Categorisation and formulation in risk management

-Essential parts of a future Experience based Risk Management model within software engineering

Peter Nilsson
Erik Ohlsson

Department of
Software Engineering and Computer Science
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden
This thesis is submitted to the Department of Software Engineering and Computer Science at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:

Authors:
Peter Nilsson
Address: Båtsmansgränd 4, 371 31 Karlskrona
E-mail: peter-nilsson@telia.com

Erik Ohlsson
Address: Torholmsvägen 11, 372 74 Listerby
E-mail: dwarn@telia.com

University advisor(s):
Conny Johansson
Department of Software Engineering and Computer Science

Department of Software Engineering and Computer Science
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

Internet : www.bth.se/ipd
Phone : +46 457 38 50 00
Fax : + 46 457 271 25
ABSTRACT

This software engineering thesis addresses three main issues. When creating the risk documents for this master thesis project, we became even more aware of the problems with categorization and formulation of risk statements and the scope is now focusing on categorization and formulation as a necessity for Experience based Risk Management (EbRM). The EbRM process is the foundation of the thesis and the categorisation and formulation parts had to be solved before implementing the EbRM model. To give the reader a notion about the background of this work, a brief introduction to the Experience based Risk Management model is given in the thesis. The thesis is based on literature studies, experiences and experiments.

The formulation system is gathered from the Software Engineering Institute (SEI) and is called the CTC-format (Condition, Transition, Consequence). This format allows you to separate conditions and consequence of the risk and thereby provides you with easier categorisation and understandability.

The categorisation system used is the SEI Taxonomy Based Categorisation (TBC). A categorisation system built as a search tree where each leaf represents a rather narrow risk domain. In order to evaluate those two different systems we performed an experiment showing that the combination thereof gave a much higher match in sorting risks between different groups.

The conclusions of this work are that the TBC in connection with the CTC structure forms a very good basis for risk management when it comes to categorisation and formulation. In addition to properly formulated and tagged names and a thorough process when identifying and documenting risks, the risk management will be facilitated by using our conclusions in further risk management. Oral information must as well be on a sufficient level to gain full benefits from a risk management process.

Keywords: Risk management, risk formulation, risk categorization, software engineering, documentation.
CONTENTS

ABSTRACT

CONTENTS

1 READERS GUIDE

2 INTRODUCTION

3 INTRODUCTION TO THE EXPERIENCE BASED RISK MANAGEMENT MODEL

4 INTRODUCTION TO CTC

5 INTRODUCTION TO TBC

6 EXPERIMENT

7 HOW TO FORMULATE RISKS

READERS GUIDE

1.1 GET AN INTRODUCTION TO THE EbRM MODEL

1.2 LEARN MORE ABOUT FORMULATING RISKS

1.3 LEARN MORE ABOUT HOW TO CATEGORISE RISKS

1.4 DICTIONARY

1.5 SUMMARY

INTRODUCTION

2.1 RISK DEFINITION

2.2 INTRODUCTION TO THE QUB-PROJECT

2.2.1 A project with high risk-exposure

2.2.2 Project outcome

2.3 MOTIVATION FOR THE SUBJECT

2.3.1 Survey

2.4 MOTIVATION FOR RISK MANAGEMENT IN SOFTWARE PROJECTS

2.5 PROBLEM FORMULATION

2.6 HYPOTHESES

2.7 METHOD DESCRIPTION

2.7.1 Literature study

2.7.2 Experiments

2.8 SUMMARY

INTRODUCTION TO THE EXPERIENCE BASED RISK MANAGEMENT MODEL

3.1 BETTER USE OF EXPERIENCES

3.2 MORE EFFECTIVE RISK MANAGEMENT

3.3 HOW EbRM WORKS

3.3.1 Predefined form

3.3.2 Updateable form

3.3.3 Project

3.3.4 Final project report

3.3.5 Software process improvements

3.4 SUMMARY

INTRODUCTION TO CTC

4.1 THE CTC CONCEPT

4.2 SUMMARY

INTRODUCTION TO TBC

5.1 THE SEI TB CATEGORISATION

5.1.1 The structure

5.1.2 The categories

5.2 SUMMARY

EXPERIMENT

6.1 RESULTS

6.2 SUMMARY

HOW TO FORMULATE RISKS

7.1 UNDERSTANDABILITY – THE ESSENCE OF FORMULATION

7.2 COMPLETENESS

7.2.1 Completeness: problems
1 READERS GUIDE

This guide gives you a hint of how to read this thesis in three different ways, depending on the information you are after. These are:

- Get an introduction to the Experience based Risk Management model (EbRM).
- Learn more about formulating risks.
- Learn more about how to categorise risks.

Of course, it is always better to read the whole thesis from first to last page.

1.1 Get an introduction to the EbRM model

The core reason for writing this thesis was to introduce the EbRM-concept, but on our way doing that we encountered some obstacles. These were:

- We have to have a standardised way for formulating the risks so that we can decrease the risk for misinterpretations.
- We have to have a well defined categorisation system that could manage a great growth of the risk register.

These two issues became the two big parts of this thesis, but there is still a chance of learning about the EbRM-model without reading about the formulation and categorisation of risks.

First read the “Risk definition” chapter 2.1, and then the “Method and Result” part of the “Survey” in chapter 2.3.1, this in order to gain knowledge about the needs for the model. If you want more motivation for why to improve the risk management within your organisation you should read chapter 2.4 “Motivation for risk management in software projects”. After this you read chapter 3 “Introduction to the Experience based Risk Management model”. This gives you an introduction to how EbRM shall work. As a complement you should read chapter 9.2 “Future works with the EbRM model” as well as the “Conclusion”, chapter 10.

1.2 Learn more about formulating risks

In order to be able to complete the work with the EbRM-model we had to accomplish deeper studies in how to formulate risks. If you only want to learn about our findings about how to formulate risks you should read this thesis as follows.

First read the “Risk definition” chapter 2.1, and then chapter 4 “Introduction to CTC”, this will give you the knowledge you need to fully understand chapter 7 “How to formulate risks” which is the next part for you to read. As a complement you should read chapter 9.1 “Future works with formulation and categorisation of risks” as well as the “Conclusion”, chapter 10.

1.3 Learn more about how to categorise risks

Additionally to the formulation part we needed to make some deeper studies in how to categorise the risks in order to come up with a well-working EbRM-model. If you only want to learn about our findings about how to categorise risks you should read this thesis as follows.

First read the “Risk definition” chapter 2.1, and then chapter 5 “Introduction to TBC”, this will give you the knowledge you need to fully understand chapter 8 “How to categorise risks” which is the next part for you to read. As a complement you should read chapter 9.1 “Future works with formulation and categorisation of risks” as well as the “Conclusion”, chapter 10.
### 1.4 Dictionary

To identify terms and abbreviations used in this master thesis, we will here try to give a brief description of the item and thereby facilitating the reading of this thesis without the reader having to consult references or further literature.

