2. Objectives and intended audience for this book

In order to arrive at software engineering as a mature profession, methodologies are required to make the process of software development controllable and therefore predictable. Apart from factors like project phasing, planning and budgeting, it is the authors' opinion that a rational and as much as possible an objective approach to risk management should be an essential aspect of any mature software engineering methodology.

However, it is in particular in the area of risk management, where the authors have found commonly applied methodologies falling short. The attention paid to risks in most methodologies is either constrained to general remarks and guidelines, or relies heavily on checklists of risks, exhibiting various degrees of sophistication and size. As such checklists can be useful means in helping an individual to recollect past experience. However, the danger is that, without further considerations, they become simply used as "cookbook" type of prescriptions. Such an approach would ignore the complexity involved with software development and the combination of intelligence and experience required in managing this process.

The objective of this book is to contribute to the field of risk management in software engineering. This by advocating a different approach to risks than commonly encountered in practice and applied methodologies. Apart from contributions to risk management theory and the general understanding of the phenomenon of risk, guidelines, aids and recommendations are given for handling risks in practice. The audience for this book is therefore felt to consist of scholars in the management of software development, being practitioners or members of the scientific community.

Most certainly risk management will evolve further in the coming decades as software engineering matures further. This on the one hand based on new technology and on the other resulting from a progressively improving understanding of how to manage the process of software development. The authors' claims with this book are no more than just providing another stepping-stone on this evolution trajectory.

It should be emphasised that the scope of this book is limited to software development in an organisational context (e.g. within industry, banks, government, health care). The authors' focus is not on the development of technical software, like imbedded software as commonly found today in consumer products, although some of the considerations may apply to this field as well. Also other categories of projects found in the field of Information Technology are left untouched here (e.g. implementation of standard software packages).

1.3. The authors' approach to risk management

In the authors' view, risk management needs to be an *integral part* of any method for software development. As the various activities - together making-up a software project - are considered from a time, money and resource point of view, so are they to be addressed from a risk perspective. Only this way a conscious trade-off can be made between the projected desired outcomes of a particular activity and its associated risks, contributing to the success of the overall project.

Risks need to be addressed *explicitly*, rather than handling them implicitly and thereby relying for success on the personal and subjective judgement of a single individual. This requires a focused analysis and resolution approach, based on both a generic understanding of the phenomenon of 'risk' and how this works out specifically in a software engineering environment.

The authors feel that the most effective way for handling risks is to do this in a *pro-active* rather than in a re-active fashion. The saying "a problem is a risk whose time has come" adequately expresses why the authors hold this particular view. Complementary to the required formal approach for handling risks, this requires practical experience in software engineering. The latter to make sure that the proper risks are identified and addressed at the right moment in time.

Apart from addressing risks explicitly, a *joint involvement* of all agents concerned with the project is an additional manner to address risks more objectively. The authors' view on software projects is, that it is beneficial to consider them as contractual arrangements between all parties involved. This requires reflection and explicit definition of the roles of those who will be involved in a project. The project's contract should be such, that all parties gain by ending the project successfully. As a consequence a common understanding of all risks is an essential prerequisite for arriving at a good project contract. Considered in this way, risk management is a group decision making process, based on shared information, It implies the need to explicitly make a trade-off between the concerns of the various parties involved.

Organisations should pursue the evolution of risk management into an *institutionalised practice* within their managerial processes. As any other managerial process, risk management needs to be carried out on a perpetual basis, interwoven with the business decisions being taken at the proper moments in time. In other words, risk management should not be exercised purely on a project-by-project basis, but should strive at continuous improvement in handling risks. This requires an organisational learning process, in which projects are evaluated and the findings are converted in improved risk management practices.

These are the elements felt by the authors to constitute the key principles of a sound approach for risk management. Such an approach should be well embedded in any managerial behaviour as found in an organisational context, hence in the management of software development.

1.4. Background of the study

The material presented in this publication is based on the combined findings of two research projects. The first project was carried out in The Netherlands at the *Ministry of Public Affairs - Water Management* (managed by Drs J.W. Tierolf) in which researchers from the Faculty of Industrial Engineering at the *Eindhoven University of Technology* participated (Prof. Dr Ir. F.J. Heemstra and Dr R.J. Kusters). This project provided the inputs for the practical approach as presented in the last part of this book. The second project was carried out at the Dutch software company *BSO/Origin* by Ir. R.J. Nijhuis, covering his master's graduation project at the Faculty of Industrial Engineering at the *Eindhoven University of Technology*. This project was supervised by Th.A. Hanssen, Dr Ir. Th.M.J. van Rijn (both from BSO/Origin), Dr R.F. de Vries and Dr J. Halman (both from *Eindhoven University of Technology*). Theoretical contributions, as contained by the first chapters, are primarily derived from this investigation.

The outcomes of both research projects were presented independently at the workshop "on Software Engineering for Large Complex Systems" as organised by RSG.3 "on Software Engineering" (NATO AC/243 Panel 11 RSG.03) in The Hague (The Netherlands), which was held from 19 until 21 October 1993. As the result of the response received at this workshop and similarities found in both approaches, the authors decided to prepare a joint publication, of which this book is the tangible result.

1.5. Acknowledgements

The authors wish to express their gratitude to those who have been involved in commenting on earlier drafts of this book. Constructive comments were received from Drs J.W. Tierolf (Dutch *Ministry of Public Affairs - Water Management)*, Th.A. Hanssen (*BSO/Origin Holding*), G.J. Vlasveld and P. Langbroek (*BSO/Origin Quality Innovation*), Dr R.F. de Vries (*Eindhoven University of Technology*).

1.6. Structure of this book

In this book the focus will first be on risk management as such. This covers the chapters 2 and 3. Although, as stated earlier, it is advocated that risk management should be an integral part of the managerial process, for exploratory purposes the subject is first addressed in these chapters as if it were a discipline on its own. Next, in chapter 4 the relationship between risk

management and software project management is explored, as well as some general premises that underlie the approach chosen in this book. Finally, in chapters 5 and 6 risk management will be approached from the perspective of an integral development methodology. Essentially this part represent a way of implementing the theory presented in the earlier chapters. An elaborate appendix is added, which can be used as a guideline in managing risks in practice. However, it is not recommended to use this appendix in an isolated way, without having explored the basic contents of this book.

2. DEFINING RISK MANAGEMENT

In this chapter first the basic elements of risk management are explored. The phenomenon of 'risk' is investigated and defined. It is advocated that for effective risk management one has to address the underlying factors, acting as the real causes of risk; the 'risk sources'. After having addressed some of the basic concepts, 'risk management' is considered as an organisational process. It is argued that risk management needs to be embedded in an organisation in such a way that organisational learning in this field is facilitated, resulting in continuous improvement.

2.1. Risk

As human beings we consciously carry out *activities* to reach certain ends. We seek to fulfil our objectives by carrying out a variety of activities. These activities range from daily short term practices like taking food for staying alive, to long term strategies such as the pursuance of a career. Stated in an abstract sense, human beings seek to reach states which they assign a higher preference than their initial states. It are activities, consciously carried out, which provide the bridge between old and new states, being separated from each other in time.

For instance in the evening a person leaves the office in order to go home and have dinner, someone watches a football game to get rid of his daily stress, another person pays her telephone bill to be ensured of future services, etc. These are examples of activities which people carry out to reach different states, what they find necessary for particular reasons. To summarise, people purposefully carry out activities to obtain certain desired results.

Apart from the activities executed by certain agents, there are also activities which take place beyond the control of an agent (e.g. rain, earthquakes, diseases). These activities result in different states as well, but whether these states have a higher or lower preference depends on the extent in which human beings are affected (e.g. most people don't care about an earthquake in an inhabited area, not causing any human casualties). We shall refer to this last category of activities as *non-purposeful activities* and to the category of activities consciously carried out by an agent[1] as *purposeful activities*. The purposeful activities are considered to be those activities carried out by an identifiable agent in order to reach identifiable objectives or outcomes. All other activities will be considered as non-purposeful activities.

---

[1] *It should be noted that the agent not necessarily has to a human being. Also animals display purposeful behaviour, however we shall discard from this category of agents in the remainder of the book.*

The remainder of this book will be limited to purposeful activities, as these are the ones which are to be controlled and therefore are subject to risk management. The non-purposeful activities can be considered as events or happenings, which one can try to predict, avoid or insure oneself against. However, these activities are beyond the control of an agent. As a consequence, although risks may be defined in conjunction to this type of activities, risk management at best will be limited to dealing with the negative effects, rather than fundamentally addressing their underlying causes.

For the specific area of concern here - software development - this limited focus does not hamper the investigation, as software development can be considered to constitute a specific category of purposeful activities.

For any initiated activity, one can never be completely sure beforehand about the attainment of its intended outputs. For example an unforeseen event may alter the execution of the activity, or the mechanisms applied in carrying out the activity fail to perform according to expectations. It is in particular the uncertainty in relation to the specifically intended output of an activity, which provides us the background for giving the following definition of *'risk'* :

> *a risk is the probability of a certain deviation between the intended and the actual output of an activity*

Inherent to 'risk' is the involvement of *time* and hence of *uncertainty*. The fact that an activity takes time between its start and finish, introduces uncertainty about its outcomes, as long as the activity has not come to an end. If an activity could be carried out in an infinitely small period of time, there would not be any uncertainty, as initial and final states would be known at virtually the same moment. As a consequence 'risk' would not apply to such type of activities.

Given an initial state and a desired future one, an agent usually has a choice from a number of alternative courses of action. This all together constitutes an agent's *decision space* at a certain moment in time. Making a choice is a non-trivial decision, as alternative activities will differ from an economic point of view in two ways :

- the amount of effort required to carry out the activity for obtaining the desired output;
- the risks associated with obtaining the intended outputs by the activity.

Assuming that the number of alternative actions one can reasonably chose from is finite, a decision space as a whole inherently has a certain amount of risk, which is related to its objective.

Let's take an example. In analysing user requirements one can decide either to do this in a conventional mode by question and answering types of sessions, or additionally, it can be

decided to build a prototype. The latter option takes more time and resources than the first one, however it reduces the likelihood that requirements are misinterpreted. As a consequence there is a higher likeliness that the system to be built will meet the requirements from its user community. Whether the second option will be chosen depends on the trade-off to be made between the extra effort to be spent and the expected reduction of risk.

The fact that it is decided to develop a system as such embodies certain risks, regardless of how this task is being handled. In a way this can be seen as an entrepreneurial type of risk. When it is felt beforehand that the task - developing a system - is achievable, then the overall risk will be excepted. What remains to be done, from a risk management perspective, is to carry out the various activities in such a way, that they lead to the desired end result against acceptable effort and to make sure that risks stay within acceptable limits.

2.2. Risk exposure

Risks related to a particular activity can be multiple, when an activity's specified output is a compound one, consisting of a number of different components or aspects. For instance, if an intended output is specified as 'the development of a system', there can be deviations as to timing of finishing, quality of the system, completeness, etc. Examples of such deviations are then the possibility that the development activity does not produce the output on time, the produced output does not satisfy formulated criteria, or an additional undesired output is generated. Each of these possible effects has its own probability of occurring.

When the intended outputs and possible deviations are projected on a preference scale, one can express what Boehm (1989) calls *'risk exposure'*. For a single event, the risk exposure would be a function of the *risk impact* and the *risk probability*. 'Risk impact' can be described as this negatively valued deviation between the intended and the actual outcome of the event. 'Risk probability' is the likelihood that this deviation actually occurs.

The earlier given definition of 'risk' can be classified as a phenomenological one, in a sense that it did not make any references to the value or importance of 'risk'. However, this is remedied by this definition of risk exposure which recognises that risk is characterised both by effect ("risk impact") as well as by likelihood ("risk probability").

Considering the context of software development, the preference scale applied in determining the risk impact should be directly related to the project's agreed objectives. The risk exposure in a particular type of decision making situation (i.e. the situation where an agent has the choice from a number of alternative activities), would then equal the sum of the (negatively valued) impact of all possible outcomes associated with a particular alternative times their individual probability of occurring (see also [ref. 3]). Or in more simpler terms, risk exposure can be considered as the anticipated injury that one might incur in starting up certain activities.

2.3. Risk sources

So far a risk has been considered as the possibility that a negative effect occurs. In managing these risks, obviously one would like to reduce this possibility. This implies the need to look beyond a risk and to investigate its origin; the *risk source*. Only by identifying and manipulating the factors which cause a certain risk to exist at the start of a certain activity, one can handle risks in a pro-active way.

> *A risk source is a factor potentially causing an activity to produce a deviation between the intended and the actual output of that activity.*

Returning to the statement that a risk is related to an activity, one can also say that a risk source is a factor that is related to an activity. This leads to the question: what types of factors should be looked at ? In order to answer this question a view is adopted, which is based on general systems theory.

In carrying out an activity one or several *mechanisms* are used. A mechanism can be the agent carrying out the activity for his or her personal benefits, an other agent acting as instructed by someone else, man made objects like machines, or even natural objects being manipulated to carry out an activity. A mechanism in general is the facility, which has the potential to produce an output after having been activated and provided with the proper inputs.

The mechanism may require *inputs* to be converted into an output (e.g. baking a cake from ingredients), or it may produce an output without having been provided with additional inputs (e.g. a human individual generating an idea). An input is converted into an output and therefore no longer exists after completing the activity, while the mechanism is not - or at least not noticeably - affected by the execution of the activity. The same mechanism can therefore be re-used again for carrying out subsequent activities.

The way in which the mechanism converts an input into an output (or simply generates an output) is by definition not of our concern. The mechanism is considered as an elementary tool, or black-box, which is supposed to operate in a predictable way. For example, in using a personal computer for text processing, as a user one is not concerned about the internal processes of the computer. It is merely considered as a tool and as long as it fulfils it function properly, one does not bother about the ways in which the outputs are being obtained.

However, mechanisms may dysfunction or exhibit a different behaviour from what is required. The latter in particular when the mechanism is of human nature. Human beings have their own personal - often hidden - objectives, varying moods, imperfect knowledge etc. This makes that their behaviour can never be predicted with full certainty.

Also non-human mechanisms may produce unanticipated outputs, for example simply because their behaviour is not completely known to its user. An often encountered reason for a non-human mechanism failing to produce a certain output, is that one of its components has broken down without prior notice (e.g. due to lack of timely maintenance).

It is this type of dysfunction of a mechanism, which constitutes a generic source of risks. Even when a mechanism carries out an activity according to what has been specified, it may fail to produce the required output. This can happen when its inputs are different from what has been specified. This phenomenon forms a second generic category of risk sources.

'Mechanisms' and 'inputs' provide us with two of the three fundamental categories of risk sources. The third category that is distinguished are *controls*. Controls represent a special type of inputs. While the inputs - as defined in a narrow sense - are being transformed by an activity into outputs, the controls merely influence the way in which the activity is performed, without being transformed themselves into the outputs. Controls can be successful to various degrees in exerting influence on the execution of an activity, even when inputs and mechanisms completely satisfy the specifications required for producing the desired output.

In the example of the personal computer, the user can control the process after the system has started-up, by making selections from available applications, choosing which file to work on etc. To be more specific, when the user is processing text, he or she will provide both the inputs - the raw text data - and the controls - e.g. choosing a certain style for a document - which will enable the computer to produce the text in a certain format as desired by the user.

A simple model can now be drawn - figure 2.1[2] - to show the relation between an activity and the above mentioned factors. It depicts an activity as an abstract entity, emerging from the interaction between inputs, mechanisms and controls and resulting in concrete outputs.

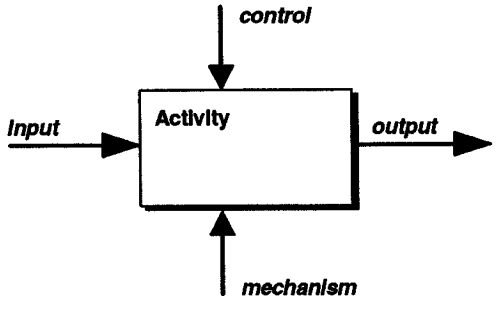

Figure 2.1: Risk source categories and their impact on activities.

[2] *Drawing conventions are based on SofTech's Structured Analysis and Design Technique (SADT™).*

2.4. Risk management

So far the basic concepts related to the phenomenon of 'risk' were explored briefly from an analytical perspective. The phenomenon of risk was defined and the basic types of factors were identified lay at the origin of risk. Inputs, mechanisms and controls provide the basic types of sources of risks. At the same time, they provide us the essential handles for managing identified risks. By selecting and synchronising inputs, mechanisms and controls, one can manage the execution of an activity. Providing additional inputs, modifying an input, aligning a control to the specifications of an input, upgrading the state of a mechanism, are all examples of risk reducing measures which can be taken prior to the execution of an activity in order to increase the likelihood of obtaining certain outputs. Key to managing risks pro-actively therefore is to obtain prior knowledge about the initial conditions for an activity, to evaluate and to influence those prior to starting the activity.

However, handling risks within an organisational environment can hardly be done on an activity-by-activity basis only. The attainment of an organisation's objectives usually brings about complex chains or networks of activities, exhibiting mutual interactions of various nature. Such a chain or network of mutually dependent activities will be called *'process'*.

For an organisation to attain its goal, this implies that the process required to produce the corresponding output has to be considered in its entirety at the first place. Next and in addition to, one has to focus on the activities individually and evaluate their result - in terms of outputs - from the perspective of their contributions to the final outcome of the process of which they are part. At the end of every activity within a process, there is the possibility to influence the conditions for subsequent dependent activities to take place under more favourable circumstances when necessary (i.e. by selecting the right mixture of inputs, mechanisms and controls). Hence, one uses the defined output for the process as a guideline in managing the individual activities, which should contribute to the final result.

Figure 2.2 schematically demonstrates the relationship between a process and its constituent activities. It also indicates the places where measurement of outputs can be done, to decide upon the execution of subsequent dependent activities.

Software development will be considered as a certain type of process, being carried out in an organisational environment. The development process consists of a large number of individual activities of various types (e.g. requirements analysis, data base design, testing) to be carried out by different types of specialists (mechanisms). The process aims at the development and hand-over of a new system, while the individual activities need to contribute to this ultimate objective.

Risk management will be considered in conjunction to processes as have been defined here. The following general definition of *risk management* can now be given, which will be used

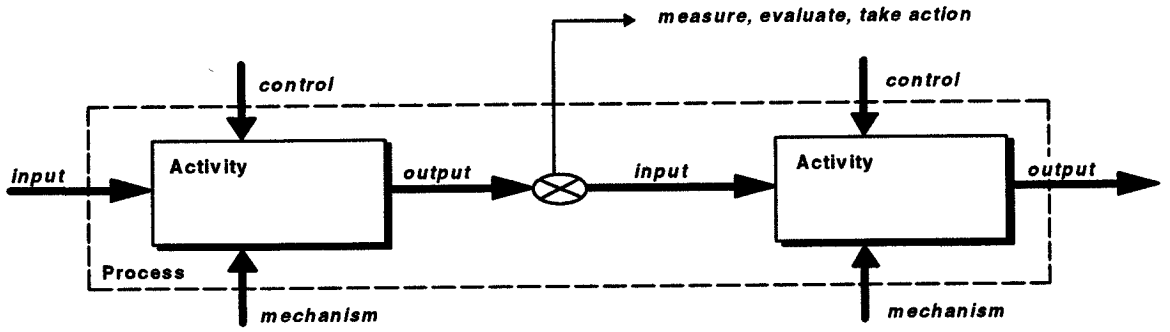

Figure 2.2: Relationship between a process and its activities.

throughout this book :

> *risk management is the totality of activities specifically deployed to minimise the probability for a process to result in deviations relative to its redefined output.*

The definition expresses a fundamental aspect of risk management. Rather than defining what is foreseen to happen, as key to basic planning and control activities, risk management explores the question why certain things may *not* happen as foreseen. Essentially risk management addresses the state of affairs from a negative point of view. Consequently it requires those who are engaged in this process to play the devil's advocate from time to time.

It should be noted that risk management itself, as defined here, also constitutes a process. Hence, risk management is also subject to risks. To be more specific, the risks associated to risk management refer to the potential failure of this process in adequately handling the risks of other processes.

2.5. *Risk behaviour*

Handling risks was earlier considered as a fundamental property all intentional human behaviour. Hence it is an intrinsic part of normal management found in organisations. Just like cost, people or time management, it addresses a specific aspect of the total managerial task.

It can be argued that without risk taking behaviour no economic progress would be possible. In a free market situation the only way for a company to survive is to take sound entrepreneurial risks. Obvious examples of such risks are the introduction of new products or the entrance of new markets. So the question is not whether to take risks, but which risks reasonably can and should be taken. The response of an individual or organisation to this question will be referred to as *risk behaviour*. Risk behaviour will be defined as :

the manifest execution of activities with associated risks by an agent.

This definition does not state whether one has been conscious of the risks prior to the execution of the activities. Whether this has been the case will depend on the quality of any risk assessment process (see chapter 4), which has been carried out prior to the initiation of these activities. Apart from accessing the risks, another factor to be considered is what can be gained from certain activities. In general the higher the potential outcomes are valued, the more likely one is to accept a certain risk.

Making this sort of trade-offs successfully assumes a rational process with complete information. As already pointed out in beginning of this book, this is a gross simplification of risk behaviour as commonly encountered in reality. In practice individuals - and companies - differ in their willingness to accept certain risks. This will be referred to as *risk attitude*. Secondly complete information is never available. As a risk relates to a future state, one can never be completely certain beforehand about the attainment of such a state.