<table>
<thead>
<tr>
<th>Abbreviation/Expression</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTC</td>
<td>A general format for describing and formulate risk statements. The abbreviation stands for “Condition-Transition-Consequence”.</td>
</tr>
<tr>
<td>CTC – Condition</td>
<td>The current situation that might lead to a problem</td>
</tr>
<tr>
<td>CTC – Consequence</td>
<td>The negative outcome of a transition given certain conditions.</td>
</tr>
<tr>
<td>CTC – Transition</td>
<td>Change of situation that might lead to that the condition results in a negative result further on. Could also be referred to as time.</td>
</tr>
<tr>
<td>DoD</td>
<td>Department of Defence</td>
</tr>
<tr>
<td>EbRM</td>
<td>Experienced based Risk Management. A risk model that the authors a working on. This model will use old experiences from project work within the company to continuously improve the risk-handling.</td>
</tr>
<tr>
<td>Exposure</td>
<td>The total figure you get when multiplying impact and probability of a risk</td>
</tr>
<tr>
<td>Impact</td>
<td>How much damage a risk that materializes into a problem would cause the project. (Usually presented on a 5-level scale)</td>
</tr>
<tr>
<td>MDA</td>
<td>The Science of people, computers and work</td>
</tr>
<tr>
<td>NDS</td>
<td>Non-developmental software.</td>
</tr>
<tr>
<td>Probability</td>
<td>How likely a risk is to materialize into a problem. (Usually presented on a 5-level scale)</td>
</tr>
<tr>
<td>QUB-project</td>
<td>The Quartz User-interface Builder software development project, that the two authors ran during spring 2002</td>
</tr>
<tr>
<td>Redundancy</td>
<td>Redundancy describes in this case something negative. For example that you have the same risk placed in two different categories.</td>
</tr>
<tr>
<td>Risk</td>
<td>An uncertain event possible to happen with negative outcome.</td>
</tr>
<tr>
<td>SEI</td>
<td>The Software Engineering Institute</td>
</tr>
<tr>
<td>SPI</td>
<td>Software Process Improvement</td>
</tr>
<tr>
<td>TBC</td>
<td>Taxonomy based categorisation. Was originally developed by SEI to group the findings from the TBQ, but we have proved that it even works without the use of the TBQ.</td>
</tr>
<tr>
<td>TBQ</td>
<td>Taxonomy based questionnaire. Used to identify risks by interviews.</td>
</tr>
</tbody>
</table>

### 1.5 Summary

This chapter aims to make the reading easier. To get the most out of this thesis you should read it from start to end, but if you only want to learn about one of the three main parts of this thesis you could read it a bit different, these parts are; introduction to EBRM, how to formulate risks and how to categorise risks.
In the end of this chapter we presented a dictionary with common used abbreviations and words that we felt had to be defined.
2 INTRODUCTION

This chapter includes a risk definition to define our perception of the risk concept for the reader. Thereafter comes an introduction to the latest software project run by the authors to give a picture of our experiences regarding the topic. This is followed by a chapter motivating the subject of this thesis. A previously performed survey over the topic is presented to give the reader a picture of today’s situation regarding risk management. This is followed by a chapter motivating risk management in general, explaining why risk management is so important to the software engineering discipline. Finally we present our problem formulation, hypotheses and the methods used in this master thesis.

2.1 Risk definition

Before starting to manipulate or using the term risk we first of all need to identify the risk concept. [Gluch 94] defines a risk as a noun in the following fashion:

- The possibility of suffering harm or loss; danger
- A factor, element, or course involving uncertain danger; hazard
- The danger or probability of loss to an insurer
- The amount an insurance company stands to lose
- A person or thing considered with respect to the possibility of loss to an insurer: a poor risk

[Gove 81] defines the concept of risk as:

- The possibility of loss, injury, disadvantage, or destruction
- Someone or something that creates or suggests a hazard or adverse chance
- The chance of loss or the perils to the subject matter of insurance covered by a contract
- The product of amount that may be lost and the probability of losing it.

This is just an example of the many definitions of the risk concept there are. In this thesis we are going to use the word (concept) “risk” as “the product of amount that may be lost and the probability of losing it” in connection with the statement “the possibility of loss, injury, disadvantage, or destruction” taken from [Gove 81]. To us a risk in a software development project is purely the risk of something negative to possibly happen. Though [Taha 87] describes a risk as a probability associated with an outcome or choice, regardless of the nature of the outcome. This implies that a risk could very well be associated with an outcome of positive nature. This and the definitions that describe risks as events with outcome of positive or neutral nature or both are left out by the authors of this thesis. For more information about different definitions, please see: [Hilsson 02] that presents a covering list of risk definitions. We will concentrate on risks that produce negative impact to a project if materialised into a problem.

2.2 Introduction to the QUB-project

Our thoughts about how important risk management really is to software projects begun during a large school project. The projects run as a candidate work during spring 2002. We were ten software engineering students that worked full time for 20 weeks on this project. There were also two part time MDA-students in the project mostly focusing on user interfaces and functions.

In order to clarify how our thoughts about risk management started we have to mention something about the roles we had in the project. Both authors have been
engaged in the project as project managers with responsibility for risk management. The assessment and continuous identification were made in cooperation with an assigned risk manager. During this work, the thoughts of a new model regarding risk management grew and the more we analysed the contents and the process we became more and more convinced that something had to be done to develop a less time consuming model with major benefits to the whole organisation and not having to invent the wheel over and over in every single project.

2.2.1 A project with high risk-exposure

The project was quite technological and risky due to that we had to work in a field none of us had worked in before. The product that was to be developed should be a tool for building user interfaces for Symbian OS. The code that our product should generate had to be perfectly working Quartz code that should run on handheld devices running Symbian OS. QUB stands for Quartz User-interface Builder. But we had more risks than the technological. Since not many of us knew each other before the project and even less had worked together we had social risks and communication risks that had to be solved in order to have a smooth development. We knew that our customer at UIQ-Technology (former Symbian AB) had a lot of work to do since they were close to a big release and thereby we were acting quite autonomously with less customer interaction than desired.

As Project manager and Quality manager we felt that if we could identify all of the essential risks that were connected to the project and then eliminate them the project should be successful. We therefore put together a risk-document in order to visualise which risks that had to be handled in what order and then worked out plans for how to walk around them or to decrease the impact of them. Every risk-plan was connected to someone who was responsible for that plan and a last-date for when it should be carried out.

We started our risk identification with a whole-group brainstorming session that ended up with a first draft of the risks. A couple of weeks later we had matured a lot in our roles within the project and we held another brainstorming session with both identification of risks as well as assessment there of. Then we delegated the responsibility for the risk plans to those who were fit to take care of that risk. We also had an attitude that made everyone aware of the importance of not letting a new identified risk slip through the system. All new risks were added to the risk document. We had a risk manager that worked part-time with the risks and reported to the project manager. On a weekly group meeting we discussed the risks, made plans for them and assigned them to different project members as well as made follow-ups on earlier plans. We felt that through having this risk management process we could more easily focus on the important things in the project, and the planning went easier.

2.2.2 Project outcome

We used an evolutionary development approach and delivered every part on time and we also delivered the final product one week before the deadline that was put out in the beginning of the project. The customer and the end-users were satisfied with the quality on the product and last winter UIQ-Technology decided to spend more time and money on the product by adding additional features.

2.3 Motivation for the subject

Our interest for risk management started during the QUB-project described above. When we the next semester attended courses in project management and quality management, these were focusing quite heavily on software process improvements
(SPI) which we think are very important in order for software companies to develop and mature continuously. A survey during one of these courses told us that there is a need for better risk management processes in the industry today. With this experience, knowledge and interest we mixed risk management with software process improvement (learning organisations) and we got what we chose to call Experience based Risk Management (EbRM). The essence of this model is described later.

2.3.1 Survey
The survey was constructed in co-operation with seven other students attending the course PAD005 (Project Management, 10 credits) held at Blekinge Institute of Technology. The questions in the survey were open-ended and provided the interviewee with the possibility to express themselves freely. The original purpose of the survey was to identify different problem areas within the project management domain and it was not specifically intended to emphasis risk management and experience reuse problems. The purpose of the survey is to form a foundation for our suspicions that risk management isn’t fully adequate. All students used the same interview template that consisted of 11 different areas of interest, all broken down into specific questions regarding the area. Every student performed two interviews each and thereafter the result was analysed by us and used within this master thesis. All in all there were 18 interviews conducted. The survey was conducted during the fall semester 2002 and all interviewees had the title “project manager”. All the interviews were carried out in Swedish and the result will therefore be translated into English when used in this master thesis. Due to confidentiality agreements we will not specify the identity of the interviewees or what company they work for.

2.3.1.1 Survey uncertainties
As in all research methods, there are some uncertainties and risks involved. In our case we have identified quite a few according to our experiences and we will present them here to make the reader aware of them and take them into consideration while reading this work. Each uncertainty is also connected to an explanation of how we tried to lower the impact and occurrence thereof.

- Time – as in most cases there is not infinite time. The interviewees were project managers and were all rather busy. This could lead to answers not as exhaustive as wanted. The interviewee got to choose time and place and he/she got the questions in advance. The interview was announced at least one week in advance
- Translation – All interviewers took notes during the interviews and there is a measure of uncertainty regarding how much of the answers that was captured as well as the quality of the notes. By formulating the template together we got a chance to discuss the questions and the expected outcome in advance. This gave us a common picture of what to expect and how to perform the interview
- Different interviewers – Despite all interviewers having the same questionnaire there might have been differences in interview techniques between the interviewers. We all had agreed not to steer the interviewee in any direction. The questions were not to be interpreted by the interviewer.
- Open-ended questions – This kind of questions is rather tricky to manage. The answers may be too long, too short or the answer might not reflect the intended outcome of the question. Because of limitations in interpretation freedom, there might have been misunderstandings that the interviewer couldn’t (wasn’t allowed to) straighten out with the interviewee. By asking the interviewee to tell more about a specific issue, we could modify the exhaustiveness of the answer without affect the outcome too much.
- Formulations – The formulations of the questions might in some cases have been leading or misleading. The fact that the answers were very
much alike in format shows that the perceptions of the questions were also much alike. This would contradict the presence of misunderstandings. Also the categorisation of the questions could affect the answers because of playing an influencing role regarding contextual interpretation.

- Background – The interviewees have very different background and the companies differ in size as well as in domain. The population of this survey has been chosen by letting every interviewer select two interviewees independently. This might have given us a non-representative population in terms of statistics. The intent of the survey was not to get a representative picture of the whole project management domain, but to see if our questionnaire could highlight some specific, predefined problem domains.

- Interpretations – As in most cases people try to analysis what they hear and the interviewers have probably done so even here in some degree. Some interpretation of answers is to expect when rewriting and reformulating answers to a presentable format.

- Purpose – The purpose of the survey was initially to find problems in general within the domain of project management and therefore the questions and answers might seem a bit small. The questions are large enough to identify problems within the domain as a binary function (i.e. is there a problem? yes/no)

- Measures – When expressing the extent and severity of a problem found there are difficulties in finding sufficient measures. By using binary identifiers (yes/no) we can easily measure the percentage of problems detected.

- Fear – The fact that people often are afraid of talking about mistakes and drawbacks could lead to interviewees modifying the truth. By providing an environment of trust we feel quite confident to believe that our interviewees were open-hearted and trustworthy. Also the fact that the interviewee got to choose the location for the interview might have reduced this influencing factor a bit.

The questions used from the survey regarding risk management were the following:

- “Project management = risk management? True or false?”
- “What are your experiences regarding risk management? Pros and cons?” and
- “How involved have you as project manager been in the risk management activities? Pros and cons?”

2.3.1.2 METHOD AND RESULT

When collecting data from the survey, we read all interview results entirely from beginning to end to see if our problem domain was represented unexpectedly in answers to other questions in the survey. We found that no usable data could be collected outside the answers to the questions mentioned in this chapter of our thesis. We then concentrated on the specific questions described above and collected all the answers in an EXCEL chart. In every answer where we found an explicit sentence mentioning a setback in risk management respectively experience reuse, we marked this with red. All answers containing red text was from now on regarded to be containing information about problems within the domain. The red text answers were counted and the result from here was that:

- In more than 83% the interviewee had answered that there are problems within risk management of some kind.
In 2/3 of the answers regarding reuse of experiences the interviewees had mentioned problems.

The main findings in the domain of experience reuse were that there is no time for it, the usage of the documented experience is too low, usually the process takes a lot of time and that the documentation is poor or on wrong format. Regarding risk management the main findings were that it takes time to find and manage risks, the management of identified risks is often neglected and there is no formalized method/model of how to perform these activities. This result indicates that the two domains contain problems of significant occurrence and that something can be done to improve these two areas.

2.4 Motivation for risk management in software projects

Every project is risky in the meaning of that there always is a chance that things won’t turn out as they were planned to do. The outcomes of the projects are determined by many things, some of these are unpredictable and the project manager has little control over them. Risk level is associated with the certainty level about technical, schedule, and cost outcomes. High-certainty outcomes have low risk; low-certainty outcomes have high risk. Certainty derives from knowledge and experience gained in prior projects, as well as from management’s ability to control project outcomes and respond to emerging problems. [Nicholas 01]

Everything that could cause the project to fail can be seen as a risk. If you from the beginning could identify every possible risk and either entirely remove the impact of them or avoid them your project will be successful. [Down 94] [Nicholas 01] Our experiences tell us that the hard parts with this are to foresee all major risks within a project and to come up with good plans for them. With good categorisation and checklists for risk identifications you minimise the first problem. [Down 94] [Nicholas 01] [Pfleeger 98] If you then use a good formulation-standard for the risks you can quite easy understand how to avoid them or come up with contingency-plans for those that cannot be avoided. [Nicholas 01]

According to Norris [Norris 00] there are a lot of benefits to using risk management. The ones taking advantage of risk management are said to be the members of the project, the customer, the entire organisation, senior management and project managers. The benefits described by Norris are:

- Increased understanding of the project, which in turn leads to more realistic plans, estimates and timescales.
- Increased understanding of risks and their impact. This leads to minimisation of risks and the ability to allocate the risks to those most fitted to handle them.
- An understanding of how risks can lead to more suitable contracts.
- An independent view of risks which can help to justify decisions and enable more efficient and effective management of risks.
- Knowledge of the risks in a project which allows assessment of contingencies that actually reflect the risk and which also tends to discourage the acceptance of financially unsound projects.
- A contribution to the build-up of statistical information of historical risks that will assist in better modelling of future projects.
- Facilitation of greater, but more rational, risk taking, thus increasing the benefits that can be gained from risk taking.
- Assistance in the distinction between good luck and good management and bad luck and bad management.
Also Tom Gilb appreciates the benefits from risk management and states that: "If you don’t actively attack the risks, they will actively attack you" [van Scoy 92] [Gilb 88]. This statement is one of the fundamentals in our appreciation of risk management. As a software engineer you operate in an ever changing, complex and uncertain domain with risks ever present. There is a probability that the project will fail, and by identifying and acting upon related risks, we feel that this major risk is decreased. The present situation where projects bound to fail are allowed to proceed to produce unwanted software could according to us be reduced. In [van Scoy 92] there is a description of a sample of 9 projects totalling $6.8 million run within DoD (Department of Defence) where less than 2% of the delivered software was usable as delivered. To further underline the importance of adequate risk management we would like to quote [van Scoy 92] with this sentence: “The only thing harder than predicting the future is changing the past.”

2.5 Problem formulation

In our efforts to formulate and categorise risks identified within different projects, we have found that these two areas are not as obvious as believed to be. The problems regarding formulation are that we have not used a standardised way of formulating statements and descriptions. This has lead to hard work in over-viewing and interpreting the risks. What made it work in the QUB-project was that everybody was located in the same project room and therefore the oral communication could be used effectively when interpreting risks.

When it comes to the categorisation, we have learnt that making your own categorisation system is not that easy. Because of categories that were ambiguous, too small or too broad we had problems finding risks, organising them and making them fit into an appropriate category. The ambiguity and the size of the categories made us end up with multiple instances of approximately the same risk and left out risks because of problems when categorising them.

We also lack the reuse of identified risks from project to project. The EbRM concept is introduced as a system for avoiding time consuming risk identification sessions and to give companies a possibility to mature through risk management.

Our problem is thus; we need to find a functional, easy to use system to categorise risks and thereto a format when formulating those risks which are compliant to the categorisation system. The merged systems shall be applicable to the EbRM process and it has to require a minimum of instructions to be used.

2.6 Hypotheses

Our master thesis includes the following hypotheses that shall be addressed:

- There is a categorisation system that is applicable to all risks used as sample for this thesis
- There is a risk statement formulation system that facilitates and standardises understanding, categorisation and traceability of risks
- The two merged systems shall be usable regardless what risk management methods and shall be compliant to the EbRM model as well as to each other. Ease of use is beneficial, because of project managers already lack time for this activity.
2.7 Method description
This chapter describes the different methods used for this master thesis by the authors to gain knowledge. The main activities are: literature study, a previously performed survey and an experiment. The survey indicates that there are problems within the risk management domain of today and the literature study and the experiment is about getting knowledge and trying to verify our textual findings.