In spite of these limitations, risk management should focus on identifying and addressing risks as rational, objective and complete as possible. It can be argued that a great detail of business failure is due to organisations' lack of professionalism in dealing with risks.

2.6. *Risk management as organisational process*

In managing organisations it makes sense to decompose the total managerial task into different sub-tasks, each with its own scope, objects of concern and time span. A commonly used decomposition is based on the distinction between management at strategic, tactical and

operational level, referring to the different types of decisions involved. As a consequence risk management can be treated accordingly at three levels.

At *strategic level* an organisation determines its direction and objectives. Here statements are to be made which guide an organisation in deciding about the degree and types of risks which are acceptable. Trying to avoid every possible risk is the best guarantee for going bankrupt. However, an organisation should not be a casino either. Therefore 'risk' should be a topic to address explicitly as part of the business strategy development process, e.g. in deciding about new products or markets to explore. A conscious choice should be made about the types and degrees of risks the organisation is willing to take in relation to accomplishing its mission. This results in a certain risk behaviour.

At *tactical level* decisions are made about the means to be deployed for attaining the objectives. At this level the processes should be established to support an organisation in systematically gathering experience and learning, which enables the organisation to improve its processes at operational level and to deploy its resources in a better way. For risk management this implies an evaluation of its past performance and effectiveness of the exercised operational risk management activities.

At this level also risks are handled which can not be covered at the operational level. For instance a decision about the scope of a project should be considered at tactical level. Also exception handling - e.g. when more resources are required to reach a certain deadline - involving a reconsideration of earlier agreed priorities needs to be covered by the tactical level.

At *operational level* the designed processes are carried out. Here risk management is performed on an operational level, using the guidelines and facilities outlined at the higher levels. The possibility should be "built-in" at this level to evaluate the processes from a risk management perspective, in a similar way as processes are being evaluated against their budget and planning. This also implies providing the mechanisms for early warning and delegating issues to a higher level for resolution, when they are beyond the control scope of the operational level. The generic objective at this level is to ensure that the processes are being executed in conformance within the boundaries of stated constraints, while producing the required outputs.

In general what is described here is a three tier hierarchical control model. In this model at each level a specific control process occurs, with its own domain of control possibilities, reflected in the specific risk management issues to be dealt with.
Figure 2.3 shows the hierarchical structure schematically and how the various levels relate to each other.

Considering this model, it makes sense for the following reasons to apply it to risk management as well :

- decisions made at a certain level establish the constraints for processes at subordinate levels. This implies that a decision made at a higher level may incur certain risks for the lower level(s);
- at a higher level it may be decided quite consciously to start certain activities at lower level which are of high risk, e.g. prototyping activities in order to exploit the possibilities of a new technology;
- lower level activities may result in output deviations of such magnitude, that they do not only violate the objectives stated at this level, but even endanger the company's outlined strategies;
- considering the differences in types of decisions and time horizons involved with each of the levels, risk sources should be identified specifically to each of the levels and associated with specific types of measures, as opposed to using an undifferentiated model for each level.

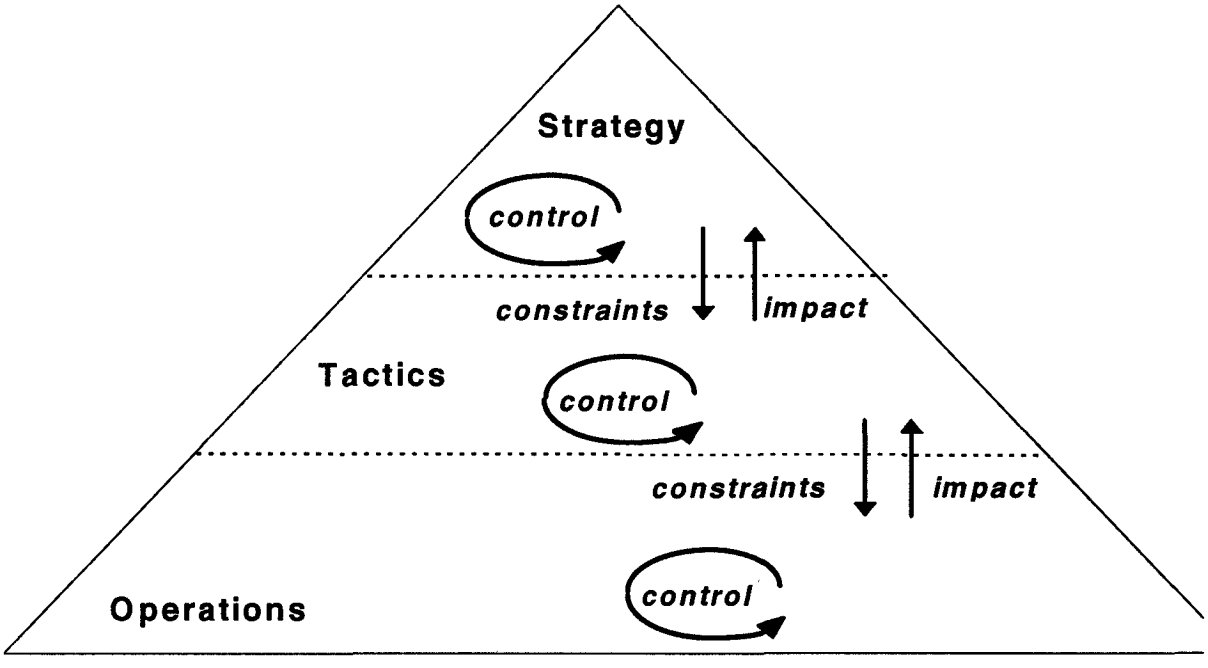

Figure 2.3: Hierarchical management model.

In summary, risk management needs to be specified for each of the three managerial levels and can not just be considered for an individual level, in isolation from the others.

Although the focus of this book is on risk management at operational level, also the relationship with the tactical level will be dealt with. We consider feed-back from operational project results to the tactical level as an essential requirement for improving a company's

capabilities for handling risks in a structural way. When preparing a project, it is therefore a requirement to explicitly build-in the necessary support for the tactical level (e.g. procedures for recording measures taken to reduce risks, for recording the evaluation outcomes of those measures).

In practice this relation is often neglected, every project is approached as a unique event which should be organised from scratch. Rather than attempting to take advantage collectively from previous experiences, one simply tries to select the most experienced and available individuals as participants for the project, hoping that this will provide sufficient guarantee for the project to realise its targets. Once the project is over, one tries to forget the failures as soon as possible, implicitly assuming that next time somehow it will be better. In this situation it is up to the personal initiative and capabilities of the individuals to process their experiences.

Apart from the institutionalisation of the learning process, the relationship with the tactual level is vital in arriving at a project definition beforehand (e.g. scope, objectives, available resources), which is realistic from - amongst others - an operational risk management perspective.

It is at the tactical level where one decides on the project's contents, based on factors like budget, priorities, resource availability and also more abstract factors like the organisation's readiness for change and the perceived complexity of the project. This usually requires some form of compromising between the business priorities, tight in to the strategic level, and the organisation's capabilities of handling the required changes. Splitting up the initial scope into a number of smaller projects, which are planned sequentially in phases, usually is the outcome of such a trade-off and provides a specific way of handling the risks for the project as a whole.

So far the focus has been on risk management in its entirety, occasionally hinting at some of its elementary components. In the next chapter the various elements of 'risk management' will be explored more explicitly and in detail, thereby providing a first stepping stone to the operational approach as outlined in chapter 6.

3. RISK MANAGEMENT ACTIVITIES

In the previous chapter 'risk management' was identified as a specific type of process and it was positioned in a wider organisational context. It was advocated that risk management should be carried out consciously in order to achieve the desired result.
Obviously in carrying out risk management, one can be successful to various degrees. Hence the question arises; how can risk management be engineered, to ensure beforehand both its efficiency and effectiveness ? To answer this question the various types of activities collectively making up risk management are explored in this chapter.

3.1. Introduction

As stated earlier, risk management is a process focused on the attainment of certain goals. In general, goal seeking behaviour - considered from a combined economic and rational perspective - assumes the presence of a number of different types of activities, to be carried out in a certain sequence. This typically covers the following; defining the goals to be pursued, identifying alternative courses of action for reaching the goals, an evaluation in order to rank the alternatives on a goal related preference scale, taking a decision, implementing the selected alternative, measurement of the effects and again an evaluation activity to determine the effect of the activated choice relative to the stated goal.

The execution of these activities may need to be carried out repetitively, in order to approximate the goal sufficiently in a number of steps or iterations. One can say that a process is being controlled - as opposed to erratic behaviour - when its goals are being converged upon by going through a limited number of iterations.

The *generic goal* of risk management is to reduce the likelihood of certain risks by taking appropriate measures. Stated differently, the objective of risk management is to permit certain activities and the control process of those activities to obtain their outputs with reasonable degree of certainty. What 'reasonable' implies is determined by an evaluation of the risk management effort to be deployed and its contribution to reducing the risk impact. As alternative courses of action are available, this requires a trade-off between effort and impact reduction by projecting both on a preference scale, based on a common denominator (e.g. money).

Assuming the applicability of the generic risk management objectives, operational risk management will be considered to exist of the following elementary activities :

- *identify* risks; which specific risks apply to the activities to be performed ?;

- *analyse* risks; what are the underlying potential causes of the identified risks (i.e. the risk sources) and their mutual relationships ?;
- *prioritise* risks; what is the exposure of identified risks in relation to the objectives stated for the activities ?;
- *conceive* alternative actions; which alternative actions are available for obtaining a satisfactory situation and what is their anticipated effect ?;
- *choose* and implement actions; decide on the desired actions and influencing the setting of the involved activities by carrying out the selected actions;
- *monitor* effect; determine the nature and magnitude of the effect of implemented actions.

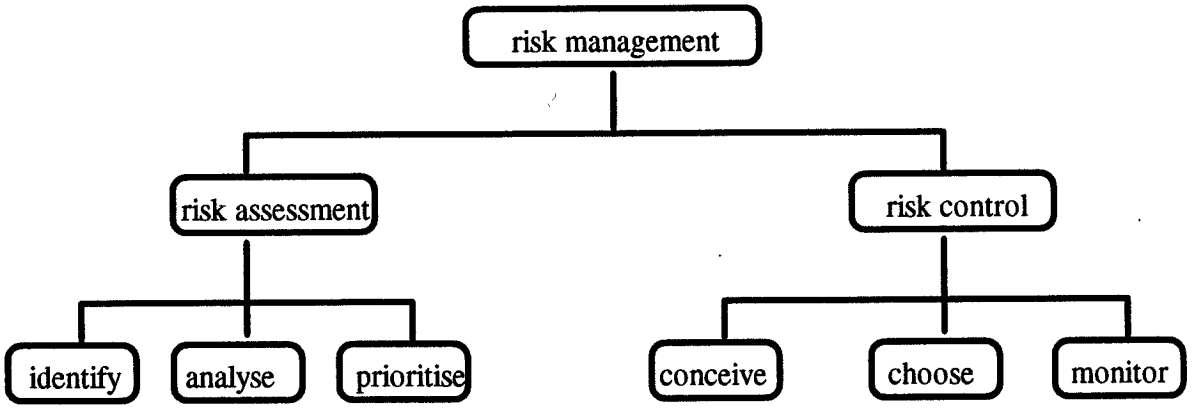

Figure 3.1: Risk management activities.

The first three activities are often collectively referred to as 'risk assessment', while the last three are called 'risk control' (see Boehm, 1989). Figure 3.1 schematically shows the main elements of risk management.

The process of risk management is a perpetual task. As argued, the management of an individual activity may require multiple iterations before its output is obtained. But apart from this, the focus of risk management shifts from one activity to another as the process materialises. As long as a process has not come to an end, finalising of activities results in the need for new activities to be started, which build upon the results from the previous ones.

This requires a virtually continuous *assessment* of the environment in which the process is taking place, to determine the presence and potential impact of risk sources. Every activity has its specific initial state, which is partially determined by previously obtained outputs, but also incorporates inputs and controls coming from outside the project. At the start of a project imperfect knowledge is available about the relevant environment for all activities and how it

will react on the project after its has begun. Changes like users having second thoughts about requirements, an alteration of the business strategy affecting the priority of the project, a sudden illness of a key participant, a new manager as problem owner, are just a few of the unpredictable phenomena which can heavily affect an ongoing project and require a redefinition of its direction.

Key to risk management is to identify these potential disturbances at an early stage, in order to prepare, decide about and implement the selected measures. These activities cover the *control* part of risk management.

At the end of a project, a separate risk management evaluation should be planned, to assess the results of deployed risk management activities. This evaluation is intended to support learning activities from which subsequent projects may benefit. By evaluating the quality of the process, one can improve the processes of risk assessment and risk control as such, resulting in turn in a reduction of risks in the activities to be managed.

For instance by evaluating a completed software development project, one can come to the conclusion that an unfortunate choice of development tools was made. While analysing the causes of a project delay, the conclusion is drawn that none of the team members had prior experience with the selected tools. The tools were selected based on technological criteria only, mainly because they represented the state-of-the-art in this area and the IT manager wanted to make a quantum leap forward. The consequence was a lower productivity than planned and many errors made, which caused the delay in delivery of the new system.

To avoid this, in future projects one has to pay additional attention in deciding about the tools. Checking for available experience within the team is a criterion to be incorporated in evaluating alternatives. This awareness could lead next time to selecting those tools which most team members are familiar with and which have a stable performance. This choice will present a less risky option.

The evaluation of risk management is carried out by explicitly investigating the effects of deployed risk assessment and risk control activities. In order to do this, a proper recording should be in place throughout the project of all risk management activities carried out. One should be able to completely reconstruct the perceived decision making situation, including the rejected alternatives and the reasons used for doing so. Apart from this, a systematic logging of project data also has to be carried out to support the operational risk management activities as such. To avoid unnecessary efforts, one should be able to tell, as part of operational risk management, whether risk management activities are truly paying off.

Risk management evaluation is part of risk management at tactics level, but should have its roots firmly within the operations level. By considering the reasons for success of failure of projects, one can take measures to create more favourable initial states for future projects.

In the next sections the various operational risk management activities are explored in more detail. The alternative techniques which are available to support the different steps in risk management are not addressed here, as these are sufficiently covered by available literature (see amongst others Boehm, 1989).

3.2. Identifying risks

The identification of risks is essentially a matter of conceiving anything that realistically can go wrong with an activity. Considering the specific case of a software development project and its inherent complexity, the range of events possible which theoretically may disturb its execution is endless. As a consequence, in identifying risks one has to be selective. Based on a combination of up-to-date information about a project's environment - being its initial state - and experience with similar types of software development, a selection is made to determine whether a certain risk is realistic enough to spend the effort in exploring it further.

Also in our daily lives there are infinite risks, some of them being real, while most are purely theoretical. For instance the roof of our home may collapse, we may be struck by lightening when we go out, a rabies infected dog may attack us when shopping down town, etc. A person normally does not bother about these rather theoretical risks.

However, other risks are less hypothetical, as we have learned. Before crossing a busy road a normal individual would halt to check the traffic. This typically is a routine activity, driven by experience, which tells us that a real danger exists in crossing the street carelessly. It is presumably this implicit awareness of what constitutes a significant risk and what can be ignored, which allows us to carry out are daily practices with reasonable degree of efficiency and self assurance.

The danger of this step is in forgetting certain significant risks, being over-optimistic, or simply making a wrong estimate of the risk impact. The way to reduce this danger, is to make the risk identification activity more explicitly. Involving a group of people - rather than allowing it to be the task of a single individual - can have two positive effects. At the first place a group collectively has more experience than a single person. Secondly, in a group one has to make ideas, feelings, opinions explicit in order to discuss them within the group.

To make the right choices in the context of software development, different types of experience are required to enhance the likeliness that the right risks are brought on the table. This involves experience with the field of software engineering as such, but also general project management experience and familiarity with the involved users' organisation provide necessary ingredients.

Identifying potential risks requires a different attitude than in normal project planning found. Instead of addressing what needs to be done - a positive attitude - one should be able to adopt a critical or negative attitude to really address what can go wrong. For certain individuals this may present a fundamental conflict with their own personal management style.

In order to identify risks, it is necessary that a definition of the activity's intended output is given as a basis for reference. This definition should unambiguously state the intended outcomes of the activity.

3.3. Analysing risks

The authors' approach to risk analysis is based on the idea that the fundamental cause of a risk can and has to be traced down. It is only by identifying this factor and next by manipulating it, that one can exercise risk management in a truly pro-active way. Focusing on risks as opposed to the risk sources would essentially imply addressing the symptoms rather than the real causes.

The underlying factor of a risk was referred to as 'risk source' (see section 2.2). A risk source refers directly to the entity which can be managed and which imposes one or several risks. For instance a lack of expertise of a user may result in wrong statement of requirements, may cause delays in coding due to changes, or may even provide a road-block in the final acceptance of the system by its users.

The leverage of a risk source can be quite significant in terms of the number of different risks it may cause. Another complicating factor is that these risks may not become manifest before a significant period of time has elapsed. This stresses again the point why the focus in effective risk management should be on the real sources of the risks.

To determine the relevance of risk sources, one should consider cause-effect type of relationships. Considering the model presented in the previous chapter, this implies understanding the general processes which allow a risk source to result in the undesired effect. To understand what harm can be done by an inexperienced user, one has to understand the basic process of information analysis, how this affects program specifications, subsequent coding activities, testing and finally the hand-over to the users. It is through this chain of activities that the particular risk source will exert its negative influence.

In practice the identification and analysis of risk can go hand in hand, being carried out in an iterative mode. One may start by initially spotting a particular risk source as significant and then, by analysing its associated risks, determine whether it is worth while taking measures. Or one identifies the potential risks first and next tracks down their risk sources, to establish their likeliness of causing the identified negative effects.

In the example of the requirements analysis, there is the general risk - as experience shows - that one ends up with incorrect requirements. Within the specific organisational context one has to screen the users, as risk sources, to establish the significance of this risk. On the other hand one may identify a low level of experience as such and then analyse the potential consequences of this property, thereby establishing the risks associated with the risk source.

3.4. Prioritising risks

This is the quantitative aspect of risk management. Having identified the significant risks and risk sources, their potential negative effect on the desired outcomes has to be quantified. This was earlier - in section 2.1 - defined as 'risk exposure', being a function of 'risk impact' and 'risk probability'. Quantifying risks is a matter of estimating the risk impact relative to a preference scale and determining the probability that the risk will materialise.

For instance, for a software development project it is likely that the plan will be over-run. Further analysis shows that it is quite likely - more than 90 % - that this will be more than one month. However, one is fairly certain that the delay will be no more than 3 months. As such these data do not yet tell what the real risk exposure is. To determine the gravity of the estimated delay further analysis is necessary of the impact on the project's objectives. Perhaps the project is not on a critical path and some delay is acceptable, or are other activities dependent on the success of the project and should delays be avoided by all possible means ? These considerations result in a weighing factor, expressed relative to a chosen preference scale (e.g. 6 out of a 10 points scale, or 'significant risk exposure' being one of the values of a scale ranging from 'risk exposure should be absolutely avoided' to 'insignificant or none risk exposure').

The example shows that the quantification of risk exposure is a matter of :

- quantifying the negative effect of the risk;
- assigning a weighing factor to this effect according to a preference scale;
- estimating the probability of the risk.

The second factor essentially is the risk impact. Having quantified probability and impact, their mathematical product can next be taken to simply determine the risk exposure.

Why should one attempt to quantify risks ? There are two main reasons for this. First, it is another vehicle for making the risk management process as formal and therefore as objective as reasonably possible. Secondly, in any situation a variety of risks may have been identified. As limited time and means usually are available, it makes sense to rank the risks and to focus on the severest ones first. The latter implies that in assigning weighing factors the same

preference scale is used, to be able to rank the risks relatively to each other. This ranking of identified risks is the actual risk prioritisation activity.

One should take care however that quantifying risks does not become an objective in itself. Values and scales - not necessarily quantified in terms of concrete figures - should be used in a meaningful way and need to be in line with the experience and sophistication of the team applying them.

Estimating risk probability and impact is not an easy job to do. Different approaches have been identified for estimating risks, some more quantitative than others. Techniques which can be used to arrive at a quantitative estimate usually are based on preference scales and associated procedures (for groups) to arrive at a common judgement about the ranking of the risk (e.g. based on a Delphi approach or questionnaires).

The subjective nature of this step can never be eliminated completely. It is therefore especially important to approach this activity in a structured way. This implies attempting to visualise also the subjective aspects, to make them explicit and debatable by the team involved with the risk analysis.

3.5. Conceiving actions

This is closely related to the identification of risk sources. Once a source has been identified, one can define the agent(s) being responsible for managing it and set out alternative actions for dealing with it. There are four basic strategies available for dealing with an identified risk source :

- *avoidance*; simply do not start the activities through which the risk source may assert its influence in terms of materialising risks;
- *reduction*; take appropriate measures beforehand, which reduce the risks associated with the resource to acceptable proportions;
- *compensation*; assume that the risk source will have a negative impact on the intended activities, but take measures to rule out or to reduce its negative impact (i.e. the risks) to acceptable proportions;
- *contracting*; assume that the risks will materialise and agree beforehand how to handle these risks, once they have occurred.

In the extreme case one *avoids* the risks associated with a particular decision situation. At first glance this may seem like simply running away. However, there are situations where this is the best possible option.