2.7.1 Literature study
The literature study [Patel 91] will involve various literatures. The main focus will be on software engineering literature though. The literature will be found through manual search at the Infocenter at Blekinge Institute of Technology and through Internet search described below. Literature gathered from the standard literature list of the software engineering program will also be scanned. The things we mainly search for are: categorisation of risks, formulation of risk and risk management checklists. The search for books and articles will contain the following search words [Patel 91]; risk, risk management, software engineering, management, formulation, risk formulation, categorisation, risk categorisation and risk checklist. Also a certain amount of correlating words will be used as search words. Special databases used for the search will be; IEEE Explore, Libris, Infocenter database and IBM’s database. We have also used regular search engines on the Internet such as Evreka and Yahoo! Uncertainties regarding this method are the risk of missing an important search word or neglecting an important source of knowledge as databases or special libraries. There is also a risk that some disregarded areas contain information of worth because of them being out of our traditional thinking about where to get knowledge of a certain kind.

2.7.2 Experiments
2.7.2.1 Description
The risk documents used in this master thesis are gathered from the “Large Software Project” course held at BTH in Ronneby, Blekinge, Sweden and one risk document is gathered from this master thesis project. The projects consisted of students and lasted from January to June (20 credits). We’ve collected risk documents from totally six projects (six answers out of eight projects). Five of them were conducted during the spring semester in 2002. The other project was taking place simultaneously with the writing of this master thesis. We have additionally collected the risks identified for this master thesis project and included them to the material used for the experiments. Uncertainties regarding these documents are that they can have been manipulated after the termination of the projects, the risk management might not have been fully adequate, the lists might not be exhaustive enough to fit our purposes and they might not be fully understandable to an external reader. The risks might not be seen as real industrial data because of the staff only consisting of students. Interpretations of the risk statements will be made by the authors where considered appropriate. When risks are difficult to understand, their causes will be derived from the prevention- and contingency plans created by the projects’ risk manager.

After a lot of literature studies [Patel 91] searching for usable categories and thoughts of creating our own categories, we found a categorisation system developed by the SEI. This system is called the Taxonomy Based Risk Categorisation model. [Carr 93] and consists of a tree structure with categorisation in three levels where the end nodes represents the smallest part of risk level decomposition. By following these categories and their explanatory statements, every risk found should easily fit into one and only one category. Our experiment [Wohlin 00] is to see if this risk categorisation is applicable on all risks found gathered from the six sample projects. We also want to see if the result of the categorisation will be the same when using the CTC structure (a standardised textual risk statement format) as when it isn’t used and the risks are
categorised by a non-standardised formulation. The purpose is of course also to verify the hypotheses stated in this master thesis.

To further extend the result set of the experiment, an external group will be used to repeat our experiments. This part of the experiment is about to show us whether there are differences in our way of categorising risks and if a standardised risk statement formulation would give a higher match between two independent groups than if the risks were kept in a non-standardised way. The measures of this experiment are derived from the subjects’ subjective point of view and can therefore be said to be subjective measures. [Wohlin 00] The experiment is also to show if the Taxonomy Based Categorisation system can be used as a stand alone structure not using the TBQ.

The risks will be run through project by project and sorted into the tree structure. All eventual risk categorisation made within the projects will be neglected. The SEI provides a questionnaire to use while identifying risk, but as the risks were already identified, this questionnaire was ignored. To make the experiment efficient, we will create an HTML-based risk sorting system with the same structure as the categorisation. Each page in the system represents a level in the tree structure and consists of links (name of next level of category) and the description of the category. The last link leading to the end node opens a text editor and the risk will be written down and stored in a text document with the same name as the end node of the tree.

All risks will be saved in their original form and we keep the naming and tagging to ensure backward traceability further on. The reason for the system is mainly to ensure that the authors of this thesis didn’t cheat by sorting directly onto the end nodes without traversing the tree structure. Risks that are sorted would also receive a free text comment about difficulties that we might have encountered. When this is done, we will transfer all risks into CTC format and do the whole procedure all over again. The difference here is that all risks will be categorised by their origin, i.e. the “Condition”-clause of the CTC structure.

After this we will send our HTML-program to an independent group consisting of two M.Sc. students, also doing their master thesis on risk management, and let them repeat the two experiments. The selection of this group was strictly a convenience sample [Wohlin 00] and does not represent the whole population of software engineers, but software engineers with an interest in risk management. The target group of the results of this thesis is risk managers. For the external group not to compare the results from the two parts of the experiment, there will be one week between the experiments and all information and the categorised risks from the previous experiment is told to be erased. Because of the size of the risk material only slightly more than 10% of the risks will be given to the independent experiment group (i.e. 15 out of 141 risks).

### Uncertainties

As in other experiments there are some uncertainties to mention in order to make the reader aware of them. These uncertainties are described in the list below together with efforts to eliminate or minimise them where appropriate.

- **Level of knowledge** – The two experiment groups do not share the same knowledge. The group consisting of the authors of this master thesis have gained a lot of knowledge about TBC and CTC which the other experiment group (probably) lack. This is not to be considered a problem but an opportunity to see whether people with little or no knowledge of these two systems are able to use them with minimal information as input.
- **Result manipulation** – There is a risk that the independent experiment group do not follow the restrictions and rules set up by the authors. Because this is a non-controlled experiment, made separated from the
experiment leaders, there is no formal supervision of the methods used by the independent group.

- **Risk selection** – As we have randomly chosen approximately 10% of all risks when consulting the independent group, there might be a difference in difficulty to sort and understand the risks.
- **Communication** – The level of communication within the two groups might differ. The experiments are carried out in groups of two and involve a measure of internal communication. Disagreements in where to put risks when selecting them might be omitted in the discussion because of the level of personal knowledge and communication and therefore not being regarded as, and documented as, problems.
- **Interpretations** – In the first experiment part, the risks might be interpreted in different ways by the groups and thereby categorised in different ways. This is not to be regarded as a problem because of this is a result outside this experiment. The other results will be unaffected by this fact.
- **Background** – The knowledge of risk management as a concept might differ. We do have fairly the same education but because of people being unique, there might be some deviation between the groups.
- **Instructions** – These might have been hard to understand or they might have comprised a too small amount of information. This is done deliberately to see if the system can be used with almost no information.
- **Documentation of difficulties** – There might have been risks that have been excluded from the documentation when describing problems of categorising them. Because of the authors being very keen on documenting small difficulties regarding categorising, the independent group might have disregarded these small difficulties and therefore giving the authors a larger number of difficulties detected.
- **Size of material** – The material comprises seven projects with 141 risks in total. This might be too small to draw conclusions from.
- **System usage** – Regarding the HTML-system created for the experiment, it might have been used differently by the independent group than by the authors which might have affected the outcome of the experiments.
- **Subjective measures** – Because of this being based on subjective measures, the conception of what is to be regarded as a problem when talking of risk categorisation might differ from person to person. This might lead to differences in the result when it comes to the frequency of problems identified. The categorisation result might also differ because of gained knowledge or the perception of a risk statement.

The results of the experiments will be shown in the paragraph experiment result, 6.1.

### 2.8 Summary

To give the reader a notion of the authors’ points of view, we briefly describe the latest project that we managed together and the risk management process therein. We have noticed some drawbacks with our process that we would like to correct and we have also noticed the benefits of working closely together in terms of facilitated oral information.

From our own experience as well as from a survey performed on project managers, we conclude that there are problems within today’s risk management process. Often lack of time and lack of commitment is given as a reason for avoiding using risk management. Also the categorisation and formulation of risk statements creates difficulties when overviewing and taking actions on risks. A risk is here defined as a
current state that possibly could lead to a negative outcome in a future situation. Uncertain events with positive outcome are referred to as chance by the authors.