An example is a definition of a software development project, which is assumed to be frozen by an organisation and not subject for further debate anymore (e.g. for political reasons). When a software service company is invited to bid for such a project and it is obvious to them that this will lead to disaster, a no-bid strategy can very well be the best option for such a company. The consequence of choosing this strategy, is that the original objectives will not be reached, as the required process towards them is simply not carried out.

When choosing for the reduction strategy, one decides to manipulate the identified risk source to eliminate or reduce the possibility that it will result in the associated negative effects. Essentially this involves additional activities, to be completed successfully before the main process is started.

In a software development environment one can think of sending people to a training course to learn a new development tool, prior to starting the project in which this tool is to be applied. This would reduce the risk of delays in the project because of lack of familiarity with the new technical environment. The reduction strategy implies additional investments to be made, which do not have to interfere with the objectives of the project as such.

The compensation strategy focuses on the creation of leeway, which will filter-out the effect that a risk source of an activity can have on the attainment of its primary objectives. Choosing this option usually implies that either it is not possible, or it takes too much effort or time to try to influence the risk source beforehand (*reduction*).

Applying the compensation strategy for a software development project means that measures are taken, like building in addition time slack to compensate delay of an activity, or to put a claim on resources when can be pulled in when additional expertise is necessary to finish on time. The consequences of these additional measures are usually that the planned throughput time of a project becomes longer and its budget becomes higher. However, these effects can be considered as acceptable, when they enhance the likeliness for the project to obtain its primary output, being an operational and accepted system.

In case of the contracting strategy, one has identified the potential risk, but either sees no way to reduce it, or required measures can not be justified from an economic or political point of view. In the case where a software development project is done by a third party service company against fixed price conditions. To be more specific, the contract is carried out by a German company and stated in US Dollars. In this case the service company is liable to a currency risk, which can neither by them, nor by the customer be influenced. A possible measure to take, is to build-up a credit position of similar magnitude in US Dollars, to counter balance the risk at the incoming side. In a way this is an insurance for the risk, taken beyond the scope of the project, not affecting its execution or objectives.

Another type of contracting, is to agree beforehand with a customer about covering the negative effects of identified risks. 'Equal share of additional costs', is a concrete example of such a measure. In general the strategies differ in terms of handling risks pro-actively or re-active and in addressing the source of a risk or handling its effects.

3.6. Choosing actions

The basic strategies were identified which need to be considered in conceiving alternative actions. For a particular decision situation these alternative options are merely an aid in conceiving specific types of actions.

Which strategy and specific action to chose should be considered per risk and will depend on a combination of the following :

- the risk exposure as determined formally, or simply as perceived;
- the decision space consisting of alternative actions one can chose from;
- the expected effect of a particular strategy and action in terms of exposure reduction;
- the effort (costs) associated with a particular strategy and action;
- the risk behaviour in terms of willingness to accept certain risks.

A few examples:

- If somebody lives in an earthquake sensitive area, but is not in the position to move because of employment reasons and the person involved perceives the risk exposure as bearable, he or she may decide to put aside a certain sum of money to cover any potential material damage. This is a form of a *compensation* risk strategy, which provides the only option, as the person will not move and the event of an earthquake can not be influenced.
- If one is considering to go out for a walk in the park, but the darkening sky indicates that a thunder storm is approaching, that person can decide to postpone the intended walk till later. In this example this risk of getting a wet suit is simply *avoided.*
- This strategy can also be applied in the example of the earthquake sensitive area. This would be the applicable to somebody planning to remove, but consciously deciding to *avoid* the critical area in choosing a new home. In this case the decision space does allow for the choice of alternative actions.
- A participant in a Formula 1 race exposes himself to considerable danger. However, it is a conscious decision to accept the risks associated with this type of sport. On the other hand, the driver may have a substantial life insurance to minimise the material consequences of an accident for his family. This is a form of a *contracting* risk strategy.

Of course also combinations of strategies are conceivable, as the next example shows. When somebody formulates the objective to be a day away from the office and to spend it with one's

family, there are various activities that person can do. One way of spending that day is to go to the beach. However, whether this provides the required relaxation depends largely on the weather. If the weather forecast predicts a fair probability of cloudy weather, our family could decide to *avoid* the risk of having a spoiled day an go to a museum instead.

As the weather can not be influenced, *reduction* measures are not available. However, in choosing the particular beach where our family will go to, they may take into account the option of other facilities nearby (like a restaurant) to *compensate* for the lack of joy, due to the eventuality of bad weather. A last option is to agree on the day before, that when it is not sunny tomorrow, everything will be cancelled and they will wait for a better day. By making this *contract* beforehand with the whole family, disappointment can be kept to the minimum.

Basically these types of measures can be considered for any risk source and one type of measure, or a combination of them can be deployed to perform risk management. Returning to the earlier used example of the cumbersome requirements analysis, similar measures can be taken. Assuming that the risk of the inexperienced user has been identified beforehand as significant, one may decide to consult other users instead *(avoidance)*, or other users may be consulted as well to provide a verification of the statements of the inexperienced user *(compensation)*.

Another strategy would be to wait until the user has become more experienced *(reduction)*, but this option most likely would be incompatible with the time scales of the project. The strategy of simply taking the risk *(contracting)* would be unacceptable as well, due to the high risk exposure and unnecessary as other options provide better solutions.

Measures may result in a reduction of risks or of their effects. However, this may be offset by the effort required to carry them through which can not be ignored in selecting a particular course of action. Risks can be reduced significantly, but quite often at great expense and delays.

The idea of eliminating all possible risks is a myth. It is impossible to foresee all possible risks and therefore to take all possible measure enabling an activity to reach its goal with 100% certainty. Any living entity - being an abstract creation like an organisation or a concrete agent like a human being or an animal - has to interact with its environment in order to survive and to grow or evolve to a higher level. In doing so, one has to take "entrepreneurial" risks. These are the risks one has to face in reaching one's objectives.

The essence of risk management therefore is not in attempting to eliminate risks at any expense, but in finding a reasonable balance between pursuance of stated goals as such and the efforts to be made to maximise the likeliness that these goals indeed will be attained. Depending on the adopted risk behaviour, one has to make a conscious decision as to what maximum degree of risk is still acceptable.

Finally there is another aspect to consider. Any additional activity performed to manage risks implies an additional source of risk itself. In an extreme case the medicine could be more harmful than the disease. If one decides for reduction or compensation measures, one should be sure that these measures as such do not induce greater risk than they are intended to eliminate.

3.7. Monitoring effect

When it is decided to choose either a reduction or a compensation type of measure, it is necessary to evaluate whether and when the measure does result in the intended positive effect. Based on the outcomes of the evaluation, one can decide to extend the measure, to stop the effectuated action, or to consider an other one. To be able to monitor the effect of risk management, it is required to formally and frequently record the state of affairs in such a way, that changes - either positive or negative - can be identified.

4. CRITICAL FACTORS IN MANAGING SOFTWARE DEVELOPMENT

In this chapter first a number of premises underlying this risk management approach will be presented. The chapter will conclude with explaining the link between risk management and project management.

4.1. Introduction

So far generic treatment has been given to risk management. In this chapter the focus is on risk management in the specific context of software engineering. At first a number of premises is explored as being fundamental to the nature of software engineering projects and therefore impacting the way risk management should be addressed in this context. These premises can be divided in the following three categories :

I. Premises related to an individual

People participate in projects and bring in their own - sometimes hidden - personal motives, drives, feelings, and perceptions, which strongly determine the way in which they behave themselves. From this perspective the behaviour of an individual may be productive or may be counter-productive relative to the project objectives.

II. Premises related to a group

A software engineering project as considered in this book normally materialises as the outcome of the joint endeavour of a group of people. A group is here considered as a number of individuals, who are supposed to co-operate. Such a group constitutes the project team. Considering the project as a social event, mechanisms and patterns of human co-operation can be identified, which have a strong impact on the outcomes of the project.

III. Premises related to an organisation

A project will normally take place within the boundaries of one or more standing organisations. This organisation will make use of the system which is to be developed by the project team and therefore can be considered as 'customer' of the project team. The project therefore needs to be organised formally to ensure its desired outcomes within reasonable boundaries of likelihood. The formal organisation, in terms of explicitly laid out agreements and rules, acts as an external guideline in directing the operational behaviour of the group of people participating in the project.

In organising a project formally the impact of the first two types of premises needs to be recognised explicitly and should be reflected in the specific set-up of the project.

At this point it should be noted, that it is beyond the scope of this book to give a treatment in full of all the socio-psychological factors which influence a project. For those we refer to the relevant literature in the realms of organisational theory, industrial psychology and sociology. However, based on experience of the authors with software projects, some of the major premises and their impact on this type of project - and therefore on risk management - are explored in the following sections.

4.2. Premises related to an individual

Autonomy of the individual
The participants of a project either seek fulfilment of their own personal ambitions, or they actively contribute to accomplishing the project's objectives. These two modes of action not necessarily have to be in conflict, provided that personal incentives are directly related to the project's goal attainment. Enhancing the intrinsic motivation of a project's participants, combined with open communication amongst everyone, is a way to obtain the level of active participation, as required to deal with contingencies. The effect is an enlarged self-control potential of the project team.

Still too often one attempts to foresee all kinds of eventualities and to treat those by preparing complicated and rigid procedures, or simply by creating a heavy multi-level project management structure. These measures, while neglecting personal motives or qualities, take away all flexibility and result in delays in case of unanticipated events.

Attitude of the professional
Software development requires a high degree of professionalism. Professionals usually value the following aspects in their daily work :

- individualism to fully exploit their creativity and experience;
- excellence and technical superiority in the results of their efforts;
- freedom in the execution of activities, not constrained by formal rules, plans or budgets;
- expecting management to create the conditions under which they can perform optimally.

Often, due to a lack of formal assignment of specification, specialists find themselves in the situation where they have to find solutions for problems which should have been dealt with

by others. This can lead to the situation where the specialist starts working on the basis of personal initiatives and best guesses which might result in project failure. The extreme alternative is that the specialist refuses to co-operate as long as the matter has not been settled properly by project management. This would tend to result in extremely long project lead-times.

A formal organisation of work by means of unambiguous plans and recognised quality levels needs to be available at the start of the project. This should be discussed properly by project management with the individual team members. Open communication, involvement and mutual agreement should be pursued to reconcile the personal drives with the formal role related requirements. This attitude can easily conflict with the type of behaviour required on behalf of the formal role a professional is expected to fulfil within the context of the project (see 4.6).

Attitude towards risks

In conducting risk management effectively, it is vital to tackle risks as early as possible in order to minimise their impact. To do this, it is necessary to understand the underlying mechanisms, which cause certain risks; the "risk sources".

Dealing with the actual sources of risks is different from what is often encountered in daily practice, where one simply tries to compensate for the negative effects of risks in a reactive way. This is merely addressing the symptoms, rather than dealing with risks in a fundamental way. Addressing the actual source provides the best possibilities for handling risks in a pro-active way.

Often the tendency is to avoid facing problems explicitly. The risk probability is sometimes implicitly estimated too low, and one hopes for the best. Also, early symptoms such as delays or lack of user involvement are often ignored in the hope that it will improve over time.

Pro-active treatment of risks requires an attitude where one actively looks for potential problems, rather than avoiding them, the latter still too often being the type of behaviour found in projects.

4.3. Premises related to a group

Joint decision making

Managing risks should not be a matter of a single individual (i.e. the project manager), rather it is a matter which needs to be a concern to everyone involved in the project. In particular at the start of a project it is essential that all parties involved reach a common understanding about the risks which are involved and on a project plan reflecting this common opinion.

Research shows that the decision quality of groups exceeds the ones of individuals in specific circumstances. Those circumstances relate to the types of decision, knowledge and experience of the persons concerned. Especially complex decisions with many variables, far-reaching consequences, a lot of uncertainty, taken in a professional setting, where required experience or knowledge is fragmented over the participants, are pre-eminently suitable to be prepared through a process of group decision taking. Risk management within the context of a software project meets these characteristics.

Collective decision making can outperform the decision making processes of individuals for various reasons, which also apply to a risk management process :

- the amount and types of information to be supplied and processed are too much for one person;
- sharing ideas within a group can be the away to avoid or moderate the impact of personal bias;
- participation in the decision making process stimulates personal commitment to the decision;
- to the participants it better makes clear the overall context of a project and how their individual opinions relate to the overall context.

However, in practice it is still too often left to the personnel judgement of the project manager what risks apply to a particular project and how to deal with these.

Groups as self-contained entities

When a number of people within the context work together to achieve common goals - which all participants commit themselves to - the danger exists that after some time these goals and the agreed ways for accomplishing them become a fixation in the minds of everyone. In that case the group is no longer open for influences from outside, or for questioning or even

criticising the way in which the group proceeds. Convinced about their own ideas, the participants tend to deny observations which are in conflict with their own ideas. Even when these observations are raised by a group member, there is a chance that the others will reject this. As a result, the individual raising the issue may become an outsider, affecting his credibility to the other members of the team.

A frequent and open communication between the user organisation and IT management is a necessity for reducing the likelihood of this effect. Also the involvement of a risk advisor, external to the project team, at critical points in time, is considered as a means of avoiding this danger. Such a specialist should be informed on the progress of the project on a frequent basis in order to allow him to intervene at moments he feels appropriate.

Win-win attitude
A software project materialises as the joint effort of a group of individuals. Although each individual brings in its own perception and motives, it is essential for the success of the project, that a common understanding and agreement is reached beforehand about the results to be obtained. During the project this shared value should guide all participants in their daily activities and should provide an unambiguous base for conflict resolution.

Instead of opposing each other when a problem arises, the basic attitude should be aimed at solving problems together in the best interest of all parties involved. Handling problems in a antagonistic way will only result in a loss for all, although this may seem different at first glance.

In many projects, when a difference of opinion is encountered at some point in time, this automatically becomes a conflict where the different parties oppose against each other. After this has happened once, the next problems are is likely to escalate as soon as they occur and may even lead to an end of the project.

4.4. Premises related to an organisation

Formalised roles
As stated, a software project involves a number of individuals. These individuals together have to cover different formal roles in order to make the project succeed. This is by no means a trivial matter. A proper understanding and agreement on the assignments to the various individuals in terms of tasks, responsibilities and deliverables is part of preparing a project. A careful definition and communication of roles provides the basics for efficient co-operation

(i.e. team work) and procedures for problem resolution. This applies both to the participants of a project team and to the people involved with the project on the outside (e.g. user management, service provider management).

In practice proper demarcation of roles is often neglected. This may cause severe delays or disputes when unforeseen circumstances appear. For instance in such a case it may turn out that no proper mechanisms are foreseen to raise additional budgets, or to decide about trimming secondary requirements. The type of roles required in a project are explored in section 4.6.

Uniqueness of the environment

Risk management in practice does not stand on its own, but should be applied in an integral way with the general project management and project phasing methodologies as chosen by a particular company. The specific way of applying risk management should also be in tune with organisation specific factors, like culture and values and should also be in line with the characteristics of the specific project.

Blindfolded applying a standard approach according to the book does not work, in particular not in the field of risk management, as subjective, non-formal and environment specific factors play such an essential role.

In practice too often one clings to methodologies, techniques, or tools, which are simply used to keep one busy, while avoiding critical issues. This is the best guarantee for not solving any real business problem.

Organisational learning

An organisation usually is involved in multiple projects at a single moment in time. This may even apply to the IT area. Although each project should be addressed as a unique event, certain factors are re-occurring, like the people involved or the approach to be applied. From an improvement objective point of view, it is therefore not only possible but even required that organisations draw their conclusions from past behaviour and use those conclusions to improve future actions. This implies in particular to risk management, where future risks can be avoided or minimised by learning from the past.

A prerequisite for this type of learning is the systematic recording and availability of factual data and evaluation results from previously performed projects. Such an information base should be developed and maintained by the organisation itself, as outside data may be

difficult to compare or to interpret. Maintaining such a project file should be organised as part of the normal project activities and should be done on a frequent base. This implies frequent evaluation of the state of affairs, drawing conclusions about the effects of past events, deciding on any adjustments in the project's course and documenting this all in such a way that it can easily be accessed and understood at any future point in time.

In practice too often only limited data recording is done in an ongoing project situation. Quite often it only concerns planning and budget related data (i.e. actual hours spent on activities), official minutes of management meetings and the private notes from the project manager, which usually disappear once the project is over.

Projects as contracts between parties

As a group of people is involved in a project for some period of time, who are supposed to co-operate during that period, ideally a common understanding needs to be reached amongst all participants beforehand about the nature of the project, its objectives, outcomes, activities to be carried out and by whom, constraints, etc. This is much more than just organising the project formally by defining its budget, plan and management structure.

The term *'contract'* shall be used to refer to this common agreement about the project, which needs to be prepared beforehand between the project's participants and it is assumed that such an agreement can only be completed in full after every participant's formal and personal objectives have been fully explored.

Project risks can now be defined relative to such a contract as the possibility that certain deviations between the actual outcomes and a defined contract occur.

The necessity to make a contract seems more obvious when external contractors are engaged to carry out part of the project's activities. However, even in the case where a project is carried out by an internal automation group, an explicit consideration amongst all concerned parties and an explicit statement of the project's objectives and means to be deployed is required. In particular in the case of an "internal" project this is often neglected.

The 'formal roles' premises is explored in section 4.6, where a generic description is given of the various formal roles, which can be distinguished. From a risk management perspective, it is essential to identify these roles, as involvement in risks management should be an essential element of the formal description of each role.

As the identified roles are related to the project context of software development, first some considerations are given in the next section about the relationship between project management and risk management.

4.5. The relationship between risk management and project management

A fairly common description of a project will be used in this book :

> *a project is an organised set of activities of temporary nature, aiming at the accomplishment of defined goals, which are of a unique nature.*

As such every project is a unique happening, since it is supposed to result in an outcome which is one of a kind. More specifically, what makes a project 'unique' is that the way in which the outcome is to be obtained is of a non-trivial nature. Consequently, the activities of a project need to be selected, specified, planned, and budgeted explicitly. Since a project is of non-trivial nature it can never be predicted completely and depends to a large extent on what's being encountered along the road.

However, when considering a project's individual constituent activities and tools a high degree of commonality and repetitiveness can be found. The uniqueness of the project arises from the way in which the individual activities are combined into processes. Hence, the challenge of managing a project is in arranging and co-ordinating those activities. It is in this process of composing a project where the management of risk is an indispensable element. Consequently, risk management is an integral part of project management.

A way to consider a project is by examining its individual dimensions. This concerns for example 'time', 'money', 'capacity', 'quality', but also 'risk'. This implies two different views, to be applied alternating within the context of project management :

- from time to time one needs to consider the aspect individually and on its own. Just like one controls a project's budget or plan, so should risks be controlled;
- however in taking decisions, one can not consider a single aspect in isolation. A trade-off is made in taking decisions, involving multiple aspects at a time. For example spending more time on analysis could result in higher quality, but also in postponement of the systems' delivery.

In this view project management is very much like a juggler's act. The balls are handled one by one, but the total set is required to really make the act complete. In this metaphor a focus on managing risks only is like keeping just one of the balls up in the air.

Returning to our description of a project and emphasising the element of uniqueness, it can be stated that risks are inherent to any project. If there would *not* be any significant risk, one would obviously be looking at routine activities, resulting in specific outcomes with a large degree of certainty that no special attention would be required from a risk management point of view.

Hence it can be stated, that risk management is an essential activity within the context of a project, discriminating project management from more routine type of control processes.

4.6. The formal roles in relation to a software project

How people contribute in preparing a project's contract and participate in the ongoing project, depends on the formal *roles* they fulfil and their own personal motives and capabilities. In this section and the remainder of this paper we constrain ourselves to the formal roles, as these should provide the guideline in selecting the specific resources for the project. In describing these roles, a customer-supplier type of relationship is assumed to exist between the user organisation and the organisation charged with the development of the system. The latter can be part of the same company as the user organisation or it may be an outside IT service provider. A mixture of both options is possible as well.

The following roles can be identified and described in a generic way.

- *Users*, they who will make direct use of the software to be developed, by applying it in carrying out their daily activities.

  Main responsibility: formulating the requirements which the new software has to satisfy.

- *User Management*, the line management of the involved users, who may or may not make use themselves directly of the system to be developed.

  Main responsibility: ensuring validity of requirements as formulated by the users.

- *Sponsor*, the person who (usually on behalf of his formal job title in an organisation) takes the decision to start the project, provides the necessary funds for the project and is responsible for committing his organisation to the formal contract as agreed upon with the IT development organisation. The Sponsor may be the organisation's general manager, the financial manager, or a discipline manager for whom the project is to be carried out.

  Main responsibility: agreeing initially on the formal contract and providing the means for the project to take place as agreed upon in the contract.

- *Contract manager*, the person who commits his organisation (i.e. the supplier) for carrying out the project against the agreed contractual conditions.

  Main responsibility: formal agreement of the project's contract on behalf of the supplier, organising and supervising the participation of his organisation to the project in such a way that it will be carried out against the formal arrangements of the agreed contract.

- *Project Manager*, the person in charge of managing all operational activities carried out under the umbrella of the project's contract. This role can be assigned to a representative of either the customer or the supplier. The main responsibilities of this role include:

  - customer; ensuring availability of resources at his side to enable the project to progress as planned. This may include arranging availability of Users in co-operation with User Management.
  - supplier; arranging and managing the resources as contractually agreed and for enabling project execution as planned.

- *IT Staff*, the Information Technology specialists who are charged with designing and building the software. A mixture of IT staff from customer and supplier can be deployed, depending on availability and the customer's preferences.

  Main responsibility: carrying out the foreseen design and realisation activities according to agreed quality standards as constrained by the project plan, which will result in the required software.

Figure 4.1 schematically shows the different roles. In this figure a distinction is made between the operational Project Team, being charged with carrying out the activities in the

ongoing project situation, and a Steering Group which is responsible for agreeing upon the formal contract and supervising the project during its life-time.

The Project Managers together act as linking pin and - next to their managerial activities - can also be involved in operational tasks, like preparing an initial design, which amongst others depends upon the size of the project and availability of the project manager.

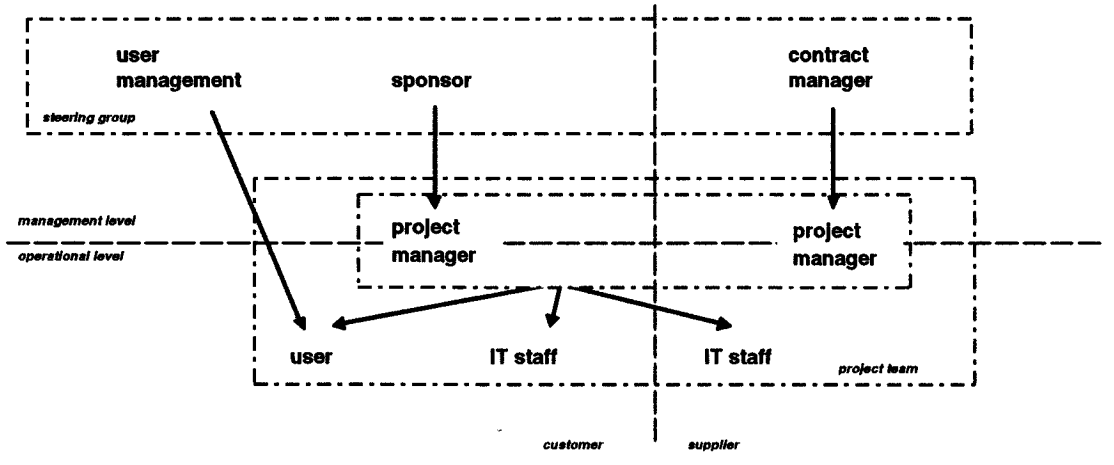

Figure 4.1: Relationships between roles

Every role will be occupied by one or more individuals with their own personal objectives and capabilities. Discrepancies may arise between the formal roles, which these individuals are supposed to fulfil - as derived from the project's contract - and their personal capabilities or objectives. As such this represents a source of risk. In structuring the project and choosing the occupants of the various roles, attention should be paid to synchronise personal objectives and expectations with the ones of the project.

Up till now 'roles' have been identified from the perspective of a formal project organisation. Even if one succeeds to find the perfect match between these roles and the characteristics of the individuals occupying these roles, this will not be sufficient to create a successful team.

Considering the project team from a group dynamics perspective, other types of roles can be identified. From this point of view a team needs to have a clear leader, innovative personalities should be part of the team, and it should have a sufficient number of 'followers' to get the job done. In creating a project team the right mixture of these role should be present. The specific mixture of these 'informal' roles largely determînes the natural tendency of a group to behave in a certain way.

The project should be organised in such a way that the formal role structure is in line with the informal one. For example, the most logical thing to do is to is to select a project manager which can also operate as the 'leader' on basis of his personality skills (e.g. personal authority).

To some extent different roles may be combined - on either customer or supplier's side - however this should be done with care, as conflicts of interest may be built into the project's organisation.

As the involvement of the various roles differs in the course of a project and even before the project is started, so will their contribution to risk management vary. This will be explored in chapter 5.

5. EXPLORING THE SOURCES OF RISK

In this chapter an operational tool is presented which can be used when exploring risks. The tool provides support to the process of identifying sources of risk and assessing to what extent they can have an impact on a specific project.

5.1. Introduction

A key activity in risk management is the identification of those risks that might influence the project on hand. In this chapter an operational tool is developed that can support the process of risk identification. The tool takes the form of a checklist. The next sections will be used to develop the risk sources used in the checklist. First a taxonomy describing the categories of risk sources is developed. Based on this taxonomy the checklist will be developed next. In order to be able to use the checklist a structured way for each risk source a number of descriptive attributes is distinguished. Finally we will look at means for assessing these risk sources. The chapter will conclude with a short discussion of the assumptions behind the approach chosen in this chapter. A complete coverage of the proposed checklist is presented in the appendix.

5.2. Categories of risk sources

The checklist presented later in this chapter is based on a higher level taxonomy of risk source categories. This taxonomy will be presented first in this section. It will be based on the cybernetic and organisational viewpoints that were already considered in the previous chapters in exploring the nature of risk management. Of course, many different taxonomies are conceivable, each with its own merits and drawbacks. The choice made in this book was based on the personal experiences of the authors who found it to be a useful yardstick.

The taxonomy described below as such can easily be demonstrated to be 'complete' in the sense that any risk source conceivable will belong to one of the categories presented here. This however does not apply to the actual list of risk sources (the checklist) as presented in the next sections. Although the most frequently encountered risk sources were identified, there are always specific cases which can not be foreseen completely by a generic tool.

In this respect the taxonomy presented here can be of assistance in the actual use of the checklist. By addressing every new situation at first at the level of the taxonomy by identifying the specific risk sources per category one can avoid the risk of using the checklist as a prescription. This is explored further in section 5.5.

In order to achieve the required coverage of risk sources the taxonomy is built up by combining several aspects which characterise software development. Software development is labour intensive. The main production input is delivered by the people involved. Also many problems and uncertainties are caused by behaviour and/or knowledge of the people involved in the development project. Therefore a first distinction is made from an organisation perspective in:

- human risk source area. In this area we range those risk sources that exist because the main production facility and source of information consists of human professionals
- non-human risk source area. This area contains risk sources which are not directly related to the human factor.

Another way of looking at the field of software engineering is by assuming a simple input, mechanism, control model. This cybernetics model was first presented in chapter 2. In this model the project is considered as a single process. This results in:

- Inputs are all those 'ingredients' for the process, which are transformed into outputs during project execution and which themselves have changed, or even have ceased to exist, after the process has ended.
- Mechanisms are the means that are needed to carry out the project activities but which remain unchanged after the conclusion of the project.
- Controls cover those factors that influence the way in which the process is executed, that is: formal project objectives, personal motives of people involved and constraints which should be considered as given boundaries for the process.

Finally, the utilisation of a product normally is outside the influence (and in most cases the concern) of the project participants. However, it is quite feasible that actions within the scope of the project, or a lack of actions, will influence systems operations. Therefore, from a life cycle perspective a distinction is made between:

- risk sources that have an impact on the project itself (the project phase) and

- risk sources that play a part after the system has been taken into active use (the operations phase).

We defined three views on the field of software engineering which each give a relevant way of characterising risk sources. When combining these three views we get in principle 2x3x2=12 risk source areas as is shown in table 5.1.

Table 5.1: Risk source area taxonomy: a first draft

life cycle perspective	organisation perspective	cybernetics perspective	category #
project	human	input	1
		mechanism	2
		control	3
	non-human	input	4
		mechanism	5
		control	6
operations	human	input	7
		mechanism	8
		control	9
	non-human	input	10
		mechanism	11
		control	12

However, in the area that focuses on the main risk sources that can be associated with the operations phase of the system there are many more risk sources imaginable than those that will be presented here. Since the objective of this book limits itself to the development effort this risk source area will not be looked at closely. The area is included only because the type of problems represented here are typical of what can be expected and give at least an indication of what to look for when stepping outside project boundaries. Therefore, this category will be looked at as a single group.

Also, the category human/input has little or no meaning since (apart from some unfortunate burn-out cases) no human resources are consumed during systems development. This results in the final taxonomy which is represented in table 5.2.

Table 5.2: risk source area taxonomy: final selection

life cycle perspective	organisation perspective	cybernetics perspective	category #
project	human	mechanism	1
		control	2
	non-human	input	3
		mechanism	4
		control	5
operations			6

5.3. The checklist

The selection of risk source areas described in the previous section was used as a framework (taxonomy) within which the risk sources of the checklist were positioned. Table 5.3 presents an overview of these risk sources. This selection of risk sources was derived after a lengthy iterative process in which theoretical reflection combined with the results of a literature search as well as extensive practical experience gained in a large number of different environments (government as well as non-government and for-profit as well as non-profit) both played a major role.

In order to be able to use these risk sources in a structured way for each risk source a number of attributes will be distinguished:

- description
- example
- extremes
- questions
- relevant role
- phase
- responsibility
- examples of measures

Each of these attributes will be looked at below. Apart from a further explanation of each of these attributes also attention will be paid to the way in which the checklist can be adapted to local circumstances. A full description of the checklist can be found in the appendix.

Table 5.3: risk sources

1	HUMAN RISK SOURCE AREAS: CONTROL
1.1	Position
	The formal and effective authority of the sponsor within his organisation.
1.2	Commitment
	Commitment indicates readiness to action and willingness to initiate the actions which are expected within the framework of the project.
1.3	Organisation
	Characteristics of the organisational context within which the system will have to function which might influence the project. These characteristics exist independent of the proposed system. Differences in culture or in sophistication will be of importance here together with the existence of a (mis-)match between the type of organisation and the type of project.
1.4	Team composition
	The composition of a project team determines its ability and the motivation of its members to solve problems in an effective and efficient way within the framework of the agreements (planning, budget and formerly defined results). The project team ideally has: • the 'right' mix of personality types • one goal to which all participants subscribe
1.5	Management of decisions regarding the project
	The quality of the decision making process of management not directly involved in the operational activities (i.e. the user management, sponsor, contract manager roles, which have an impact on the project's progress and the problem resolution potential.
2	HUMAN RISK SOURCE AREAS: MECHANISM
2.1	IT-knowledge and experience
	Has the right type of knowledge and experience on relevant IT-areas been incorporated into the project organisation. Is this knowledge available in principle.
2.2	Domain area knowledge and experience
	Has the right type of knowledge and experience on the subject matter of the area to be supported by the new system (user-domain) been incorporated into the project organisation. Is this knowledge available in principle.
2.3	Availability
	Is sufficient knowledgeable staff of the required role available at the moment they are necessary to support execution of the project. Is this expertise available at the right moment.
2.4	Organisational support
	The degree to which the organisation as a whole is likely to support the project.

3	PROJECT RISK SOURCE AREAS: INPUT
3.1	Clarity of specifications
	The degree to which to specifications are unambiguously defined and accepted by the parties involved
3.2	Stability of specifications
	The degree to which previously agreed specifications are likely to shift during project execution
3.3	Complexity
	The degree of complexity is the degree in which user and/or supplier see the proposed system and the associated project as complex. Complexity as such is thus defined on a subjective basis as related to the experience of those who are involved in the project. The perceived complexity for an individual can be considered as a function of the number of factors and their mutual relationships experienced in a particular decision making situation. In a software engineering environment typical examples of these factors are the number of different departments involved in developing the system, the number of programs or interfaces to be developed, size of the project team in terms of numbers of people.
3.4	Degree of innovation
	Degree to which the systems design incorporates functions or other requirements that are new either in an absolute sense or in relation to the experience of the people involved in developing the system.
3.5	Size
	Size of system development project, either in time or in effort and of the proposed system
3.6	Subcontractor performance
	The subcontractor's performance is defined by the degree in which the subcontractor fulfils the agreements made with regard to the specific project.
4	PROJECT RISK SOURCE AREAS: CONTROL
4.1	Demarcation of the project
	The demarcation of the project is given with respect to the content of the project determined by its objectives on the one hand and on the other hand by organisational, technical and financial constraints.
4.2	External conditions
	Constraints which are dictated to the system by the external environment.
4.3	Project plan
	Plan of activities, mutual interdependencies and to be delivered products that together will result in the desired system
4.4	Slack (contractual and otherwise)
	The amount of leeway that is available within the project planning

4.5	Monitoring
	Are procedures in place to monitor, evaluate and influence progress in terms of on-going activities and deliverables of these activities. It should be noted that monitoring can only effectively be carried out when a formal plan is in place and is well understood by all involved parties.
4.6	Position project approach in the organisation
	The compliance to and experience with formal project management methods
4.7	Position of quality management in the organisation
	The organisation's awareness of and focus on quality, as expressed by the professional attitude of the project's participants and through formal quality management practices being applied in the organisation (e.g. ISO 9000 certified)
4.8	Interdependencies
	Interdependencies occur when the project for its success is dependent on activities of people outside its scope.
5	PROJECT RISK SOURCE AREAS: MECHANISM
5.1	Suitability of working conditions
	The suitability of the working conditions regards quality and availability of equipment, facilities and of the working environment within which the team will have to function
5.2	Hardware
	The computing machinery to be used in the project (host and/or target environment).
5.3	Software and tools
	The software and software tools to be used and re-used during the project. These can be short in supply but also lack of experience may cause trouble..
5.4	Use of methods and techniques
	The use of methods and techniques indicates to what extent a common vehicle for communication can be created between the collaborating parties.
6	RISK SOURCE AREAS DURING SYSTEM USE
6.1	System support organisation
	The activities needed to educate future users and to support them in using the system which have to be provided for during project execution
6.2	System maintenance
	The degree in which the system can be easily enhanced to accommodate evolving user requirements. This usually depends on how well the architecture of the system has been designed, the quality of the technical system documentation and the quality of the training given to the people who are supposed to support the system in its operational phase.
6.3	Data conversion
	The degree in which the designers have anticipated on the conversion of data from existing systems to the new system.

Description

This attribute contains a short description of the risk source. No attempt was made to achieve a complete and exhaustive definition of each risk source. For this there are two reasons, the first being that our approach is based on the uniqueness premise mentioned in chapter 4. Our descriptions appeal to common software engineering practices and language. The reader should be aware that in individual cases company specific practices and jargon can be applied to make the descriptions meaningful for a particular audience. A second factor is that we decided against an exhaustive description in favour of the presentation of a number of questions accompanying each risk factor elucidating it.

Extremes

To add to the definition some extremes are presented that give an indication of the circumstances that indicate high or low risk. The notation in the checklist will be:

- high: key word or sentence indicating high risk
- low: key word or sentence indicating low risk

Example

For each risk factor one or more examples are presented that indicate how that risk factor might influence project execution. These examples can be used as a starting point from which the organisation can develop its own reference of relevant examples.

Questions

For each risk source in addition to the definition a number of questions is presented. These questions serve to further illustrate the meaning of this risk source. If the questions have been answered for the project in question a view of the meaning and possible impact of the risk factor for this specific project can be obtained. As such the questions and answers together form a definition of the risk factor in terms that directly relate to the project in hand making communication on possible effects of the factor a more viable proposition.

Relevant role

A number of risk sources is related to the fact that people play a major part in software engineering. As mentioned before the concept of 'role' that indicate the different responsibilities and background represented in a project plays a major part here. Most risk sources in the human risk source area may have an effect on project execution from the point of view of several roles. For instance, commitment may cause a problem both with users and developers. For those risk sources where this is relevant an explicit indication of the roles to

be considered is given. In defining the roles we took a situation in which an outside contractor is involved as a starting point. If the project is entirely carried out within the client organisation the roles will still be relevant, but some will tend to overlap. As was mentioned before, the following roles will play a part in defining the relevant risk sources:

- *Sponsor*
- *User*
- *User management*
- *Contract manager*
- *Client project manager*
- *Client edp staff*
- *Supplier project manager*
- *Supplier edp staff*

Phase:
Not all risks from a given risk sources can occur at the starting phase of the project. Some are only able to play a part as late as during implementation of maintenance. This means that during the preceding phases additional information may be gathered to deal with this risk source. We also know the latest phase in which the measures, which have to be taken to reduce the likelihood of risks to occur to an acceptable level, have to be effective.

Managing also means looking ahead as far as necessary and possible. So at a given moment the view must not be limited to the risk sources of the coming phase only. In any case before the start of every phase, necessary measures have to be taken for each risk source which might pose danger in that specific phase.

To define the moment the risk source has to be analysed, it is necessary to know the moment this risk source becomes active and the time span needed to take appropriate reactive measures, see Figure 5.1.

In the checklist an indication is given of the phase in which the risk source can become active for the first time. Given the large number of situational factors that can play a role here this indication might not be correct for every development environment. In order to be able to use this attribute in this checklist we had to define a systems life cycle model. We opted for a fairly general model consisting of the phases:

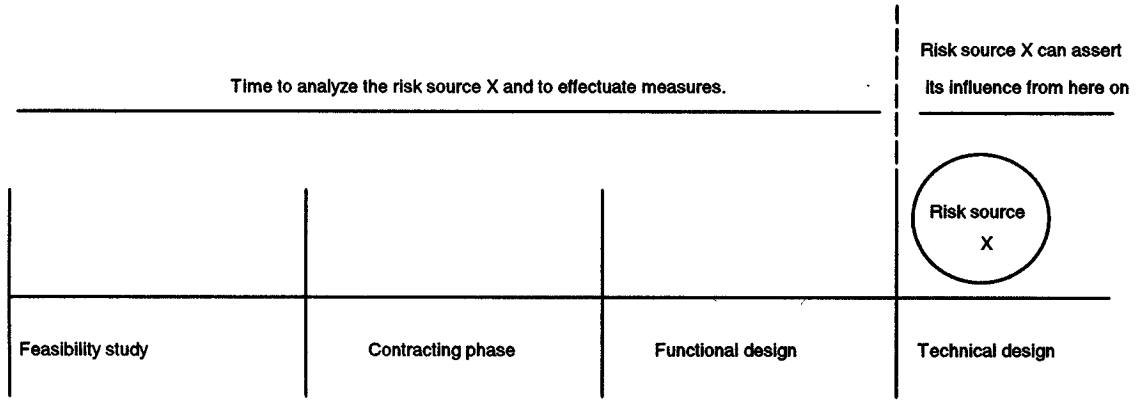

Figure 5.1: activation point of a risk source

- feasibility study
- contracting phase
- functional design
- technical design
- realisation
- operation

In each actual implementation this model will of course have to be adapted to local conventions. A global overview of the activation points of all risk sources is presented in figure 5.2.

Responsibility

Some of the risk sources are susceptible only to the assertion of influence by the client organisation, some are the sole responsibility of the supplier organisation and for some there is a joint accountability by both organisations. In the checklist an indication of the responsible party is given. However, this responsibility may also depend on the result of a negotiation process between the parties involved which may yield different results. Finally it has to be noted that in the final reckoning a personal responsibility will have to be assigned where within the limits of the project organisation a clear responsibility is required. An overview of the main responsibilities per risk sources is presented in figure 5.2.

Examples of measures

When a risk source is judged to be a relevant danger to the project one or more measures will have to be taken. These may range from measures that can be taken directly to contingency plan that may be referred to when the treat posed by a risk becomes reality.

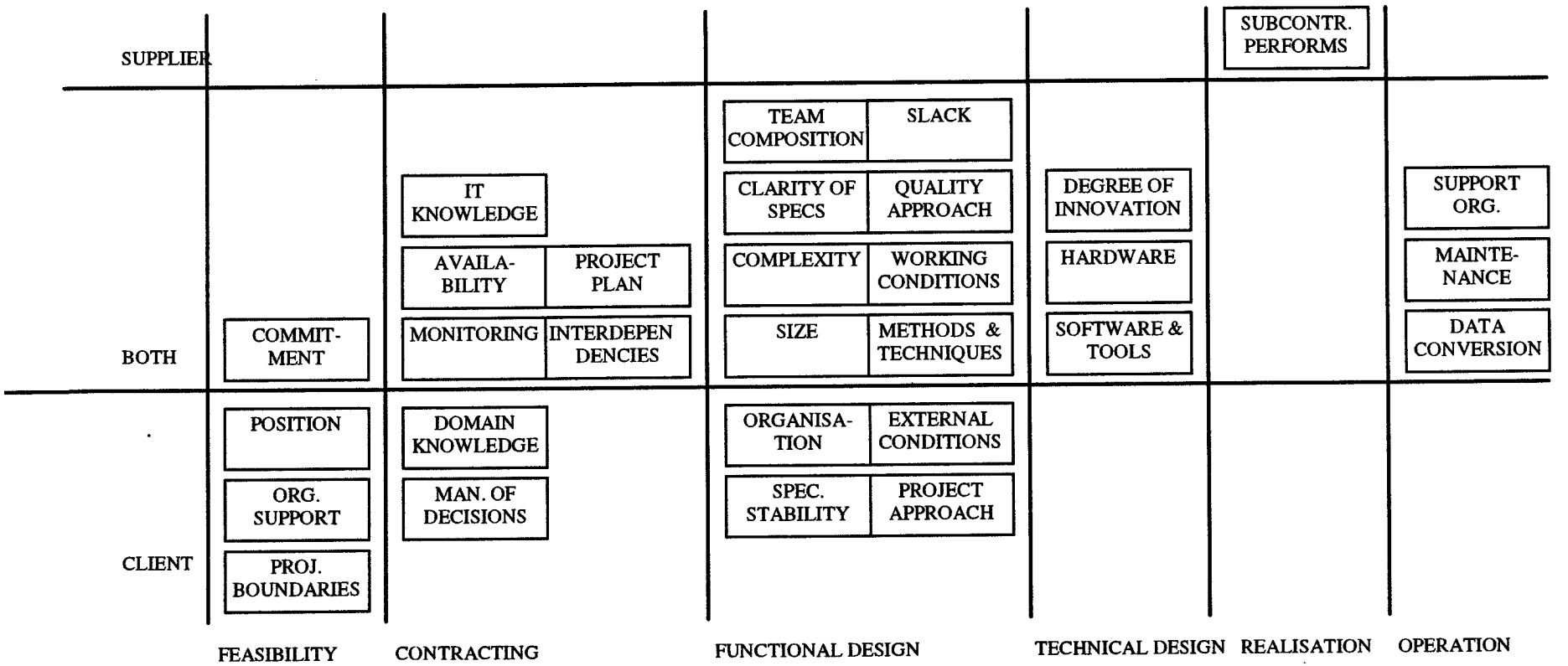

Figure 5.2: overview of risk sources with responsibility and activation point

As was mentioned in chapter three these measures will generally belong to one out of the following four categories:

1. Avoidance
2. Reduction
3. Compensation
4. Contracting

In the checklist a number of examples of measures that may be taken to deal with the effects of the risk source is presented. For each example an indication of the category (avoid, reduce, compensate, contract) is given. Of course no claim of completeness is made here. The number of measures is limited only by the imagination of the people involved and the and diversity of effective measures will be enormous.