Our intent was from the beginning to describe a new risk management process, called the EbRM process (Experience based Risk Management), but because of running into problems with categorisation and formulation of risks, we narrowed the scope down to this and to briefly describe the EbRM model for future implementation. In order to solve the problems with categorisation and formulation we have performed literature studies, an experiment and we have also drawn conclusions from our own experiences. The issue to deal with is to find a categorisation system that is compatible with a standardised formulation system. Merging these two systems would provide a foundation of basic pieces of risk management applicable to EbRM as well as any other risk management process.
3 INTRODUCTION TO THE EXPERIENCE BASED RISK MANAGEMENT MODEL

During the survey made in autumn 2002 that is described above, we started to think of the great need for a risk management model that shouldn’t take too much time but still gave payback to the project. We also saw the needs for taking care of experiences between projects in an organisation and in addition to this even facilitate a software process improvement. These thoughts ended up in a completely new model which we chose to call Experience based Risk Management (EbRM), and how it works is to be described below.

EbRM is a model that continuously merges experiences with risk management. This model enables the organisation to learn from earlier mistakes as well as taking advantages from successful projects and its methods used to achieve that. This will in long terms increase the chances for projects that run within the organisation to succeed. [Nicholas 01] [Pfleeger 98] This means lower development costs, higher quality and higher customer satisfaction, which results in a more profitable organisation. [Down 94] The ultimate purpose is to let the EbRM process improve the software development process of the company and thereby providing for continuous improvement to the whole organisation and not only affecting single, separate projects. [Down 94] This is the highest maturity level an organisation can achieve. [Zahran 98] [Paulk 93]

3.1 Better use of experiences

It is usually the same mistakes within a company that makes project after project more expensive or less successful. There is also a problem with new project managers who has to make the mistakes before they can avoid them in the future. [Down 94] To solve this you have to learn both from your own earlier mistakes as well as from others. [Nicholas 01] Or as Otto van Bismark-Schönhausen said “Fools you are to say you learn by your experience. I prefer to profit by others’ mistakes and avoid the price of my own.” [Nicholas 01, p.534].

There have been many attempts to incorporate a knowledge management system into the organisation that should solve these problems, but often it just ends up with a project report that no one ever reads. [Down 94] As we see it you should solve the problems by having the experiences continuously coming up in the project work in a smooth way. If you make the same mistakes over and over again you probably should change something in your routines. Oprah Winfrey once said “Failure is God’s way of saying ‘Excuse me, you’re heading in the wrong direction.’” [Nicholas 01, p.534]

3.2 More effective risk management

Many project managers say, according to our study described earlier, that the risk management consumes too much time. The problem as we see it is that they don’t feel that the risk-work pays off. Our intentions with the EbRM model is not that it directly will decrease the time spent on risk management within a project, but it would give much more back to the project than traditional risk management models. For example would you have a more comprehensive risk management and you would also get a foundation for a continuously software process improvement. The current project is not the only one to gain benefits from this, but it is gained by every project that runs and will be running in that organisation.
3.3 How EbRM works

In order to be able to shortly explain how EbRM are thought to be constructed we start by giving the reader an overview picture of the model. The following paragraphs refer to picture 3.3 where the main characteristics are addressed.

**Picture 3.3 Experience based Risk Management, general model.**

- **Predefined form (General risk domains)**
- **Updateable form (Risks, solutions, avoidance, assessment)**
- **Project (Ongoing risk management)**
- **Final project report (Good/bad)**
- **Frequently used actions are transferred into the software development process.**
- **Good experiences are used for creating risk-avoidance plans or contingency plans for some appropriate risks.**
- **Bad experience becomes new risks or adjusting the values for impact and probability for some appropriate risks.**

Good experiences are used for creating risk-avoidance plans or contingency plans for some appropriate risks.
3.3.1 Predefined form
These are the general risk domains with its general risks that are delivered with the EbRM-model. This is there mostly for having something to build on and as an example for how to continue the risk work.

3.3.2 Updateable form
The brain in this model is the collection of all the risks identified in the organisation together with plans and values as well as indications for what kind of projects they apply to. These risks could be stored in any form, but we recommend the use of a computerised database system because of its advantages over manual storage. [Connolly 99] It is from this list the risks for every project are chosen, this is primarily done in the beginning of each project but the database will be continuously consulted during the whole project. When you select a risk for your project you get a subscription for that risk and if another project also having that risk comes up with a great plan for avoiding the risk you will be automatically notified.

3.3.3 Project
Every project makes a collection of the risks from the updateable form that suits that project. Of course every project has its own new risks, and these are of course added to the list of risks for that project. [Down 94] When taking a risk from the updateable form you will also get some plans for how to avoid the risk or decrease the impact if that risk materialises. With every risk there will also follow values for impact and probability, these values comes from statistics from earlier projects and will be continuously updated. This can be seen as the heart in the EbRM-model.

3.3.4 Final project report
After each project you write a final report that are used to adjust the values for risks in the general database, as well as adding action plans for some risks. Perhaps there even will be some new risks added to the database.
- If the project went bad – why not capturing the faults in order to not make the same mistakes in the future? If the final conclusion could quash just one high risk then it will be an investment well rewarded. [Down 94]
- If the project went very good – why not capture the good things? There is no law against doing things right a second time. [Down 94]

3.3.5 Software process improvements
The risk management will be continuously improved, and so will also the software process model. For example if your organisation has come up with avoidance plans for a risk so that the probability becomes nearly completely reduced, these plans should be part of or change the software process. Perhaps you then can delete that risk, since it is no longer a risk as long as you follow the software process. [Down 94]

3.4 Summary
The Experience based Risk Management model merges risk management and experience management in a way that continuously improves the organisations ability to succeed with software projects. This chapter has gone through only the foundation of how to build the EbRM model and only the most important parts are mentioned here.
4 **INTRODUCTION TO CTC**

Before giving solutions to the problems regarding formulation of risks in this thesis, we have to introduce the concept of CTC. The CTC structure will be the basis of the solutions to the formulation problems stated in this master thesis.

4.1 **The CTC concept**

The CTC risk description was developed by the Software Engineering Institute (SEI) and was first described in July 1994 by David Gluch. [Gluch 94] The CTC abbreviation stands for Condition-Transition-Consequence. The report describing the CTC concept aimed to present a formalism for describing risks associated with software development so that risks could be managed as an “…internal part of routine program management activities.” [Gluch 94] The CTC format was constructed to form a standardised risk statement structure. The criteria for a risk to appear are according to Gluch [Gluch 94] a known environment/state (Condition) that provides a foundation for future probable problems (Consequence). There should also be possible negative outcomes from the current condition if some (often not known) event should alter the condition so that the risk materialises and becomes a problem. The uncertain event (the Transition) could often be time itself because certain conditions might be adequate at one point in time but not in another. A risk doesn’t exist if there is no activity (time) and it doesn’t appear in a static environment.

The Transition part of the CTC structure implies that some (maybe unknown) change could alter the environment in such a way that the current conditions are no longer adequate. “The interrelationship of uncertainty and time is evident in the uncertainty associated with risk, in that this uncertainty reflects the uncertainty regarding future events.” [Gluch 94] If the conditions on the other hand are sure to lead to negative consequences then there is no longer a risk. The risk is said to have materialised into a problem and the negative outcome is a fact. Gluch identifies two attributes that are the fundamental characteristics of risk; time and value. As the outcome is measurable, at least qualitatively, a risk is said to be measurable by default. Time is as described above a necessary ingredient regarding the non-static environment that provides for the ability for things to change in an uncertain way. The uncertainty is often included in the risk representation while defining risks [Gluch 94]

The main purpose of the CTC structure is to transform a perception of risk (fundamentally subjective) into a distinct risk entity that can be described and measured. The CTC structure is originally based on a state-space representation of a project as a point in space with ability to move in a three dimensional way depending on the different values of measures for the current situation. For more information about this representation, please see [Gluch 94]. This model was used intentionally to present risks in an accepted and scientific way and our intent is not to further describe this representation but merely give the reader a background of how the CTC structured was formed. To facilitate the representation of a project as a fuzzy point in a three dimensional structure and on the same time show how risks correlate to the concept of time we would like to present to the reader a replica of Gulch’s Value-Time representation of risks, their values and their transition over time, picture 4.1.a. The figure shows the dependency between value and time and in this case, the current state leading to a future less desired state over time. The concept of state and transition is summarised by Gluch as follows:

- “The system is a fuzzy point in the space, and
A risk is a potential transition, a trajectory through the space, from the current state of the system (project, authors’ remark) to some other (undesired) state.”