However, this list can be used as the basis for a locally defined reference list that reflects the local policies that in the past were seen to be effective. Of course it is necessary that the parties involved in the project review and maintain this list on a regular basis. Looking back at the risk sources and their associated measures regularly and determining their effect provided a valuable feedback on effectiveness and efficiency of measures.

A project logbook can be a help with this. A project logbook is defined here as a registration of the risk sources identified as being potentially dangerous, of the measures taken for dealing with this problem, and of the results that were obtained. It is used for monitoring purposes during project execution. Another advantage is that the amount of information which is usually lost when people or parties change during the project can be limited to a minimum. Finally it is possible to use this registration as a reference for new, comparable projects.

5.4. Assessment of risk sources

The framework presented in the previous sections can be a useful aid in identifying the relevant risk sources for a project. However, more than this is required. A risk has been defined in chapter 2 a 'the probability of a certain deviation between the unintended and the actual output of an activity'. The operative notions here are the size of the expected impact and the probability of this impact actually occurring. Before one is able to assess the effect of a risk in such a way that it may be used in risk management these two notions will have to be

operationalised. In this section we will first look at what type of metric is required and which demands have to be met when designing them. Next metrics are presented for both probability and impact.

Metrics

The evaluation of risks has to take place on the two aspects of probability of risk occurrence and expected result. In order for these results to be captured in a straightforward and reproducible way clear metrics have to be established. Requirements for such metrics are:

- ease of use,
- clarity of meaning

The first item is self evident. Many people will have to use the metric over the years. This makes ease of use paramount. The second requirement will be impossible to meet fully. We ask people for a subjective evaluation in an uncertain situation on a subject for which no intuitively correct metric is available. Not only will different people have a different view on reality, also they will have a personal interpretation of the metrics. Any metric will have to take account of this problem. This means taking care of the following aspects:

- any point on the scale will have to be as concrete as possible; since no intuitive scale is available this means that a synthetic scale will have to be devised,
- given the inherent subjectivity involved, too large a number of points on the scale will only result in fake accuracy; this means that the scale must be as small as possible,
- people often are averse to checking the extreme end of a scale, especially in cases when this extreme has no exact meaning (e.g. 'large'); this means that extra space has to be provided at the end of the scale (as a kind of decoy).

Based on these requirements we will now design the synthetic metrics for probability and impact.

Probability.

The most obvious scale to use for probability is the one ranging from 0 to 1 that is normally used in statistics. However, this is not feasible since on the one hand humans are notoriously bad at mental statistics and on the other hand this scale is continuous providing the risk of fake accuracy. A synthetic scale will at least have to accommodate these two points:

- the probability of the risk occurring is negligible,

- the probability is very high (the decoy).

In these cases normally a five point (Likert) scale is used, but this type of scale accommodates decoys at both ends. Given the clear meaning of the point on the lower end of the scale no decoy is needed there. In between these two extremes at least a point is needed for a relatively high probability and one of a somewhat lower probability. This gives a four point scale:

- negligible,
- medium,
- high,
- very high.

Impact.
This is more of a problem since in determining impact respondents have the choice between time, effort, functionality, and quality. In which of these terms an eventual impact occurs depends on decisions that will be taken later during project execution. Experiences in testing this approach also indicated that people were hesitant to choose for any particular direction. This prompted us to choose for more global terms. However, in order to further the ease of use of the metric some structure has to be provided. Given the basic five point Likert scale as a starting point, and taking into account the fact that no decoys are needed because the meaning of the extremes is clear, this gives us the following scale:

1. any impact is negligible or can at least be handled without problems within the present budget,
2. the impact can not be handled within the existing means, but does not endanger the project,
3. the impact endangers the success of the project.

5.5. Discussion

In this chapter a tool to support identification of project risk sources is presented. In this checklist a number of possible risks is presented which can then be scanned for each actual project in order to determine which of those risks are relevant in the given situation. Using such a checklist has the advantage that a large number of potential risks is looked at in a systematic way thus 'jogging' the memories of the staff involved. One can be fairly certain that relevant risks that are on the list will be identified.

However, this type of list tends to focus the attention of users to the exclusion of other items. Therefore it is less likely that risks which are not on the list will in the end be identified. The danger of a checklist is that it can easily be implemented as a kind of prescription which may be used as a substitute for the critical and creative thought processes that are required here. This is probably all right when it is used for structured, repetitive tasks (e.g. the standard pre-flight check in an aircraft), but not in software engineering projects where creativity and flexibility are of prime importance.

The advantage of using a checklist is often seen to be such, that it outweighs this disadvantage, but it does mean that great importance is attached to the 'completeness' of the checklist. This leads to checklists consisting of hundreds of questions, each aimed at identifying one or more risk factors. A typical example is the list prepared by the Software Engineering Institute (SEI) which is based on a risk taxonomy (thus striving for completeness) and consists of 264 questions.

This approach causes its own problems. To illustrate this, taking the SEI checklist as an example, several comments can be made:

- It is highly unlikely that even this large list will be complete. It is fairly easy to come up with additional risks which are not incorporated into the list. Completeness on the level of risk factors is in our view an illusion. The amount of things that can go wrong is limitless. Therefore, any checklist that on a detailed level tries to enumerate as many as possible risk factors is bound to miss a significant number of factors. Also it will tend to give false confidence, since given the size of the list people would accept that it contains all relevant factors.
- A list consisting of 264 questions is very hard to use. If one wants to involve all relevant parties this means that a significant number of people will have to scan and discuss this list. Experience shows that even a much smaller list is difficult to handle effectively. However, as was mentioned above, it is part of out philosophy to make explicit use of the knowledge and commitment of the parties involved. Therefore if one is in favour of using a checklist, it has to be manageable for a group of people, which indicates a smaller list.

To summarise, a checklist is too powerful an instrument to ignore. The benefit of a list of things that can possibly go wrong is such, that it outweighs the disadvantages. However, when designing and using a checklist two aspects should be kept in mind.

This first is, that any checklist should be used properly. The most a checklist can aspire to in the area of risk identification is to act as an aid in a complex decision making situation. Given the variability of circumstances it should never be allowed to take over control from the decision maker. Also, additional techniques to support the usage of a checklist should be encouraged. Such techniques are: the involvement of resident or hired expertise, group involvement in which all parties involved join in identifying risks, analysis of data on comparable historic projects, and the analysis of assumptions underlying the current project. The combination of a checklist together with these additional methods will usually enhance its effectiveness.

The second aspect to be taken into account is the structure of the checklist. Above we commented on the disadvantages connected to large checklists aimed at identifying and explicitly naming as complete as possible the possible risks that might influence a project. To counter these disadvantages in this chapter a checklist was developed that is not focused at risks as such. Since these risks can be seen as mere symptoms of a deeper lying cause attention was concentrated on the underlying mechanisms, which were named *risk sources.* Given that the underlying causes will be of a more fundamental nature, in terms of cause-effect type of relationships, a risk source based approach will have several advantages.

- The main advantage can be found in the indication this higher abstraction level gives as to how this risk can be managed. The higher abstraction level means that the attention will no longer be focused at mere symptoms but at the real roots of the problem. As a consequence pro-active handling of risks will be facilitated.
- Given the higher level of abstraction it is easier to achieve a complete coverage. At the same time it is clear that for a given project many manifestations of this risk source might occur. The abstraction level used is such that it is obvious that the checklist can not be used as a standard receipt that will provide all information required. This will reduce the adverse effect of misplaced trust in a checklist.
- Finally, a list based on risk sources will be significantly smaller than a list enumerating risks. This is also due to the higher abstraction level used in determining the list. This means that the list is easier to handle for a group of people.

5.6. Conclusion

In this chapter a framework, operationalised as tool consisting of a checklist of risk sources to support risk identification, was identified. Using a checklist has its disadvantages which mainly have to do with the tendency of a checklist to focus the view of users to the detriment

of an open, flexible and creative outlook. However, if this problem is kept in mind a checklist can be very useful indeed. In chapter 6 we will look further at a methodological approach to the way risk identification can take place.

6. AN IMPLEMENTATION OF THE RISK MANAGEMENT APPROACH

In this final chapter an example of a concrete risk management procedure is presented. The chapter ends with some implementation concerns and practical experience gathered while using the method.

6.1. Introduction

This final chapter deals with a concrete elaborated example of the abstract definitions, concepts and premises treated in the previous chapters. As demonstrated earlier a key question for software management is 'how to deal with risks'. A possible answer to this question will be given in this section by presenting an example of a procedure an organisation can choose carrying out risk management. The authors applied the procedure successfully several times both in governmental and for-profit organisations. The procedure is described in section 6.2. In section 6.3 some attention is paid to the introduction, use and control of a risk management procedure. The method described in section 6.3 has been used during the execution of three large IT projects. In section 6.4 the experiences of the risk method users and the so-called risk advisors are presented. The experiences of the risk advisors are formulated as 'lessons learned' or recommendations.

6.2. Risk Management in practice: an example

The risk management approach the authors applied in practice and described in this section can roughly be divided into three phases. Each phase is split in several steps. In figure 6.1 and table 6.1 the structure of the risk management procedure is presented. The various steps are explained in the remainder of this section.

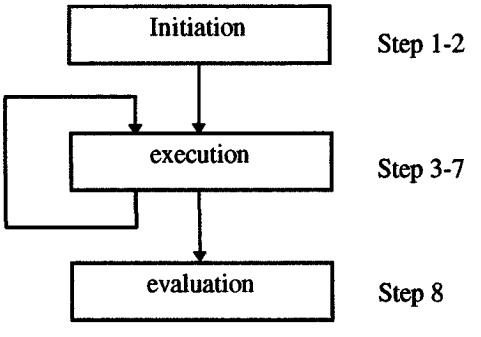

Figure 6.1 The overall structure of the risk management procedure

Table 6.1 The steps of the risk management procedure.

Step 1	Selection members risk management team (RMT)
Step 2	Explanation of the method and planning future activities
Step 3	Identification of risk sources and risk attributes
Step 4	Handling and first analysis of the interviews
Step 5	Selection of identified risk sources and risk attributes
Step 6	Final risk identification and choosing action via a RMT meeting
Step 7	Risk monitoring
Step 8:	Compiling a risk management evaluation report.

The first phase is called the initiation phase and consists of two steps:

Step 1 : *Selection members of the risk management team (RMT),*
Step 2 : *Explanation of the method and planning 'risk activities' / team members contributions during the project,*

In the initiation phase activities are carried out with regard to a start up of the procedure. Both step are executed at the very early start of a project and can be considered activities of the contracting phase. The two steps, as well as the ones that will follow, will be described in this section using a fixed format.

The second phase is called the execution phase. In this phase activities are carried out regarding assessment and control of risks. More specific the next five steps must be executed:

Step 3 : *Identification by means of interviews of risk sources and risk attributes,*
Step 4 : *Processing and first analysis of the interviews,*
Step 5: *Selection of identified risk sources and risk attributes,*
Step 6: *Final risk identification and choosing action via a RMT meeting,*
Step 7: *Risk monitoring,*

Step 7 in facts is a repetition of the previous steps and should be repeated more than once during the project in cadence with the normal project management activities. As mentioned earlier in this book risk management is a continuous object of attention during a project. An organisation can agree for example that that these 5 steps must be executed at the end of each project phase and at unexpected tricky moments during a project.

The last phase finally is called the evaluation phase. Besides compiling a formal evaluation report, "risk experiences facts and data" concerning the project must be recorded in order to be fruitful for future projects. These kind of activities must be executed in step 8:

Step 8 : Compiling a risk management evaluation report.
Step 8 is a one of the final parts of a project.

The eight steps will now each be described in more detail in terms of objectives, guidelines for execution, and deliverables.

Step 1 : Selection members risk management team (RMT),

Goal:
The goal of the first step is the composition of a Risk Management Team (RMT). The representation of each party involved in the project by minimal one agent is an important starting point in the selection procedure. In chapter 5 the different parties and their roles in the project were already mentioned. Arguments for a composition of all parties involved are:

- the acceptation of and commitment to the results is enlarged if all parties are involved,
- the communication between the parties concerned is promoted. Everyone gets a clear insight into each other's opinions, motives, objectives, etc. by working with the method as a team,
- working as a team makes use of the advantages of group work compared to individual work in uncertain / risky situations. Well known advantages are principles like 'more people know more than one' , uncertainty reduction (confirmation of one's opinion by others) , team building , communication , etc.

The resulting team is likely to have a composition that is wider than that of the project team. In that case a separate team will be required to carry out the risk management activities. This team will of course have a high overlap with the project team which will at least include the project manager.

Execution:
It is recommendable to limit the size of the RMT to approximately ten persons. For smaller projects obviously less people will do provided that both the supplier and the customer role are represented. The project manager, in co-operation with a so-called risk advisor are the initiators for the selection. The risk advisor, an experienced user of the risk management procedure, should be someone from outside the project.

Earlier it was advocated that ideally risk management is a task for every member of the project team and that it should be considered as an integral part of all managerial activities before and within the project. It was also argued that projects should be managed on the basis of a win-win premise between supplier and customer. In spite of this it makes sense to involve a risk advisor external to the project and independent from both supplier and customer. This for the following reasons:

- to avoid the danger of group opinion convergence ('group think'), an outsider can be expected to take a fresh look at things,
- to act as an neutral referee because in practice it will be difficult to maintain a win-win attitude throughout the project, especially when problems arise,
- to act as a professional expert in risk management practices as such who can play the role of process guide and stimulate the decision making process.

Our experiences with the described procedure indicate that the contribution of a risk advisor is very desirable. In section 6.4 these experiences will be described in more detail.

Result:
List of names of RMT members.

Step 2 : Explanation of the risk management procedure and planning activities

Goal:
The goal of step 2 is twofold. First of all it is important to inform all parties / members about the work procedure. Secondly it is necessary acquiring the required commitment. If one doesn't believe in the results and isn't motivated to participate a successful application of the risk management procedure is out of the question. This means explanation about the working-method and agreements about the work to do. It must be clear what is to be expected of the team members, how much time it costs and what the results will be. As

mentioned in chapter 5 premises like commitment, communication and information provision are required for successful implementation of risk management. In step 2 these premises are implemented.

Execution:
The step can be carried out right after the selection by means of a meeting. If the team members are experienced with the procedure then forwarding of written documentation suffices. This approach fits for small projects too. The meeting is organised by the project leader, he/she is chairman at the same time. The risk advisor takes care of the explanation.

Result:
Work plan of risk management team, agreements, list of tasks and responsibilities.

Step 3 : Identification of risk sources

Goal:
The risk advisor interviews each member of the RMT. By means of the interview a member's risk identification is made. For this purpose a checklist is used (see appendix). The checklist contains a list of pre-defined questions about risk sources. The list should be adapted to the local language (jargon) of the organisation and the specific characteristics of the project concerned. This step results in an overview of the expected risks of each team member individually. Besides the identification of expected risk sources each team member is asked the effects of his expected risk sources.

Execution:
Each team member receives the checklist beforehand for preparing the interview. The preparation takes 2 hours and the interview 1.5 hours. During the interview the risk advisor checks for each risk source if the probability of appearance is negligible, medium, high or very high and checks the effects in case a risk sources indeed becomes topical. It is advisable to interview all RMT within a week or at least as soon as possible.

Result:
Interview results containing the answers (the identification of the expected risk sources) of each RMT member. That means for each identified risk source the probability of appearance and the effect in case of appearance.

Step 4 : Processing and first analysis of the interviews

Goal:
The individual interview results are passed on to a summary table. After completing the table a critical analysis must be carried out. For each risk source from the checklist the following situations can occur:

- nobody identifies it as a potential danger for the current project,
- it is identified by some RMT members, but the views on the expected probability of occurrence and possible effects differ,
- everyone considers it to be equally relevant.

The first group of risk sources, those identified by none of the RMT members, are for the moment not relevant. It might of course be the case that a collective blind spot exists, but there is no way that this blind spot can be now identified. The last group of risk sources, those identified by all members need no further analysis apart from deciding on counter measures. The grey area in the middle is really interesting for further analysis. The parties in the project apparently have a different view on possible risks, expected probability of appearance and possible effects. In the next steps it is explained how to deal with these differences.

Execution:
Completing the summary table is an administrative task and takes about half a day. The first analysis is merely a first rough selection eliminating the irrelevant risk sources and checking the distribution of the answers. Right after finishing the interviews the risk advisor starts with this step.

Result.
Complete filled summary table

Step 5 : Selection of identified risk sources

Goal:
The risk advisor selects the most relevant risk sources from the summary table. These will be discussed in more detail in step 6.

Execution:
During the interviews the risk advisor has gather structured data from up to ten people. This information can be structured by ordering the risk sources by the number of times they were mentioned as being relevant for the project. Also, since the interviews will preferably take the form of a discussion, a lot of additional information has been gathered. Together with the risk management expertise of the risk advisor this should provide him with sufficient insight into the project to carry out this pre-selection. The opinion of the risk advisor is decisive.

The complete time span between the interviews and the risk selection must be restricted to at least two weeks. Slow progress leads to loss of attention. Moreover the project has it's own pace. The selection is carried out by the risk advisor.

Result:
A pre-selection of the most relevant risk sources. For each risk source it is reported by how many people it was identified, together with the expected probability of occurrence and the expected effects. Furthermore the risk advisor will often be able to add some comments on the issue.

Step 6 : Final risk source identification and choosing action via a RMT meeting,

Goal:
The objective of the risk management team meeting is:

- confront the team members with each other's perception of potential risk sources,
- to start a discussion on probability and effects of identified risk sources,
- to reach the most uniform team decision as possible about risk sources, probabilities and effects,
- to agree upon which risk reducing actions have to be chosen, who is responsible for the execution, the supervision of the execution, when which action has to be done, how to report, etc.

Execution:
In order to prepare all the members of the RMT receive the interview results. Through here it becomes possible that each team member compares his/her interview results with the other one's. Step 6 must be carried out at least one week after receiving all the information. The procedure during the meeting is:

- the selected risk sources are checked one by one. The discussion must lead to a decision like: 'this is a risk source with a probability of x% of appearance and with y, z and u as possible effects',
- for each selected risk source possible actions have to be conceived and chosen. The ultimate goal is to choose a set of actions that are accepted by all RMT members. Consensus and commitment are key elements at this point. The project leader takes the decision if the team doesn't come to an agreement,
- if the team comes to terms the team has to agree upon if the risk source is relevant enough reckoning with during the next steps of the risk management procedure. In this case the project manager has the last word too,
- to check if the team values some of the not selected risk sources as important after all,
- to check if the ultimate list of risk sources is complete. By using the checklist one runs a risk to stick to the list.

The risk advisor is responsible for the completeness and timeliness of the information to dispatch. The project leader invites the RMT members and chairs the meeting. The risk advisor conducts the meetings, the project manager formalises the decisions. The risk advisor minutes the discussion and records the decisions.

Results:
A final list of 'current risk sources'. These are the most important (the topical) risk sources, together with their probability of occurrence and possible effects at appearance. For each identified risk source an overview of chosen actions, responsibilities, competencies, work-scheme, etc. This list of current risk sources forms the basis for further risk monitoring activities.

Step 7 : Risk monitoring,

Goal:
During the execution of the project it has to be checked if the correct risk sources are indeed identified and properly estimated, the chosen actions are executed and the intended effects have been achieved.

Execution:
During the execution of the project the risk team has to meet more than once to pronounce upon the state of the art of the current risk sources. This is done by formally reconvening the RMT on a regular basis. When and how often depends on the size, complexity, importance,

number and type of risk sources etc. An obvious option is to combine these meetings with the decision points at the end of the project phases. A possible guideline is four meetings (including the start-up and evaluation meeting) for a one year project. For each current risk source the next points have to be discussed:

- have the identified risk sources been correctly estimated,
- have the risk reducing actions been executed,
- what are the effects of the actions,
- should new actions be considered,
- is it possible to delete the risk source from the current list.

Furthermore it is checked if new risks sources (not identified before) have to be added to the current list. This can be done by discussing the checklist during the meeting in a structured way. For each addition the risk exposure has to be estimated and it has to be decided which risk reducing actions have to be taken.

The risk advisor prepares the meeting in co-operation with the project manager. He furthermore invites the risk team members and chairs the meeting. The objective of the meeting is streamlining the discussion and aiming to reach the most feasible consensus. The risk advisor conduct the meetings, the project manager formalise the decisions.

Results:

- a status report containing the last meetings' identified risk sources with risk reducing actions to match (the new current list),
- report and decisions of the risk monitoring meeting.

Step 8 : Compiling a risk management evaluation report

Goal:
The objective of the risk management evaluation report is to:

- give an overview of the risk sources and the chosen actions that play a major part during the project,
- summarise which risk sources after all are the best bits,
- conclude the 'lessons learned' for future projects.

Execution:
The risk advisor in consultation with the project manager write the report at the end of the development i.e. after the acceptation tests. A so-called risk profile of the project is compiled according to a fixed format. The risk profile is a structured summary of the reports of the previous steps.

Result:
- an evaluation report which can provide a reference for future projects,
- concrete actions for improvement.

6.3. Introduction, use and control of a risk management procedure

The introduction
Introducing a risk management procedure takes a lot of time and effort and requires therefore a well prepared introduction within an organisation. An introduction is needed to convince all parties of importance of identifying and controlling risks. Also the premises the risk management procedure is based on mentioned in chapter 5 must be accepted by the organisation. To enlarge the acceptance the organisation has to aim at combining it with existing risk approaches, project control methods and working procedures within the organisation. This means for example adapting the risk management procedure to the used organisation language, definitions and ideas. Adjusting the checklist is inevitable. It is advisable starting the introduction with a pilot project. Nature and number of adaptations and degree of acceptation becomes clear then.

The introduction asks for a good announcement, for instance education meetings, information by way of internal bulletins, feedback of results from elsewhere. An important part of the introduction is also the selection and education of a risk advisor and fitting up some kind of help desk.

Use
The described premises and risk management procedure is not limited to a specific type of projects. It is however as a matter of course that the effort/costs using the procedure must bear a proper proportion to the effort/costs of the complete project. For small projects a so-called 'short cut' is advisable. As far as the costs of using the procedure goes it turned out to be from the author's own experiences that the time the procedure asks from risk advisor project manager and risk team members is calculable. Table 6.2 gives a rough indication of the required time.

The parties i.e. the users of the method must be acquainted with the limitations of their competencies. Identifying risk sources and choosing actions is one side of the picture, paying the costs of executing the actions is the other side of the coin.

Table 6.2: How much time costs the use of the risk management procedure (N = number of risk team members, M = number of risk team meetings).

Step	Risk advisor	Project manager	Member
1 Selection members	1 hour	1 hour	0 hours
2 Explanation	4 hours	1.5 hours	1.5 hours
3 Risk identification	3 * N hours	1.5 hours	1.5 hours
4 Processing data	1 * N hours	0 hours	0 hours
5 Pre-selection	8 hours	0 hours	0 hours
6 Group meeting	8 hours	3 hours	3 hours
7 Risk monitoring	8 * M hours	3 * M hours	3 * M hours
8 Evaluation report	8 hours	0 hours	0 hours
Total	29+ 4*N+ 8*M hours	7 + 3*M hours	6 + 3*M hours

Control

A risk management procedure will or even must evolve during use within an organisation. The procedure must therefore be adapted to changing circumstances, opinions etc. The results of step 8 (evaluation) should provide input for this. This all means that the procedure needs control, changes must be realised and new releases must be distributed. A central point in the organisation, a help desk, is recommended. We also refer at this point back to chapter two where a distinction was made between strategic, tactical and operational risk management. The control actions mentioned here both demarcate the boundary between operational and tactical activities and allow the exchange of views and ideas between them.

6.4. Evaluation of the risk management method

The described risk management method has been tested on consistency and usefulness several times in 'real life' projects. In this section the results of these tests will be explained in short. The description of the results consist of two parts. In part one the opinion of the risk team members are presented. In part two the experiences or the 'lessons learned' of the authors / risk advisors are explained.

6.4.1. Evaluation by the risk team members

The risk team members' opinion had been find out by a written inquiry. In the inquiry questions were used like 'have you used a similar method once before, did you like working with the method, what are strong and weak points of the method, etc. The most important results are mentioned in short.

- None of the risk team members used a similar risk management method in the past. For all of them it was a new approach.
- Most of the team members liked working with the method. Some critical comments were related to
  - the time intensity of the method;
  - the project felt himself in the middle of the "firing line";
  - it takes some time to get familiar with the method;
  - the discussions during the risk monitoring sessions were too extensive sometimes.
- All the team members were enthusiastic about the use of the checklist. The list turned out to be a valuable tool for risk identification. The team members didn't had the feeling that they were pushed to the risk factors mentioned in the checklist.
- Also the group meetings were unanimously positive appreciated. Not only the discussions about risks, risk reduction etc. appeared to be useful but also the side effects namely communication about goals, expectations, responsibilities etc.
- In all the test cases the risk team members were of opinion that the implementation of the method was a success factor for the project. Using the method increased the risk awareness among the team members extremely.
- The strong points of the method were:
  - clear risk evaluation and decision moments during the project;
  - discussing risks in group meetings;
  - unexpected risk factors are identified;
  - a good overview of possible risks;
  - more insight in risk, origin of risks;
  - during the project, from the start until the end, the attention was focused on tackling the identified risks.
  - the external risk advisor stimulates objectivity.
- Weak points:
  - the description of the risk factors were sometimes abstract;
  - time intensive;

- unclear relation with project management;
- too much focused on IT dimensions of projects;
- at the start of a project the risk advisors were not enough familiar with the project.

- Despite the positive opinions 50% of the risk team members were reserved using the method in the future for their own projects.
- About 75% of the team members were of opinion that the risk advisor should be someone from outside the project, but familiar with the organisation.

The results show clearly that the method is appreciated by the risk team members. The checklist, the interviews, the group meetings, etc. turned out to be rated positively. All respondents are convinced that the application of the method has contributed success to the projects.

6.4.2. The evaluation or the 'Lessons learned' of the risk advisor

The authors operated as risk advisors during three large IT projects. Although the method turned out to be a useful tool for all the participants in the project, the authors have some special points of attention for those who are interested in implementing the method after reading this book . These recommendations are based on our experiences with the method and the many discussions we had with a number of organisations who intended to start using the method. The recommendations are related to:

- the role of general management;
- commitment of all parties;
- the role of the risk advisor;
- the role of the project manager;
- the selection of the risk team members;
- use of the checklist during the interviews;
- data handling;

In our view a successful use of the risk management method depends strongly on general management support, commitment of all the participants in the project especially the risk team members and an enthusiastic project manager who believes in the approach. That is why so much attention in the method is paid to communication / information provision and education.

We learned that not enough management support is a good reason not to start with a risk management approach, or in general: not to start the project! The project manager of course plays a crucial role in the method. He is the one who initiates a risk analysis and together with the risk advisor he is a driving force during the use of the method. However, the method bring to light weak project management. If that risk comes clear the execution of the method can get into danger and project manager's support cannot be guaranteed anymore. In such case general management support is important.

Furthermore we learned that the selection of the risk team members must be done careful. The co-operation of enthusiastic team members who are willing to spend time and effort in the method is required. They are the ones who provide the risk advisors with data. They must be convinced of the importance of risk management and must use the results of the method as important information. We noticed and experienced that the contribution of selected team members who did not met these demands were contra-productive.

We are convinced that the presence of an external risk advisor is extremely important during the use of the method. It is in our point of view of importance that the risk advisor should be someone from outside the organisation with no organisational interest or some hidden agenda. For instance an external consultant could be hired for this job or someone could be found within the company with no interest in the project, political or otherwise.

We learned that communication about the project and possible risks in the project using the local organisational language is necessary. For that reason much effort has been spend on customising the checklist to the language of the specific organisations. We noticed during the interviews that the risk team members had the impression that the communication was focused on their specific problems. This feeling was enlarged by an open-end discussion after each interview. The team members appreciated this open discussion apart from the fixed format checklist.

As risk advisors we learned that the activities 'data collection' and 'data handling' took a disproportionate amount of time. However these activities are important and should be carried out carefully. The productivity of the risk advisor could be increased and the time consuming clerical activities could be minimised by using an automated tool set.

7. BIBLIOGRAPHY

7.1. Books

Andersen, E.S., Kristoffer, V.G, Haug, T. and Turner, J.R.,
Goal-directed Project Management,
Coopers & Lybrand, 1987.

This book discusses the basic principles to manage a project efficient and effective. The headlines and approach give the company a helping hand to plan, organise and manage a project.

Blokdijk, A. and Blokdijk, P.,
Planning and Design of Information Systems,
Academic Press Limited, London, 1987.

This book is intended to give a theoretical base and a practical method of executing the planning of organisation supporting computerised information systems, and the planning and design of individual applications, of which the boundaries and priorities are defined in the information systems plan.

Boehm, B.W.,
Software Engineering Economics,
Prentice Hall, Englewood Cliffs, 1981.

This book presents the constructive cost model (COCOMO) for software cost estimation, a useful tool for software-risk identification, analysis, and prioritisation.

Boehm, B.W.,
Software Risk Management,
IEEE Computer Soc. P, US, 1989.

This pattern tries to formalise risk-oriented correlates of software project success into a discipline of software risk management. The structure of this discipline is organised into two main branches: risk assessment and risk control.

Brooks, F.B.,
The Mythical Man-Month: essays on software engineering,
London Wesley, 1975, 1982.

This book is a classic on software project management and the risks that may occur.

Doorewaard, H. and Regtering, H.,
Integraal automatiseren;
Kluwer bedrijfswetenschappen, Deventer, 1990.

Genuchten, M.J.I.M. van,
Towards a software factory,
Doctoral Thesis, Eindhoven University of Technology, 1991, ISBN 9090041192
The subject of this book is the control of software engineering. The aim is described as: determine the characteristics of the control concept of software engineering that fit in with the changed practices and demands.

Heemstra, Fred J.,
Hoe duur is programmatuur? begroten en beheersen van software-ontwikkeling,
Kluwer Bedrijfswetenschappen, deventer, 1989.
The central subject in this thesis is the estimation and control of software development costs. It has been prompted by sharp overshoots of budgets and delivery times which have been signalled in the development of software.

Krooshoff, R.L., Thackwray, J.D., Brands, J.W., Bates, R. and Boutelegier, R.,
Guide to project management - PRODOSTA,
ISA, NV Philips, Eindhoven, 1987.
This book described the Philips in-house software development method.

Krooshoff, R., Swinkels, F. and Wal, Bart van de,
ISES-PROMISE; handleiding voor Projectmanagament (deel 1),
ISES - International, Utrecht, December 1991.
Guide to project management and quality assurance in information and software engineering. Part 1 introduces concepts for phasing, control, decision making and quality in projects and provides activity lists for contracting, setting up, monitoring and closing projects, applying the rules of ISO 9001 and guidelines of ISO 9000-3.

Rakos,
Software project management for small to medium sized projects,
Prentice-Hall, 1990.
The development method that is described in this book uses the time-phased approach, but the phases and especially the documentation are greatly simplified. The approach focuses on planning and control. The book also emphasises risk management-knowing what can go wrong in project and tempering estimates accordingly.

Riesewijk, B. and Warmerdam, J.,
Het slagen en falen van automatiseringsprojecten,
Instituut voor toegepaste sociale wetenschappen, Katholieke Universiteit Nijmegen, 1988.
This is a report on a large scale empirical study into causes of success and failure of IT projects.

Robbins, S.P.,
Essentials of Organisational Behaviour,
Prentice-Hall International, Inc., Englewood Cliffs, New Jersey, 1984, 1988.
This book extracts the key concepts ordinarily found in a 600- or 700-page textbook on organisational behaviour (OB) and condenses them into a more concise volume. the text includes discussions of those topics usually identified as the core of OB.

Rijsenbrij, D. et.al.,
Projectdiagnose,
Cap Gemini Pandata Publ., Rijswijk, 1991.
The project diagnosis is an audit tool that has to be used before the execution of the project. This book explains this method, which is based on a large list with questions.

Simpson,
New techniques in software project management,
John Wiley & sons, Inc., 1987.
This is a book on general project management techniques.

Turner, W.S., et.al.,
SDM System Development Methodology,
Cap Gemini Publishing, Rijswijk, 1990.
This books gives a full explanation of the in the Netherlands wide spread software engineering method SDM.

Wijnen, G., Renes, W. and Storm, P.,
Projectmatig werken,
Het Spectrum, Utrecht, 1989.
This is a general book on project management.

7.2. Papers

Abdel-Hamid, T.K. and Madnick, S.E.,
Impact of Schedule Estimation on Software Project Behaviour,
Software, July 1986.

Two methods of project cost estimation are compared, the differences and the consequences of these differences on the project are discussed.

Alting van Geusau, V.W. and Delen, G.P.A.J.,
Projectrisicoanalyse,
Open Universiteit, Checklisten Informatiemanagement, Afl. 5.

A standard method of looking at risks, but complete with documents etc.

Akkermans, H., Vennix, J. and Rouwette, E.,
Participative Modelling To Facilitate Organisational Change: A Case Study,
Paper presented at the 1993 International System Dynamics Conference Cancun, Mexico, July 19-23 1993.

Paper on a method to facilitate re-organisational effort.

Baker, N.R., Green, S.G. and Bean, A.S.,
Why R&D Projects succeed or fail,
Research Management, Nov/Dec pp. 29-34, 1986.

Analysis of 211 R&D project reveals that the successful ones were able to resolve initial uncertainties concerning business and technical goals.

Beidleman, C.R., Veshosky, D. and Fletcher,
Using project finance to help manage project risks,
Project management Journal June 1991.

The project finance process, characterised by limited resource to the assets of the sponsor, use the third party funds, and allocation of risks to those parties best able to manage them, can also assist project management.

Braams, P. et.al.,
Projectmanagement in Nederland, concepten en technieken,
Rijksuniversiteit Groningen, studiegroep Projectmanagement, 1982.

An article on the use of project management techniques in the Netherlands.

Brandt Corstius, J.J. and Schimmel, H.P.,
Project-Risico-Analyse, Een hulpje voor de projectleider,
Journal of software Research no. 3 October 1989.

Manage the project risks by developing a project-risk-analysis, executed with help of the PRAL method, and by making of a checklist for the project manager.

Bobrowski, P.M.,
Project Management Control Problems: an Information Systems Focus,
Project Management Journal, 1989.

This paper describes four important control problems in different projects.

Cobbenhagen, J.W.C.M.,
Innoveren: strategieën en modellen,

in: Hertog, J.F. den and Eijnaten, F.M. van(eds.):
Management van Technologische Vernieuwing;
Van Gorcum, Assen, 1990.
This article discusses the importance of models by innovation projects.

Demarco, T. and Lister, T.,
Programmer performance and the effects of the workplace.

Wide variation in programmer performance has been frequently reported in the literature. In the absence of other explanation, most managers have come to accept that the variation is due to individual characteristics.

Dingle, J.,
Cultural issues in the planning and development of major projects,
Project Management, Vol. 9 No 1, February 1991.

This paper describes investigations into the use of a simple "expert system" as a tool for measurement and comparison of cultural influences on decision-making processes in the risk management of project development.

Fikkert, D.W.,
Using the "Spiral Model" - Problems and solutions,
TNO Physics and Electronics Laboratory, STC version mei, 1990.

An objective of this paper is to explain and discuss aspects of the spiral model that are different compared to other approaches.

Fikkert, D.W.,
Usability and maintainability, the management of change?,
Division System Development and Information Technology, TNO Physics and Electronics Laboratory.

This article discusses management as an important contributor to usability and maintainability of software systems.

Genuchten van, M.,
Why is Software Late? An empirical study of reasons for delay in software development,
IEEE transactions on software Engineering, no 6, June 1991.

This paper describes a study of the reasons for delay in software development that was carried out in 1988 and 1989 in a Software Engineering Department. The aim of the study was to gain an insight into the reasons for differences between plans and reality in development activities in order to be able to take actions for improvement.

Haas, R.J. de and Wubbels, C.S.M.,
Situationeel projectmanagement bij automatisering, eerst denken, vervolgens de risico's analyseren, en dan pas doen!,
Informatie, vol. 32, no. 2, p 202-210, 1990.

More projects will be closed successful by intensifying the think- and negotiation-process and the risk analysis by environment and project characteristics.

Lierop, F. van, Volkers, R., Genuchten, M. van and Hoekstra, F.,
Heeft iemand de software al gezien: inzicht in het uitlopen van software projecten,
Informatie jaarg 33, nr 3, pag 193-200, 1991.

An investigation into the origin of delays of software projects.

McFarlan, F.W.,
Portfolio approach to Information Systems,
Harvard Business Review, vol. 59, no. 5, p142-150, 1981.

The classic paper describing the basics of modern IT risk management.

McMullan, L.E.,
Cost Control - The Tricks and Traps,
AACE Transactions, 1991.

This paper discuss who really "controls" project costs, the role of the cost engineer during the project life cycle and the impact of computer and information technology on the cost control process.

Mikkelsen, H.,
Risk management in product development projects,
Risk management, Vol. 8, no. 4, November 1990.

The preliminary result from a Danish study of the handling of uncertainty and risk in product development projects are presented in this article. The study revealed that the handling of risks depends on attitudes to risk and on the risk-taking culture in the project organisation.

Moynihan, T., McCluskey, G. and Verbruggen, R.,
RISKMAN1: A Prototype Tool for Risk Analysis for Software Project Managers,
National Institute for Higher Education, Dublin.

This paper describes a prototype tool (RISKMAN) which aims to help a software development project manager identify sources of risk to the success of a project and to help him/her identify risk reduction strategies.

Nijhuis, R.J.,
Omgaan met risico's in software projecten,
Literature study Eindhoven University of Technology, January 1993.

An almost complete summary of all methods for Project Risk Management, compared with 10 critical success factors.

Nijhuis, R.J.,
Project Risk Management: A Critical Success Factor,
Master thesis Eindhoven University of Technology in co-operation with BSO/Origin, July 1993.

This report presents the result of a study of the problem of Risk Management in software projects. After an evaluation of known software Risk Management methods, it describes how the shortcomings of these methods can be met with a practical Risk Management model for information system development, integrated in project planning and control.

Oudshoorn, H.,
Ontwikkelmethoden 5: SDM en de technieken TIA, GOS, GOP en TOT,
Informatie vol. 23 no. 4, April 1981.

This article describes different techniques, which are connected to the activities of SDM.

Reeken van, A.J.,
Leren omgaan met onzekerheden,
Handboek Bestuurlijke Informatiekunde, June 1992.

A summary and a comparison of different methods for risk management, with different approaches.

Reeken van, A.J.,
Naar een andere aanpak in de systemering,
In: Hertog, J.F. den en F.M. van Eijnaten (eds.):
Management van technologische Vernieuwing;
Van Gorcum, Assen, pp.129-153, 1990.
This article describes the risks concerning the organisation changes by automation projects and a way to handle this problem.

Rijsenbrij, D.B.B. and Bauer, A.H.,
Projectdiagnose: goed begin is het halve werk,
Informatie, vol. 31, no. 3, p182-193, 1989.
A project diagnose is an investigation into circumstances at the start of the project. The basic of this method is discussing the strong and weak points of the project with the project team.

Rothfeder, J.,
It's late, Costly, and Incompetent-But Try Firing a Computer System,
Business week, November 7, pp. 164-165, 1988.
Short paper in the occurrence of risk in software projects.

Salvati, A.,
Risk Management in Software Projects.
This paper relates risk analysis to an evolutionary planning approach that identifies a number of intermediate steps of the final deliverable product.

SBA: SarBachet Analys,
ISES International.
A description of the Swedish risk management method SBA, which contains an large checklist together with an extensive manual for use.

Senge, P.M. and Sterman, J.D.,
System thinking and organisational learning: Acting Locally and thinking globally in the organisation of the future,
European Journal of Operational Research, no. 59, pp.137-150, 1992.
General paper on knowledge management from a systems point of view.

Shafer, S.L.,
Estimate and Project Risk Analysis Approaches,
AACE Transactions, 1991.
The thesis of this paper is that each method is appropriate for particular circumstances and that rather than seeking a universal best approach, it is appropriate to define which method best fits each specific situation.

Willcocks, L. and Lester, S.,
Information systems investments: evaluation at the feasibility stage of projects,
Technovation Volume 11 No 5, 1991.

This article handles the shortage of information on how organisations go about the critical task of evaluating the feasibility of IS projects.

Wolff, G.J.,
The Management of Risk in System Development: 'Project SP' and the 'New Spiral Model',
Software Engineering Journal. May 1989.

In this article is an example described where Boehm's Spiral Model has been tried. In the last part of the article, a notation called "Project SP" is presented as a means of recording the progressively growing knowledge base of a project and the areas of uncertainty and associated risk.

Young, P.H.,
FRISK - Formal Risk Assessment of system cost estimates,
AIAA 1992 Aerospace Design Conference, Irvine, California.

The Formal Risk (FRISK) method is an analytical, rather than a Monte Carlo based, cost-risk model. FRISK evaluates the total cost-contribution of a system design, given its Work Breakdown Structure.

APPENDICES

CONTENTS

APPENDIX 1: CHECKLIST

1. HUMAN RISK SOURCE AREAS: CONTROL 87

2. HUMAN RISK SOURCE AREAS: MECHANISM 98

3. PROJECT RISK SOURCE AREAS: INPUT 106

4. PROJECT RISK SOURCE AREAS: CONTROL 113

5. PROJECT RISK SOURCE AREAS: MECHANISM 122

6. RISK SOURCE AREAS DURING SYSTEM USE 127

APPENDIX 2: FORMS

1. INTERVIEW COVER FORM 132

2. INTERVIEW RISK SOURCE FORM 133

3. RISK SOURCE SUMMARY FORM 134

1. HUMAN RISK SOURCE AREAS: CONTROL

1.1 POSITION
1.2 COMMITMENT
1.3 ORGANISATION
1.4 TEAM COMPOSITION
1.5 MANAGEMENT OF DECISIONS REGARDING THE PROJECT

1.1. POSITION

Description:
The formal and effective authority of the sponsor within his organisation.