As stated before, the CTC representation of risk involve Condition (description of the current condition prompting concern), Transition (the part that involve change (time)) and Consequence (description of the potential (negative) outcome). This structure can be represented as a visualised triplet, picture 4.1.b, built by connected blocks.

<table>
<thead>
<tr>
<th>REQUIRED</th>
<th>Transition</th>
<th>Consequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Picture 4.1.b, The CTC structure as a visualised triplet

The light-grey fields indicate future events included in the risk concept. The “Required” mark in the “Condition”-field indicates that this is the only really necessary item when transforming the CTC visual structure into actual risks, textually described. The minimum according to Gluch when describing a risk is the statement of conditions connected with an expression of concern about the potential consequences. The generic structure for formulating risks textually using the CTC representation is according to Gluch:

**Given (that) <condition> then there is concern that (possibly) <consequence>**.

This is the textual representation that we are going to use when finding a proper way to formulate a risk so that the formulation problems stated in this thesis are being avoided in the future. As an example of using this statement foundation we would like to use the current level of testing in an imaginary project. The testing concerns the project manager because he thinks that this might lead to too low quality of the final product. He therefore constructs the following statement:

**Given <the current level of testing> there is concern that (possibly) <the quality of the final product might be inadequate>**.
This is not merely an example of how to use the statement to form textual risk descriptions, but also how you can describe a risk without adding a certain value to the current conditions. The implicit statement here is that the current level might be insufficient but not necessarily so. This lets us focus on putting values to the probable outcome if the risk materialises.

4.2 Summary

This chapter has given an introduction to the Condition Transition Consequence formulation format that is used throughout this thesis. The CTC format is split in three parts:

- Condition is the current environment or situation that implies concern.
- Transition is the uncertain event (often not known) that could alter the condition so that the risk materialises and becomes a problem.
- Consequence is a measure of what negative aspects that might have impact on the project.
5 INTRODUCTION TO TBC

In order to solve the problems found with risk categorisation we have performed some experiments with existing risk categorisation set, as well as tried to come up with a new one. This chapter will introduce the reader to the only thorough categorisation system that we could find, which is the SEI Taxonomy Based Categorisation (TBC) that comes from the Taxonomy Based Questionnaire [Higuera 96].

5.1 The SEI TB categorisation

Risk taxonomies are lists of problems that have occurred on other projects and can be used as checklists to help ensure all potential risks have been considered. An example of a risk taxonomy can be found in the Software Engineering Institute’s Taxonomy-Based Risk Identification report that covers 13 major risk areas with about 200 questions.

The problems with other categorisation sets that are described in some literature are that they are just examples and therefore not complete as described by for example Down and Nicholas. There is also some literature that just tells you to find the risks but that give you no hint about how to formulate or categorise them [Pfleeger 98] The description for how to come up with an own categorisation method is presented later in this thesis.

The existing risk categorisation set used was taken from Higuera and Haimes [Higuera 96] who wrote this for the Software Engineering Institute, SEI. This is a quite thorough categorisation set divided into three main categories and with a depth of two subcategorise below these. You can find the complete categorisation tree later in this chapter. In order to find out whether it was a good enough categorisation system or not we made some experiments to test it. The results from these experiments are presented later in this thesis; we will here just introduce you to how it works.

5.1.1 The structure

The categorisation set presented by SEI are divided into three levels with different amount of categories on each level. The following picture, picture 5.1.1, shows an example of this:

<table>
<thead>
<tr>
<th>Class</th>
<th>Product Engineering</th>
<th>Development Environment</th>
<th>Program Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>Requirements</td>
<td>Engineering Specialties</td>
<td>Development Process</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Work Environment</td>
</tr>
<tr>
<td></td>
<td>Stability</td>
<td>Scale</td>
<td>Resources</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Formality</td>
<td>Externals</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stability</td>
<td>Scale</td>
<td>Formality</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Product Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Schedule</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Facilities</td>
</tr>
</tbody>
</table>

Picture 5.1.1, Overview of the SEI Taxonomy Based Categorisation System
Any given risk should only fit into one of the three “class”-categories and then just fit perfectly into one of the “element”-categories that lies under that “class”-category. The risk should then fit into one, and only one of the “attribute”-categories under that “element”-category. In whole there are 80 categories, distributed as:

- Class-level: 3 categories
- Element-level: 13 categories
- Attribute-level: 64 categories

5.1.2 The categories

The categorisation method is originally used together with the Taxonomy Based Questionnaire where the questions have as goal to identify the risks that are connected to that project. The risks identified with the TBQ will automatically be categorised in the categorisation set mentioned above. “Both the questionnaire and the application process have been developed using extensive expertise and multiple field tests.” [Higuera 96] Although the TBC is constructed to fit the TBQ you can use the categorisation system without using the questionnaire.

The complete list of the categories is found in the following picture, picture 5.1.2 and the additional explanatory statements thereof can be fund in appendix A.
The quite large amount of categories is a necessity in order to have them on an evenly
general level. If you have too few categories they have to be quite general and the risks
for a risk to fit into more than one category increases. But on the other hand, if you
have too many categories they will be too hard to overview.

The definition of each category will be further described in appendix A.

5.2 Summary
This chapter has described the Taxonomy Based Categorization system, which is the
most thorough published system that we could find. It is split up into three different
head-categories and has a depth of two subcategory-levels. Totally it consists of no
less than 64 categories on the lowest level.
6 EXPERIMENT

The experiments were conducted as described in the method description, and the results thereof are described below.

6.1 Results

We, the authors, first made an experiment where we categorised the 141 risks, which we had collected from different software projects, in to the Taxonomy Based Categorisation system. Every risk (100 %) could be categorised with only minor problems. These problems originated from badly formulated risks that we had to interpret. This interpretation could be seen as a weakness for the experiment since we perhaps thought of the risk in another way than the original author of that risk did.

After that we re-formulated all the risks into the CTC-format described above and once again categorised the risks without taking care of where we had placed them before. This categorisation was much simpler since we now more clearly saw what the condition was and what the consequence was. Again we managed to categorise 100 % of the risks into the TBC system, but we only had a concordance in 55 % or 78 risks with the non CTC-formulated risks. This doesn’t necessarily prove that the CTC-formulation technique is better than without any formal formulation standard when it comes to categorisation. But when we let another, independent group, make the same experiment more interesting results appeared.

We randomly chose 15 risks, slightly more than 10 %, that the other experiment group should categorise. The first round they got them on their initial format and they succeeded with categorising 100 % of them. After a week they were sent the same risks but re-formulated to CTC-standard. This time they also managed to categorise them all and the concordance between their first categorisation and their second was at 53 %, almost the same as we had.

The interesting was that a comparison between our first experiment part with their first part gave a concordance in only 46 %. And the second round with the CTC-formulated risks gave a concordance in nearly 87 %. This gives the following findings:

- The Taxonomy Based Categorisation system developed by SEI managed to categorise all of the 141 risks that we had collected from different projects.
- The TBC-system is quite intuitive, both the groups could easily categorise the risks without spending time on learning a new system. A minimum of instructions were needed to use the system.
- The TBC can be used without the TBQ even though they were created to be used together.
- The use of the Condition Transition Consequence formulation format decreases ambiguity of risk interpretations. 87 % concordance against 46 % without CTC-formulated risks.

What we can not say is that the CTC format is better than any other risk statement format but our focus is to find a format that actually could be categorised 100 % with the TBC system and therefore we can not see this as a negative finding, but instead as an opportunity to evaluate other formats relative the CTC format. Because of the CTC format providing the opportunity to categorise all sample risks, we feel confident enough to use this in our further risk management tasks.
6.2 Summary
Due to our experiment we have a result that says that both the CTC and the TBC work as it was meant to do. 100% of the sample risks could be categorised within the Taxonomy Based Categorisation system and we have approximately 47% better concordance between the different groups with the use of the Condition Transition Consequence formulation format. This means that we feel confident in using these two techniques with the EbRM model.
7 HOW TO FORMULATE RISKS

This chapter describes the problems that we have encountered during our risk management experiences. Every problem chapter is followed by a solutions part where we provide the reader with solutions and hints for easier management regarding risk statement formulation. The solutions are not only useful when using the EbRM model, but should rather be seen as a universal way of handling risk formulation regardless what model you use. The problem domains below were selected by brainstorming performed by the authors, amongst the experiences we have from software engineering projects.