Example:
- It often occurs that final responsibility for a project is not firmly identified before project start.
- If the project encounters problems it is difficult to identify a person with sufficient power in the organisation to get some unpopular measure (such as getting extra funding, securing co-operation from an unwilling department) realised.
- The existence of two or more sponsors might lead to conflicting lines of authority and 'buck passing'.

Extremes:
- high: sponsor not formally identified or having insufficient authority or access to resources
- low: known and of sufficient seniority

Questions:
1.1.1. Is known who the sponsor is
1.1.2. Is the sponsor aware of this fact
1.1.3. How many sponsors are there
1.1.4. Will the same sponsor be responsible for the entire project (continuity)
1.1.5. Is the position of the sponsor sufficiently senior to actively support the project
1.1.6. Is the position of the sponsor sufficiently senior to insure adequate funding

Phase:
Feasibility study

Responsibility:
Client

Relevant role:
Sponsor

Examples of measures:
- when obtaining authorisation the sponsoring organisation will have to take care of this problem. The measure is outside the span of control of the project manager but should be identified as soon as possible in order to give the proper signals to higher management (avoid)
- gain sufficient knowledge about the organisation and the problems of the sponsor (reduce)

1.2. COMMITMENT

Description:

Commitment indicates readiness to action and willingness to initiate the actions which are expected within the framework of the project.

Example:

- the sponsor does not voluntarily support the project.
- the user can be afraid that his position gets worse or that he might even lose his job, so he will not co-operate with the project or will even sabotage it.
- the contract manager wants to win the project and is willing to do everything to get it, but he forgets to assess the consequences for his own company.

Extremes:

- high:opposing
- low: supporting

Questions:

1.2.1. Is there a sound business case for the proposed system (costs/benefits)
1.2.2. Is the sponsor willing to spend time and money on a preliminary inquiry
1.2.3. Is the sponsor/ user manager capable of appreciating functional and non-functional requirements of the proposed product
1.2.4. Is the sponsor/ user manager willing to be closely associated to the project
1.2.5. Are these specifications agreed to in any formal way
1.2.6. Is the system important to the <role>
1.2.7. Is the proposed system intended to support a vital business process (primary activity) or is it for a supporting function
1.2.8. What are the consequences of project failure for the user organisation
1.2.9. Does the <role> think this is a fun project
1.2.10. Will many changes in the exercise of the task for direct users
1.2.11. Will the user encounter unfamiliar hardware
1.2.12. Are the targets seen to be realistic
1.2.13. Does the project fit in with the other activities of the <role>

Responsibility:

Both

Examples of measures:

sponsor

- gain sufficient knowledge about the organisation and the problems of the sponsor (reduce);
- make the method of working clear to the <user, user manager, sponsor> and gain his commitment to it (reduce);

- give references of projects in the same problem area. This can be done by means of a presentation, carried out by several people who will possibly execute the project. The objective with this is to gain the sponsors confidence. A consultant can be of help here (reduce);
- ask a group of users to give a formal presentation regularly during the project (supported by the supplier) and introduce parts of the end product in the meantime, in order to give the sponsor a clear sight of the end product in the making (reduce).

contract manager

- let the sponsor get in touch regularly with the contract manager. This can be arranged for example by taking up the contract manager in the steering committee (reduce);
- emphasise a win-win situation (reduce);
- put specific conditions down in the contract. These conditions concentrate on the contract manager, having to take specific actions (contract). For example his presence in regular project progress meetings is demanded.

user

- make clear to the users that the project has management commitment);
- take care that the user has enough time to co-operate within the project, e.g. by decreasing his operational task set (reduce);
- let users participate in the project and use methods which positively influence the communication with the users, such as prototyping (reduce);
- ask for help of the sponsor to stimulate the user co-operation (reduce);
- take up the stakeholders of the users group in the composition of a user reference group (reduce).

project manager

- assess the project manager's performance based on project budget (hours, costs), not commercial price (reduce);
- hold out the prospect of a bonus to the project manager when the project has been performed within time and budget (reduce);
- give the project manager more freedom and possibilities for the composition of his project team (reduce).

edp staff

- open communication in the project team for example by regular consultation (reduce);
- make the tasks and responsibilities clear for each member of the project team (reduce);
- replace non-motivated people by motivated people (reduce);
- increase rewards, put rewards on milestones (special attention for social events) (reduce).

Relevant roles and phase:

Role:	Risk source linked to role	First phase in which the risk source can assert its influence
Sponsor	X	feasibility study
User	X	functional design
User management	X	contracting phase
Client project management	X	functional design
Client edp staff	X	functional design
Supplier contract manager	X	functional design
Supplier project manager	X	functional design
Supplier edp staff	X	contracting phase

1.3. ORGANISATION

Description:

Characteristics of the organisational context within which the system will have to function which might influence the project. These characteristics exist independent of the proposed system. Differences in culture or in sophistication will be of importance here together with the existence of a (mis-)match between the type of organisation and the type of project.

Example:

- instability caused by large impending reorganisations
- lack of IT-experience within the organisation
- a bureaucratic culture that is insensitive to change
- a previous unfortunate experience with a failed IT-project

Extremes:

- High: a stable organisation, with a high degree of sophistication in its business processes and IT-deployment
- Low: an unstable organisation on a low IT-level

Questions:

1.3.1. How old is the organisation
1.3.2. What is the organisations readiness for change
1.3.3. What is the predominant culture (e.g. commercial, financial, engineering)
1.3.4. Is the environment in which the system will have to function susceptible to change
1.3.5. What is the education level of the users
1.3.6. Are we dealing with a very formal user organisation
1.3.7. How many different department are going to use the system
1.3.8. If we dealing with several user organisations do they have conflicting cultures

Responsibility:

client

Examples of measures:

- try to dim the wishes and requirements of user and sponsor by taking not to large steps and by pointing out the danger of making a leap forwards that might be too much at a time (reduce);
- deliver the product in different parts according to a carefully timed schedule. As the overall system is getting more and more advanced in time, the users are able to gradually grow accustomed to the higher level of automation (reduce);

- introduce simple and effective IT-solutions for simple problems (such as word processing of agenda support) in order to improve the attitude towards IT-solutions (avoid)
- start a 'private-PC' project in which staff is encouraged to buy cheap PC's for home use in order to improve the attitude towards IT-solutions (avoid)

Relevant roles and phase:

Role:	Risk source is relevant for role:	Phase in which risk source can become active (per role)
Sponsor		
User	X	functional design
User management		
Client project management		
Client edp staff		
Supplier contract manager		
Supplier project manager		
Supplier edp staff		

1.4. TEAM COMPOSITION

Description:

The composition of a project team determines its ability and the motivation of its members to solve problems in an effective and efficient way within the framework of the agreements (planning, budget and formerly defined results). The project team ideally has:

- the 'right' mix of personality types
- one goal to which all participants subscribe

Example:

- some team members want a 'perfect' system while others want to finish as soon as possible, thus causing a conflict of interest
- a chaotic person is expected to work closely together with an extremely tidy person

Extremes:

- High: a balanced organisation with sufficient representation of all required skills and experience
- Low: potential role conflicts built into the team organisation, or a discrepancy exists between personal ambitions and formal roles, or incompatible differences between individuals

Questions:

1.4.1. are many organisation represented in the team
1.4.2. are many different organisation cultures represented in the team
1.4.3. is the project members view on the proposed system identical
1.4.4. are conflicting goals represented in the team
1.4.5. is there an imbalance as to knowledge or commitment between the team members
1.4.6. do conflicting work methods exist

Phase:

functional design

Responsibility:

both

Examples of measures:

- set goals well in advance and make sure that all team members subscribe to them (avoid)
- try to improve the communication between team members by increasing the consultation frequency or reducing the group (reduce);

- assign resources for 'team-building' (reduce)
- when the team composition promises to be difficult choose a project manager with excellent interpersonal abilities (reduce)

1.5. MANAGEMENT OF DECISIONS REGARDING THE PROJECT

Description:

The quality of the decision making process of management not directly involved in the operational activities (i.e. the user management, sponsor, contract manager roles, which have an impact on the project's progress and the problem resolution potential.

Example:

- Can it be guaranteed that essential decisions regarding the project which are outside the teams competence will be taken timely
- in between phases the approval of milestone products is slow

Extremes:

- High: management is not capable of taking decisions on time in the correct way, when this is necessary either to resolve problems, or at intermediate milestones during the project
- Low: decisions are adequately taken when necessary not causing any disruption in the project's progress

Questions:

1.5.1. is the project likely to be hindered by a delay in external approval
1.5.2. are procedures in place that handle these decisions
1.5.3. are these procedures in place in more locations within the organisation
1.5.4. is there agreement as to how to act when essential decisions are not made

Phase:

contracting phase

Responsibility:

client

Examples of measures:

- lay down a general decision procedure in the contract including the consequences for the different parties involved (contract);
- hand over a project planning in which milestones and expected decision moments are indicated. This planning has to be signed by sponsor, user and project manager (reduce);
- make agreements about consequences when no decision is taken (project progress stops). For example call on an external expert or take up a penalty clause (contract).

2. HUMAN RISK SOURCE AREAS: MECHANISM

2.1 IT-KNOWLEDGE AND EXPERIENCE
2.2 DOMAIN AREA KNOWLEDGE AND EXPERIENCE
2.3 AVAILABILITY
2.4 ORGANISATIONAL SUPPORT

2.1. IT-KNOWLEDGE AND EXPERIENCE

Description:

Has the right type of knowledge and experience on relevant IT-areas been incorporated into the project organisation. Is this knowledge available in principle.

Example:

- the users have never worked with a computer system and do not know what they can expect
- the drivers of the old software seems not be correct, new ones have to be developed in three days. However nobody of the project team has enough experience to develop drivers.
- users have worked with mainframe applications but have no experience with X-windows; they have a wrong set of expectations
- the edp staff have never before designed a client-server application although they are versed in the theoretical aspects
- knowledge on the type of communication infrastructure required is lacking

Extremes:

- High: insufficient knowledge and experience
- Low: sufficient knowledge and several years of experience

Questions:

2.1.1. Is a project manager of sufficient seniority available
2.1.2. Is edp-staff of sufficient seniority available
2.1.3. Are the user organisation and user management computer minded
2.1.4. How many IT-applications are operational in the user organisation
2.1.5. Has the user organisation or department had previous experience with software development
2.1.6. Have these experiences been positive or negative
2.1.7. Is there resistance to change in the method of information processing within the user organisation
2.1.8. how much experience has the <role> in managing IT-projects
2.1.9. how much experience has the <role> in IT-projects
2.1.10. how much experience has the <role> in for this type relevant IT-knowledge areas

Responsibility:

Both

Examples of measures:

- buy experience, for example by recruiting a subcontractor of by hiring experienced staff (avoid);
- widen the budget so better people can be selected (compensate);

- hire the right people from another part of the company, when they are not present in your own department or from outside of the company (reduce);
- gain experience by working as a subcontractor and to make another supplier main contractor (avoid).
- invest in training (avoid)

Relevant roles and phase:

Role:	Risk source is linked to role:	First phase in which the risk source can assert its influence
Sponsor		
User	X	functional design
User management	X	contracting phase
Client project management	X	functional design
Client edp staff	X	functional design
Supplier contract manager		
Supplier project manager	X	functional design
Supplier edp staff	X	contracting phase

2.2. DOMAIN AREA KNOWLEDGE AND EXPERIENCE

Description:

Has the right type of knowledge and experience on the subject matter of the area to be supported by the new system (user-domain) been incorporated into the project organisation. Is this knowledge available in principle.

Example:

- the users are well educated people and know everything about their job, but are not able to communicate this knowledge to the edp-staff
- the jargon that is used by the users is not clearly defined which causes communication problems
- the users that participate in the project have insufficient expertise
- edp-staff have no previous experience with the very complex rules that govern the subject area in the application of social benefit regulations

Extremes:

- High: insufficient knowledge and experience
- Low: sufficient knowledge and several years of experience

Questions:

2.2.1. What is the status of the participating users (are they experts)
2.2.2. Is sufficient domain knowledge available (not only procedural knowledge indicating 'what' is to be done but also basic knowledge indicating 'why' it is being done)
2.2.3. Is sufficient insight into future developments available
2.2.4. How much domain knowledge has to be transferred from user to edp-staff
2.2.5. Can any communication barrier be identified between users and the edp-staff (jargon)
2.2.6. Is it possible to collect additional subject matter knowledge without too much effort

Responsibility:

Client

Examples of measures:

- assign knowledgeable staff (avoid)
- buy experience, for example by recruiting a subcontractor (avoid);
- gain experience by yourself by working as a subcontractor and to make another supplier main contractor (avoid).
- widen the budget so better people can be selected (compensate);
- hire the right people from another part of the company or from outside the company, when they are not present in your own department (reduce);
- invest in training (avoid)

- improve the communication between edp-staff and users by means of combined meetings and/or training sessions (reduce)

Relevant roles and phase:

Role:	Risk source is linked to role:	First phase in which the risk source can assert its influence
Sponsor		
User	X	functional design
User management	X	contracting phase
Client project management	X	functional design
Client edp staff	X	functional design
Supplier contract manager		
Supplier project manager	X	functional design
Supplier edp staff	X	contracting phase

2.3. *AVAILABILITY*

Description:

Is sufficient knowledgeable staff of the required role available at the moment they are necessary to support execution of the project. Is this expertise available at the right moment.

Example:

- the knowledgeable users are on a trip when the specifications have to be discussed
- the only user with sufficient knowledge is unavailable on a structural base because of other, more urgent duties
- a project manager is involved in several projects and assigned too low a priority to this project
- a key member of the project team accepts a position elsewhere

Extremes:

- High: no guarantee of availability of staff is given
- Low: availability is agreed upon in accordance with the plan

Questions:

2.3.1. How many staff from the user organisation will be seconded to the project
2.3.2. What is the degree of commitment of these staff to the project
2.3.3. Are all identifiable user parties involved in the project
2.3.4. Have agreements been concluded as to the availability of staff from these user groups
2.3.5. How large a fraction of the capacity of this manager is spent on this project
2.3.6. Is this project the most important activity for the project manager
2.3.7. Has a capacity plan for the development of the system been delivered; in other words, has been set down who will be needed when for how long a period
2.3.8. What percentage of the system is developed by a third party
2.3.9. How many staff spent more then 50% of their capacity on the project
2.3.10. Have agreements been concluded as to the availability of edp staff
2.3.11. What is the probability of the project continuity being endangered because of staff transfers

Responsibility:

Both

Examples of measures:

- exempt staff from part of their normal duties in order to insure availability (avoid)
- prevent fragmentation of effort by assigning a limited number of tasks (avoid)
- if key personnel is thinking of moving then:
    - choose another staff member from the start (avoid)

- insure his present by means of a contract (agree)
- assign an 'assistant-to' to insure continuity (compensate)

Relevant roles and phase:

Role:	Risk source is linked to role:	First phase in which the risk source can assert its influence
Sponsor		
User	X	functional design
User management		
Client project management	X	functional design
Client edp staff	X	functional design
Supplier contract manager		
Supplier project manager	X	functional design
Supplier edp staff	X	contracting phase

2.4. ORGANISATIONAL SUPPORT

Description:
The degree to which the organisation as a whole is likely to support the project.

Example:
- an earlier project failed miserably, thus causing an animosity towards all new developments
- the system is likely to reduce the amount of jobs

Extremes:
- High: high level of animosity towards the project
- Low: positive attitude towards the project

Questions:
2.4.1. How long has the idea for this system been in existence in the organisation
2.4.2. Is there a clear reason to start the project at this moment
2.4.3. Has the project had a long politically charged history

Phase:
feasibility study

Responsibility:
client

Examples of measures:
- state goals clearly (avoid)
- involve all relevant parties (reduce)

3. PROJECT RISK SOURCE AREAS: INPUT

3.1 CLARITY OF SPECIFICATIONS
3.2 STABILITY OF SPECIFICATIONS
3.3 COMPLEXITY
3.4 DEGREE OF INNOVATIVENESS
3.5 SIZE
3.6 SUB-CONTRACTOR PERFORMANCE

3.1. CLARITY OF SPECIFICATIONS

Description:

The degree to which to specifications are unambiguously defined and accepted by the parties involved

Example:

- the system is to perform the functions of the old system; in fact considerable additional requirements have to be included
- no thought has been given to security and privacy considerations

Extremes:

- High: unclear specifications
- Low: specifications are based on the outcomes of an information analysis which does not leave any ambiguity as to the functional contents of the system under development

Questions:

3.1.1. are the specifications unambiguously defined (on paper)
3.1.2. will determining the specifications cause problems
3.1.3. can you assess the degree of completeness of the specification
3.1.4. to what degree will this system replace an existing automated solution
3.1.5. are parts of the system already in existence
3.1.6. are the specifications known to all parties involved
3.1.7. are the specifications accepted by all parties involved
3.1.8. has this acceptance been ratified officially

Phase:

functional design

Responsibility:

both

Examples of measures:

- formulate the acceptance test criteria in the contract (contract).
- use prototyping in case of unclear user-visible requirements
- organise a joint discussion on the specification document in order to further consensus
- use an evolutionary delivery strategy in order to be able to redress differences in opinion

3.2. STABILITY OF SPECIFICATIONS

Description:
The degree to which previously agreed specifications are likely to shift during project execution

Example:
- because market and technology (in the environment of the customer) change continuously (via reorganisation), the project team might face changing requirements during the project.

Extremes:
- High: the environment is unstable
- Low: specifications have been agreed by key users, representing the entire user community and are in line with the future systems architecture of the company, or simply allow for future systems evolution through anticipating on changes in requirements

Questions:
3.2.1. what are the odds that specifications will change during project execution
3.2.2. have agreements been concluded with users/ sponsor as to a 'freeze' of specifications
3.2.3. have agreements been concluded with users/ sponsor as to the extra costs that accompany changes or additions to the specification
3.2.4. has a diversity of user types been anticipated
3.2.5. have changes in the business processes been anticipated

Phase:
functional design

Responsibility:
client

Examples of measures:
- divide the end product in parts and chose for an appropriate method for each part such as incremental development (reduce);
- minimise the throughput time of the project for example by putting more people on it. This will however increase the communication and co-ordination efforts (compensate);
- divide the project in smaller projects, executed successively in time (reduce).

3.3. COMPLEXITY

Description:

The degree of complexity is the degree in which user and/or supplier see the proposed system and the associated project as complex. Complexity as such is thus defined on a subjective basis as related to the experience of those who are involved in the project. The perceived complexity for an individual can be considered as a function of the number of factors and their mutual relationships experienced in a particular decision making situation. In a software engineering environment typical examples of these factors are the number of different departments involved in developing the system, the number of programs or interfaces to be developed, size of the project team in terms of numbers of people.

Examples:

- because the user has no idea about the outcome of his decisions for the system which has to be developed, he will postpone decisions.
- users are unable to maintain an overview over the specification

Extremes:

- High: high complexity
- Low: low complexity

Questions:

3.3.1. do you have an estimate of systems complexity
3.3.2. are you able to indicate the complicating factors for this system
3.3.3. do you have experience in building systems of this degree of complexity

Phase:

functional design (functional complexity)
technical design (technical complexity)

Responsibility:

both

Examples of measures:

- Use an available model as reference e.g. from standard software (reduce).
- Divide the overall project in separate smaller sub-projects which have none or limited interrelationships, to improve overall controllability (reduce).
- Decrease the need for integration by modular designing and by connecting the modules by means of (manual) interfaces (reduce).
- Reduce the ambition level of the proposed system (avoid).

3.4. DEGREE OF INNOVATIVENESS

Description:

Degree to which the systems design incorporates functions or other requirements that are new either in an absolute sense or in relation to the experience of the people involved in developing the system.

Example:

- the system is to support end users at the workplace; this type of system is new for the organisation
- the organisation tries to implement a voice response system in the customer service department

Extremes:

- High: never attempted before
- Low: familiar type

Questions:

3.4.1. do you have experience with this type of system
3.4.2. does this system for you have innovative aspects
3.4.3. can we generally speak of an innovative system

Phase:

Technical design

Responsibility:

both

Examples of measures:

- engage outside staff who have encountered this type of problem before (compensate)
- reduce the ambition level of the proposed system (avoid).
- increase available budget and schedule (reduce)
- do some prototyping to test the degree to which the new aspects can be realised (reduce)

3.5. SIZE

Description:
Size of system development project, either in time or in effort and of the proposed system

Example:
- a fifty man year effort for a department whose largest project up to date was 5 man year

Extremes:
- High: of a much larger magnitude than previously experienced
- Low: well within the range of experience

Questions:
3.5.1. is an estimate of the size of the system available
3.5.2. have you ever participated in the development of a system of this size
3.5.3. is an estimate of the project lead-time available
3.5.4. have you ever participated in a project with a comparable lead-time
3.5.5. Is the project controllable (lead-time less then 2 year)

Phase:
functional design

Responsibility:
both

Examples of measures:
- opt for evolutionary delivery in order to obtain manageable chunks of work (avoid)
- reduce the ambition level of the proposed system (avoid).
- hire external expertise with development of this size and scope (avoid)

3.6. SUBCONTRACTOR PERFORMANCE

Description:

The subcontractor's performance is defined by the degree in which the subcontractor fulfils the agreements made with regard to the specific project.

Example:

- The development of an E-mail system has been put out to a subcontractor. He has to deliver the E-mail system on an agreed date. Ate the day of delivery the subcontractor reports that he needs another week to finish the system
- The system is delivered on time, but is shows several important bugs

Extremes:

- High: many subcontractors without any prior co-operation experience
- Low: no subcontractors

Questions:

3.6.1. what is your (positive or negative) experience with this particular subcontractor
3.6.2. what is known of the time reliability of the subcontractor
3.6.3. to what degree is the subcontractor familiar with your type of organisation
3.6.4. how many subcontractors are involved in the project

Phase:

realisation

Responsibility:

supplier

Examples of measures:

- agree on procedures in the contract for not keeping agreements (contract);
- use subcontractors with good references (reduce).

4. PROJECT RISK SOURCE AREAS: CONTROL

4.1 DEMARCATION OF THE PROJECT
4.2 EXTERNAL CONDITIONS
4.3 PROJECT PLAN
4.4 SLACK (CONTRACTUAL AND OTHERWISE)
4.5 MONITORING
4.6 POSITION PROJECT APPROACH IN THE ORGANISATION
4.7 POSITION OF QUALITY MANAGEMENT IN THE ORGANISATION
4.8 INTERDEPENDENCIES

4.1. DEMARCATION OF THE PROJECT

Description:

The demarcation of the project is given with respect to the content of the project determined by its objectives on the one hand and on the other hand by organisational, technical and financial constraints.

Example:

- the demarcation of the project fails to indicate clearly what to do with the interfaces between the new and the old system.
- when replacing an existing system is it unclear if just replacement is the goal of if extra functionality is required
- no cover-all architecture for the project has been defined

Extremes:

- High: unclear
- Low: based upon an overall plan and architecture, in line with the company's latest statement of business strategy and development plans

Questions:

4.1.1. are the contents of the project clear
4.1.2. which are related systems
4.1.3. have interface been defined
4.1.4. is there a match between the project scope and the business responsibility area

Phase:

feasibility study

Responsibility:

client

Examples of measures:

- set up a preliminary investigation, executed for example by a consultant (avoid);
- choose standard software and use it as a reference model. In standard software usually clear demarcations are given of application areas (reduce);
- use a model, developed from similar projects, in a similar environment (reduce);
- execute the first three phases of the project on basis of subsequent calculation and then close a fixed price contract for the following phases (avoid);
- specify the basic operations the software will have to perform and arrange to meet a fixed number of reports and/of inquiries per function or arrange that extra functionality can be added to the software based on specific arrangements which have to be agreed upon during the progress of the project (reduce).

4.2. EXTERNAL CONDITIONS

Description:
Constraints which are dictated to the system by the external environment.

Example:
- the auditor dictates conditions to the system which have to be developed from the viewpoint of internal control. The user often takes it for granted that the supplier knows about these conditions.
- Changes in customs regulations can necessitate a different administrative approach
- New government regulations regarding the environment
- EDP-links that set their own demands

Extremes:
- High: a high dependency on an unstable environment
- Low: the project as defined can be considered not to be affected by changes in its environment during its planned lead time

Questions:
4.2.1. will the system function in a network environment
4.2.2. will the system be integrated with other systems
4.2.3. are restrictions imposed on the system from outside the user organisation

Phase:
functional design

Responsibility:
client

Examples of measures:
- involve an expert on the problem field in the project (reduce);
- have the specifications signed by specialists to make sure that the external conditions have been thoroughly examined (avoid);
- introduce a specialist in the user-reference group (reduce);
- take care of sufficient knowledge on rules and legislation in a specific line of business and its function in the company (reduce).

4.3. PROJECT PLAN

Description:

Plan of activities, mutual interdependencies and to be delivered products that together will result in the desired system

Example:

- no project plan is available
- the level of detail of the project plan stops at phase level
- participants are not consulted

Extremes:

- High: no explicit plan exists, the project proceeds in an incremental way based on the outcome of previously completed steps
- Low: complete, detailed and accepted plan

Questions:

4.3.1. has a project plan been drawn up
4.3.2. does the plan include a work breakdown structure
4.3.3. does the plan include a staff task allocation
4.3.4. have productivity factors been taken into account
4.3.5. has the plan been prepared in consultation with the involved parties
4.3.6. do you believe in the plan
4.3.7. is the schedule realistic

Phase:

Contracting phase

Responsibility:

both

Examples of measures:

- hand over a project planning in which milestones and expected decision moments are indicated. This planning has to be signed by sponsor, user and project manager (reduce);

4.4. SLACK (CONTRACTUAL AND OTHERWISE)

Description:
The amount of leeway that is available within the project planning

Example:
- project schedule is too tight
- project schedule is too loose
- project schedule and the quality required are fixed, but no account is taken of the existing high degree of technical uncertainty

Extremes:
- High: unconstrained, the project can take any amount of time, resources or money in order to finish; or: over constrained, no degrees of freedom left
- Low: acceptance that some leeway might be needed

Questions:
4.4.1. is the project fixed price, fixed time and fixed quality
4.4.2. does the relation with the sponsor allows for changes in agreed budget, schedule or quality
4.4.3. is the time of systems delivery fixed
4.4.4. is the project schedule tight
4.4.5. is the project schedule too loose
4.4.6. Is the degree of slack allowed in accordance with the perceived risks
4.4.7. Is the degree of slack allowed in accordance with the degree of uncertainty that surrounds the project

Phase:
functional design

Responsibility:
both

Examples of measures:
- allow for the right amount of leeway required in the project budget (avoid)
- allow for the right amount of leeway required in the project schedule (avoid)
- allow for the right amount of leeway required in the product quality requirements (avoid)

4.5. MONITORING

Description:

Are procedures in place to monitor, evaluate and influence progress in terms of on-going activities and deliverables of these activities. It should be noted that monitoring can only effectively be carried out when a formal plan is in place and is well understood by all involved parties.

Example:

- time monitoring takes place via the financial department with a turnaround time that exceeds two weeks
- procedures for progress control are rudimentary and as a consequence are ignored
- milestone acceptance is in place but functions as a 'rubber stamp' procedure

Extremes:

- High: no regard for monitoring
- Low: monitoring procedures in place and accepted

Questions:

4.5.1. are clear agreements in place as to which intermediate products have to be delivered

4.5.2. are clear agreements in place as to when these intermediate products have to be delivered

4.5.3. are clear agreements in place as to who has to approve of these intermediate products and how long this may take

4.5.4. is the culture in the development organisation such that official reports give an accurate picture of the current situation

Phase:

contracting phase

Responsibility:

both

Examples of measures:

- put down the moment of acceptance in the planning and have this planning signed (contract);
- hand out information aimed at the management, facilitating the decision taking process (reduce);
- make use of techniques as prototyping (reduce);
- make use of schedules and other graphical reproductions (reduce);
- have the intermediate results and planning for the next phase signed by the user group and the sponsor (reduce).

4.6. POSITION PROJECT APPROACH IN THE ORGANISATION

Description:
The compliance to and experience with formal project management methods

Example:
- the organisation is not used to the discipline required by a project approach

Extremes:
- High: no experience with a project style of work
- Low: sufficient experience with a project style of work

Questions:
4.6.1. Is the organisation used to working in projects
4.6.2. Is time set aside by the users for participation in the project
4.6.3. Is a project management method routinely used for all projects
4.6.4. Is the method 'alive and kicking' within the organisation
4.6.5. Is the method used to promote a project style of work or to satisfy external demands

Phase:
functional design

Responsibility:
client

Examples of measures:
- apply widely used methods and techniques (reduce);
- hire people who have the knowledge and experience with a specific method (reduce);
- look at the choice with regard to the method from the customer point of view (reduce);

4.7. POSITION OF QUALITY MANAGEMENT IN THE ORGANISATION

Description:

The organisation's awareness of and focus on quality, as expressed by the professional attitude of the project's participants and through formal quality management practices being applied in the organisation (e.g. ISO 9000 certified)

Example:

- the notion of software quality is not discussed in the organisation causing individual staff members to work according to their individual insights

Extremes:

High: high quality awareness
Low: low quality awareness

Questions:

4.7.1. Is a quality management system in place
4.7.2. Is system routinely used for all projects
4.7.3. Is the method 'alive and kicking' within the organisation
4.7.4. Is the method used to promote quality or to satisfy external demands

Phase:

functional design

Responsibility:

both

Examples of measures:

- apply widely used methods and techniques (reduce);
- hire people who have the knowledge and experience with a specific method (reduce);
- look at the choice with regard to the method from the customer point of view (reduce);
- involve people from quality assurance (if available) in the choice of a method and technique (reduce);
- work according to ISO 9000 standards (reduce).

4.8. INTERDEPENDENCIES

Description:

Interdependencies occur when the project for its success is dependent on activities of people outside its scope.

Example:

- the system has to be integrated with another system that is being developed concurrently
- the hardware environment is being developed simultaneously
- part of the data is being supplied by a third party on a voluntary basis
- the system is built, but due to a re-organisation it will never be used

Extremes:

High: high
Low: low

Questions:

4.8.1. is the project dependent on the progress of other projects
4.8.2. if so, is the other project progressing according to plan
4.8.3. is the other project a high risk project
4.8.4. is the project dependent on other not involved parties
4.8.5. are these other parties sufficiently motivated to do their part

Phase:

contracting phase

Responsibility:

both

Examples of measures:

- try to improve the motivation of these external parties (avoid)
- allow for these occurrences in the project schedule so that responsibility is allocated to the appropriate parties (agree)

5. PROJECT RISK SOURCE AREAS: MECHANISM

5.1 SUITABILITY OF WORKING CONDITIONS
5.2 HARDWARE
5.3 SOFTWARE AND TOOLS
5.4 USE OF METHODS AND TECHNIQUES

5.1. SUITABILITY OF WORKING CONDITIONS

Description:

The suitability of the working conditions regards quality and availability of equipment, facilities and of the working environment within which the team will have to function

Example:

- the necessary computer turnaround time does not seem to comply with the expectations
- shabby offices are all that is available
- no secretarial support is available
- tools required are delivered late

Extremes:

- High: not sufficient to execute the project according to plan and budget
- Low: sufficient to execute the project as planned and budgeted

Questions:

5.1.1. Is a separate working area set aside for the team
5.1.2. Are sufficient supporting facilities available
5.1.3. Are working conditions pleasant

Phase:

functional design

Responsibility:

both

Examples of measures:

- put the minimum requirements for the working conditions down in a contract (contract);
- describe penalty clauses in the contract, in case the environment does not meet the requirements (contract);
- let somebody spend time and attention to this subject (avoid)
- assign a budget (avoid)

5.2. HARDWARE

Description:
The computing machinery to be used in the project (host and/or target environment).

Example:
- the hardware needed has been maintained badly.
- availability of suitable testing environment of the target is low
- a new type of target environment is envisaged for which limited expertise is available
- the system has to be developed on hardware that supports operational systems

Extremes:
- High: no hardware
- Low: sufficient hardware

Questions:
5.2.1. is new of unfamiliar hardware required for developing the system
5.2.2. what level of hardware experience is present in the team
5.2.3. is a suitable development environment available

Phase:
technical design

Responsibility:
both

Examples of measures:
- apply widely used hardware (reduce);
- hire people who have the knowledge and experience with specific hardware (reduce);
- plan extra time to act on new hardware (compensate);

5.3. SOFTWARE AND TOOLS

Description:

The software and software tools to be used and re-used during the project. These can be short in supply but also lack of experience may cause trouble..

Example:

- The tools used do not seem to comply with the expectations
- A new case tool is introduced
- A new version of a database management system is introduced
- a structured library of system modules is available
- previous development resulted in insufficient documented products which if used would endanger the new project

Extremes:

- High: new tools
- Low: familiar tools

Questions:

5.3.1. is a standard set of development tools available for the project team
5.3.2. are advanced tools used in developing the system (generators)
5.3.3. are specific/new tools for realising this system
5.3.4. to what degree is made use of standard software
5.3.5. what degree of development tool experience is available in the team
5.3.6. to what degree is software re-used
5.3.7. do you have any idea what the level of quality of this software is

Phase:

technical design

Responsibility:

both

Examples of measures:

- apply widely used tools (reduce);
- hire people who have the knowledge and experience with a specific tool (reduce);
- plan extra time to act on new tools (compensate);

5.4. USE OF METHODS AND TECHNIQUES

Description:
The use of methods and techniques indicates to what extent a common vehicle for communication can be created between the collaborating parties.

Example:
- The client requires development using the Coad/Yourdon OO methodology which the supplier has never done before

Extremes:
- High: little experience and new methods
- Low: all parties have considerable experience with the methods and techniques to be used throughout the project

Questions:
5.4.1. is a standard set of development methods available for the project team
5.4.2. are structured methods used in developing the system
5.4.3. is it possible to develop the system while using the available methods
5.4.4. what degree of development method experience is available in the team

Phase:
functional design

Responsibility:
both

Examples of measures:
- apply widely used methods and techniques (reduce);
- hire people who have the knowledge and experience with a specific method (reduce);
- plan extra time to act on new methods and techniques (compensate);
- look at the choice with regard to the method from the customer point of view (reduce);
- involve people of Quality Innovation in the choice of a method and technique (reduce);
- work according to ISO 9000 standards (reduce).

6. RISK SOURCE AREAS DURING SYSTEM USE

6.1 SYSTEM SUPPORT ORGANISATION
6.2 SYSTEM MAINTENANCE
6.3 DATA CONVERSION

6.1. SYSTEM SUPPORT ORGANISATION

Description:

The activities needed to educate future users and to support them in using the system which have to be provided for during project execution

Example:

- no supplier staff time has been set aside for after sales service
- no client staff is available after the end of the project for user support

Extremes:

- High: not considered
- Low: embedded

Questions:

6.1.1. are plans being drawn for staff and facilities availability for training during introduction and use of the system

6.1.2. are plans being drawn for staff and facilities availability for user support (e.g. a help desk)

6.1.3. are plans being drawn for staff and facilities availability for maintaining the technical infrastructure of the system

6.1.4. are these activities part of the project plan

6.1.5. has responsibility for them been assigned

Phase:

implementation/ maintenance

Responsibility:

both

Examples of measures:

For this risk source no measures will be presented since they will fall outside the scope of the project and not hinder project execution and/or progress.

6.2. SYSTEM MAINTENANCE

Description:

The degree in which the system can be easily enhanced to accommodate evolving user requirements. This usually depends on how well the architecture of the system has been designed, the quality of the technical system documentation and the quality of the training given to the people who are supposed to support the system in its operational phase.

Example:

- no structured record of user complaints is kept resulting in haphazard maintenance
- no resources are set aside for maintenance

Extremes:

High: the system can be easily enhanced, as it is based on a sound architecture which is well understood by all technical staff

Low: enhancing the system requires excessive effort; the result of an insufficient architecture and a lack in training for support staff

Questions:

6.2.1. are staff and facilities available to adapt the system to changing requirements during the foreseen life span
6.2.2. what is the speed with which requirements are likely to change
6.2.3. is this subject covered in the project plan
6.2.4. are documentation standards present
6.2.5. is maintenance a design issue

Phase:

implementation/ maintenance

Responsibility:

both

Examples of measures:

For this risk source no measures will be presented since they will fall outside the scope of the project and not hinder project execution and/or progress.

6.3. DATA CONVERSION

Description:
The degree in which the designers have anticipated on the conversion of data from existing systems to the new system.

Example:
- the system used many volatile data, no resources have been allocated to updating these data
- the existing data are badly polluted

Extremes:
- High: not considered
- Low: embedded

Questions:
6.3.1. are staff and facilities available for entering the required data
6.3.2. how much effort is needed for converting, updating and entering system data
6.3.3. how reliable are these data

Phase:
implementation/ maintenance

Responsibility:
both

Examples of measures:
For this risk source no measures will be presented since they will fall outside the scope of the project and not hinder project execution and/or progress.

APPENDIX 2: FORMS

To support the use of the checklist some forms have been designed. These are:

1. A cover form
   This form serves to capture all information gathered during an interview that is not directly related to the risk sources
2. A risk source form
   This form should be made for each risk source individually. It serves to capture all relevant information that is directly related to each risk source.
3. An overview form
   This form can be used to summarise for each risk source the information that was gathered during the interviews.

Samples of these forms can be found on the next pages.

1. Interview cover form

Project:		Date:	
Interviewed person:		Interview by:	
Position of interviewed person:			
Main risks (before using checklist:			
Risks not mentioned in checklist:			
Main risks (after using checklist):			
Comments:			

2. Interview risk source form

Risk source:			
Probability:		Impact:	
	1) negligible		1) any impact is negligible or can at least be handled without problems within the present budget,
	2) medium		2) the impact can not be handled within the existing means, but does not endanger the project,
	3) high		3) the impact endangers the success of the project.
	4) very high		
Question 1			
Question 2			
Question 3			
Question 4			
Question 5			
Conclusion:			
Comments:			

3. Risk source summary form

Risk source:										
Respondent:	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Probability	O O O O 1 2 3 4	O O O O 1 2 3 4	O O O O 1 2 3 4	O O O O 1 2 3 4	O O O O 1 2 3 4	O O O O 1 2 3 4	O O O O 1 2 3 4	O O O O 1 2 3 4	O O O O 1 2 3 4	O O O O 1 2 3 4
Impact	O O O 1 2 3	O O O 1 2 3	O O O 1 2 3	O O O 1 2 3	O O O 1 2 3	O O O 1 2 3	O O O 1 2 3	O O O 1 2 3	O O O 1 2 3	O O O 1 2 3
Question 1										
Question 2										
Question 3										
Question 4										
Question 5										
Comments										

Eindhoven University of Technology
Graduate School of Industrial Engineering and Management Science
Research Reports (EUT-Reports)

The following EUT-Reports can be obtained by writing to:
Eindhoven University of Technology, Library of Industrial Engineering and Management Science, Postbox 513, 5600 MB Eindhoven, Netherlands.
The costs are HFL 5.00 per delivery plus HFL 15.00 per EUT-Report (unless indicated otherwise), to be prepaid by a Eurocheque, or a giro-payment-card, or a transfer to bank account number 52.82.11.781 of Eindhoven University of Technology with reference to "Bibl.Bdk", or in cash at the counter in the Faculty Library.

20 LATEST EUT-REPORTS

EUT/BDK/86 Dealing with risk : beyond gut feeling : an approach to risk management in software engineering **F.J. Heemstra, R.J. Kusters, R. Nijhuis, Th.M.J. van Rijn**

EUT/BDK/85 The development of an incident analysis tool for the medical field **W. van Vuuren, C.E. Shea & T.W. van der Schaaf**

EUT/BDK/84 Operations management and financial management information systems : a design approach for infinite and finite planning systems **P.E.A. Vandenbossche**

EUT/BDK/83 Gordian project : final report July 1996 **R.J. van den Berg, A.J.R. Zwegers**

EUT/BDK/82 Incidents in accident and emergency & anaesthesia **Wim van Vuuren**

EUT/BDK/81 Dada en adviseren geeft dadaviseren **Matthieu Weggeman**

EUT/BDK/80 Critical success factors in developing 'accepted control loops' **Harrie van Tuijl**

EUT/BDK/79 Organisatie-diagnose via de kwaliteitsincidenten methode **J.D. van der Bij, T.W. van der Schaaf, P.M. Bagchus**

EUT/BDK/78 Kwaliteitsmanagement in de gezondheidszorg : een onderzoek naar huidige ontwikkeling en onderzoeksbehoeften in ziekenhuizen **T. Vollmar en J.D. van der Bij**

EUT/BDK/77 Het ene artikel is het andere niet! : een onderzoek naar de problemen omtrent de slechte afstemming tussen artikelstamgegevens in de levensmiddelenbranche **B. Vermeer**

EUT/BDK/76 Wegtransport : vitaal voor economie, welvaart en welzijn **J.P.M. Wouters e.a.**

EUT/BDK/75 Diagnosing the production organisation of SMES **M.J. Verweij**

EUT/BDK/74 Describing, analysing and designing with the production description language **M.J. Verweij**

EUT/BDK/73 Purchasing's development role : the internal and external integration of purchasing in technological development processes : intermediate report I **J.Y.F. Wynstra**

EUT/BDK/72 De problemen van hergebruik gezien vanuit de stofstromenproblematiek **A.J.D. Lambert**

EUT/BDK/71 Problemen en knelpunten bij gebruik van MRP in de praktijk : onderzoeksrapport **M.J. Euwe**

EUT/BDK/70 De groothandel is dood. Leve de groothandel! : een branchegericht onderzoek naar de toekomst van de groothandel en de rol van informatie technologie **M.J. Euwe**

EUT/BDK/69 Methodologies for information systems investment evaluation at the proposal stage : a comparative review **Th.J.W. Renkema, E.W. Berghout**

EUT/BDK/68 Software quality management : ISO 9000, but not only **K. Balla**

EUT/BDK/67 Thematiek en methodologie in de organisatiekunde : een inhoudelijke verkenning over de periode 1986-1991 op basis van onderzoek van enkele Nederlandse tijdschriften

J.D. van der Bij, J.A. Keizer

EUT/BDK/66 Naar een tweede generatie total quality management

J.D. van der Bij, J.E. van Aken

Eindhoven University of Technology
Faculty of Technology Management

P.O. box 513
5600 MB Eindhoven
The Netherlands
Phone +31 40 247 2873