7.1 Understandability – the essence of formulation

When searching for descriptions of problems related to risk statement formulation, we soon came to the conclusion that there is not very much literature to gather information from. In order to find useful facts about statements and understandability we were forced to consult the requirements engineering literature. The problems described reflect the formulation problems regarding requirements, and this is described by Kotonya and Sommerville. [Kotonya 98]

Regarding understandability we agree with Kotonya and Sommerville when they say that understandability is probably the most important attribute for requirements. We extend the reasoning to include risks as well. Without understanding the risk in the first place you won’t be able to use it in a proper way. In our project we had identified a risk that stated: “For example if a project member wants to take a vacation.” What does this risk really mean? Is there a possibility to validate the risk? Is it possible to estimate the impact on the basis of this statement? The answer is no according to us. We cannot say anything about the probability or impact. Is there really a risk that someone wants to “take a vacation” at all? The project manager can simply reject the question and then the risk is gone. The risk is more likely to be interpreted to mean that there is a risk that a project member asks for a vacation and this formulation does not imply that the actual vacation (i.e. the absence of the person) is a risk in this project. This is what happens when you mean something and write another. This problem is seen by us as a composite problem and it will be used as exemplification in some of the following paragraphs. This implies that to have complete understandability, all problems below must be addressed and solved.

As a checkpoint for understandability Kotonya and Sommerville suggest that you should ask yourself the question: “Can readers of the document understand what the requirements mean?”[Kotonya 98] They mean if that a statement cannot be understood it cannot be validated or verified either. We could not agree more. Though they talk about requirements and not risks, we are confident to translate this statement to the field of risk management. One solution that occurs to us is to perform a review to ensure the validity of the found risks. Kotonya and Sommerville describe the review as the most widely used technique of requirements validation. If questions arise in the review team, then the formulation should be rewritten and agreed upon by the author and the review team. This could very well have been a solution to the problem if the problem was formulated to include only problems with formulations in risk management. This is not the case in our situation. We have a survey conducted on 18 project managers that generally points on the simultaneous problem with risk management taking too much time. “Since inspection is labour intensive, the return from this process in not immediate and clear to the management. Therefore there is a perception that inspections cost more than they are worth.”[Aurum 02] This is something that we would like to avoid; to incorporate even more workload to the
software management. Even though the inspection should prove to be cost efficient and swift in its performance, there tend to be reluctance to adding more work to a business that already struggles with firm deadlines, a tight budget and insufficient resources.

Our experience tell us that the person to best understand what is written and meant by it is actually the person who wrote it. From this we can conclude that the risks should be formulated by those who shall handle them in the future or they must be very well formulated. Distribution of workload requires higher degree of accurate communication in terms of notifying upper management about changes in the risk situation. It also means that, if distributed over subproject leaders, the project manager could lose the overall risk notion and thereby feels uncommitted to the risk management process. The main focus of this part of our thesis isn’t though on how to distribute risks, but how to formulate them so they become understandable to others in, and possibly outside, the project and that it is possible to communicate and measure them. The problems stated above will form the foundation of the coming paragraphs where we will run through those problems and give possible solutions thereto.

7.2 Completeness

7.2.1 Completeness: problems
This problem regards as well completeness regarding the formulation of the identified risks as completeness of the set of risks found. A risk that is incompletely formulated could lead to misinterpretations and confusion when it comes to planning the actions to take. When it comes to Kotonya and Sommerville’s description of this problem area they suggest that ALL requirements shall have been found. [Kotonya 98] This is not the proper way of handling risks according to us. The number of risks with some, however low, impact on the project soon grows to infinity and this makes the risk management an impossible task.

Referring to our survey which partly led to this thesis, project managers are not interested in risk management if the effort doesn’t correlate to the gain thereof. It is also an implicit demand from the project managers in the survey that the risk management process is swift and easy to use. Kotonya and Somerville also talk about incomplete information. To exemplify the problem we can use a quotation (used before in this master thesis to show that many problems can relate to one and same risk) from the risk management document in our project: “For example if a project member wants to take a vacation.” This is an incomplete risk because it doesn’t describe for how long the vacation would be or if it is a special person during a special phase of the project. This also makes it impossible to calculate the impact if the risk should materialise into a problem.

7.2.2 Completeness: solutions
Regarding completeness in terms of finding all or as many risks as possible, this is really out of the scope of risk formulation because of identifying risks taking place slightly before formulation issue enters the process. Nevertheless there are numerous ways of finding (identifying) risk. The one we used in our project was the brainstorming technique [Gallagher 97] and seems to be the most used technique within the software domain for identifying risks. There are also periodic risk reporting, project profile questions, taxonomy-based questionnaire (TBQ), TBQ interviews, voluntary risk reporting and so on.[Dorofee 96], [Gallagher 97] Whatever method chosen and used the outcome is a set of identified risks. This is of course not all risks that might have effect on the project because of the impossibility to make fully exhaustive risk identifications due to its infinity. The issue here is to formulate the risks so that it is understandable and verifiable. In the word understandable we include
all the issues in the Risk formulation solutions paragraph and the verification is handled in a separate paragraph and we see these two criteria to be main categories of all problems that we have identified regarding risk formulation. How do we then verify that the risk statement and its context are complete?

Our suggestion, based on our own experiences, is that the verification is done with two subjective and binary questions. The first question to be answered is: “Do I have enough information to understand and act upon this risk?” This question is gathered from [Kotonya 98] and modified to fit risk management. If the answer is no, the solution to the problem is to reformulate the sentences in a way that clarifies them without altering the contents thereof. If the person not understanding the formulation isn’t the author, the author should be consulted to explain and rewrite the risk. The other question to ask is: “Can impact and probability be measured based upon this formulation?” According to [Gluch 94] a proper risk statement is measurable by default. [Boehm 93] emphasises the necessity to be able to predict the impact and probability of each risk. If the answer to the previous stated question is no, the suggestion is to use the distinct impact and probability scale described in the impact and probability paragraph in this master thesis. This lets you get away without having to split the risk into a non-manageable amount of risks and on the same time describing a sequenced, measurable risk with both impact and probability assignable to it. It also clarifies the different values (states) that the negative consequence could mean to the project. This system could of course be used even if the risk statement formulation is clear and adequate to increase the understandability of the consequences.

Using the CTC representation of risk entities, you are ensured to capture the conditions leading to a possible future problem. [Gluch 94] The CTC format also suggests that you as well include a context clause further describing the main important issues and circumstances that surround the conditions of the risk. [Gluch 94] This clause could also give additional information not fitting into the risk statement. The importance of fulfilling certain commitments and tasks could also be described in this clause. This is described more thoroughly in the “Exemplification” paragraph in this master thesis.

The CTC format can help you in order to get a more complete risk formulation than with an ad hoc formulation, but there are no guarantees that the risks will be waterproof in terms of completeness. This can be seen as formal criticism towards the CTC structure, but because we have failed in our quest to find another way of formulating risks that provides with total completeness assurance and the fact that the CTC format is standardised and quite exhaustive in risk description, we choose to recommend this anyway.

7.3 Ambiguity

7.3.1 Ambiguity: problems

This problem is described by Kotonya and Sommerville as terms being used that were not clearly defined. [Kotonya 98] In the QUB project there was a risk in the customer category identified as; “the relation to the customer turns out to be unsatisfying.” Here the reader might wonder what the word “unsatisfying” means. Is it too little communication, too little time to meet with him/her, wrong kind of information or just the presence of dislike between the customer and a project member? Statements like this provides for subjective interpretations of words. Despite this doubtful formulation, the risk stayed in the document the whole project through and was evaluated, calculated upon and monitored as any other risk. Even plans of avoidance and prevention were set up. Did the calculations of this risk really have any value to the
project? How can you measure something you are not sure on what it means? In the risk documents used in our experiment, we have found quite a few technical abbreviations that were not understandable to us as external readers and this is according to us a sign that not everybody even within the project would be able to read the document.

7.3.2 Ambiguity: solutions
As described in the ambiguity problem paragraph above, this is really about making sure that everybody understands the formulation and the words therein. As for the example with the risk that: “the relation to the customer turns out to be unsatisfying”, the problem in terms of ambiguity is the word “unsatisfying”. This implies a couple of interpretation possibilities and it therefore also makes the risk composite. This problem could have been solved by using the CTC structure. The risk as described above is merely the “consequence”-part of the whole CTC structure. [Gluch 94] What is forgotten is to focus on the condition that might lead to this unwanted state. The CTC structure would for example, in the QUB-project, imply that the risk should have been formulated: “Given that <the customer is very busy running several other projects> then there is a concern that (possibly) <there is too little time to meet with the customer to discuss issues>”. This formulation not only decomposes the condition given at hand, it also breaks down the consequences of the risk. This makes it easier to deal with according to us and it reduces the level of complexity and thereby the ambiguity.

The CTC structure also states that the condition should be further described in a context clause giving a motivation and an explanation to the current situation possibly leading to the negative consequence. If you feel that the risk (compositely described) isn’t a big issue and the total impact and probability would be low, we suggest that you keep it as it was from the beginning and start to decompose it when it feels like the risk would really start to affect the project. Of course, if you identify some of the composite parts as important, or uncertain, and that the risk is very depending on this, a decomposition of the statement is required by us in order to get a manageable set of risks.

In order to straighten out risk statements regarding confusion of words, Gause and Weinberg suggests that you should avoid comparative words like “small” or “expensive” and instead give certain values to these parameters instead. [Gause 89] In our case with the word “unsatisfying”, Gause and Weinberg suggests that you should make definitions which states that are included and which is not. This means that we would end up with a list specifying what really is in the word “unsatisfying” e.g. “too little”, “non-constructive” etc. We should also specify what should not be regarded as unsatisfying. For example the personal relation between a project member and the customer should maybe not be included in this risk statement, because of its small impact. This should according to us be decided by the person handling the risks. A drawback with making this sort of conceptual explanation list is that the risk management becomes more time consuming and that is an unwanted state. It also could lead to that the risk awareness becomes narrower because of the exemplification.

An ambiguous formulation could actually mean that more (previously disregarded) aspects of the risk statement are handled. If you have a list with what is in a risk and what is not, it is our opinion and experience that people tend to stick to it rather strictly and therefore misses interpretations of importance, just relying on the written word.

What is of importance to us when handling this ambiguity problem is that the risk should be measurable in terms of impact and probability. We suggest that you read through the statement a couple of times to see if there are some obvious ambiguity
problems. Then we suggest that you check that the risk is really measurable in terms of impact and probability.

One thing to remember is that “One word is worth a thousand pictures.” [Gause 89] This means that interpretations are made differently by everyone reading a sentence and this interpretation gives the group of readers different pictures trying to visualise the concept. By keeping this in mind when communicating the risks, we feel that an oral two-way communication is one way of verifying a shared picture over the statement. Because of the literature not giving too much support regarding this issue we will have to base this recommendation on own experiences. With questions and explanations a discussion would probably reduce the interpretational differences according to our knowledge.

7.4 Standards

7.4.1 Standards: problems

If the risk management document does not conform to standards there is a risk, according to us, that it is being treated as something remote from the project plan. This could lead to risk management not getting the attention and time that is needed for this activity. If not following the standards regarding language, layout and identification, it is likely to be seen as an obscure, time consuming by-product of real work (software development). Also a problem we’ve encountered was that the formulation of risks wasn’t standardised and therefore they were hard to read and interpret. The focus in this master thesis is on formulation and not on how to create templates for entire risk management documents and therefore the solution to this problem will emphasise on the formulation of risks.

7.4.2 Standards: solutions

Gallagher [Gallagher 97] suggests that you should use a risk document, called a risk form, in order to get all essentialities of the risk specified. The risk form includes the very basics of a risk description. Their risk form includes fields for impact, probability, timeframe, identification number, date, risk statement, context, immediate action indicator, action recommendation and classification. [Gallagher 97] This would help in terms of standardised appearance and would according to us facilitate the reading and thereby also the decision making. It also decreases the likelihood of information missed out and increases the understandability. [Kotonya 98] This form should according to us be internally created by the company to fit in to the rest of the textual work products. By using externally created templates there will probably be an unwanted divergence between the risk document and the rest of the project documentation. By letting risk management produce internal work products that are incorporated in the software development process, the status of risk management activities will increase and future on it will gain the status of a naturally involved process.

Also how to write the risk statement and the context should according to us be standardised in a way that verification always is allowed. As well as other software documents the risk management document should follow the same standard as other project documents. This to signal that this piece of paper (or whatever storage media) is just as important as the work products of the project such as project plans, organisations charts and so on. This is all about giving legitimate status to the risk management activities and thereby encourage project managers to use the document as well as its surrounding process. Developing and following of standards also provides the facility of checking that all that ought to be done really is. The CTC structure [Gluch 94] allows for a standardised format of the risk statement and its related values. The format consists of the statement: “Given that <condition> then there is concern
that (possible) <consequence>”. This provides according to us an easier reading and understanding. It also expresses what problems might occur and what condition that would be the source of the probable future problem. As shown in the Experiments result part, the correlation between different people’s perception increases and this will facilitate the categorisation process.

One of the problems encountered by us is that the format of risk statements varies within the risk management document. Sometimes the condition is described and sometimes the consequence is described. In some risk document we’ve even found risks formulated as questions. This confusion of formats makes the handling of risks exceptionally difficult. It becomes very difficult to categorise the risks, leading to difficulties overviewing the risk document. It makes it difficult to create action plans because of the fact that it becomes hard to see what the cause of the problem is. It makes it difficult to see where in the chain of depending risks this particular risk should be located. By using the CTC format, the condition and consequence is clear to the reader and thereby the problems stated above could be easily avoided.

7.5  **Left out risks**

7.5.1  **Left out risks: problems**

As said before in this part we didn’t manage to identify all risks in the beginning of our project, and we don’t think it is possible either. We are sure to believe that we hadn’t identified all risks in the project when coming to the closing of the project either. There might (probably) have been a lot of risks with impact on the project that just never materialised into problems which we never identified. Something to think about when managing risks is that it is not likely that all risks you will have to deal with, running a software project, are identified and documented. If you don’t consider this you will probably end up hitting a problem while handling the identified risks as if they were the only possible ones. When it comes to the connection between left out risks and formulation, our experience tells us about risks missed out because of improper formulations. This could be a problem of interpretation or ambiguity but anyway; a risk that is not formulated in a strict and proper way could mean a composite risk including several dependent risks to the writer whilst the person who is about to handle the risk, planning for contingency or assess the risk might only see one of those angles. This could actually lead to missed out risks because of formulation errors.

7.5.2  **Left out risks: solutions**

As we suggested in the problem description, we are not after identifying and managing all project related risks. We though strive towards a formulation technique that ensures the finding of all risks included in a certain formulation. A composite risk provides for the possibility to get the risk manager to think that all risk included in the composite risk are handled while perhaps just a few of the conditions have been found, analysed and taken care of. In order to get rid of this problem we suggest that you use the CTC structure [Gluch 94] and its breakdown structure when formulating the risk statements. In combination with the Taxonomy Based Categorisation System [Carr 93], you get a clear picture of all identified risks with all of their conditions and impacts. This standardised format decreases the possibility to leave out risks by mistake. [Kotonya 98] You also get a formal structure of risks which leads to the ability to overview the risks and a possibility to see where the risk belongs and thereby separate risks that look alike but belong to different taxonomies. This, according to us leads to decreased possibility to confuse risks, making you believe that you have already addressed the particular risk. By traversing the Taxonomy Based Categorisation tree in a specific order you are sure to find and address all risks identified.
7.6 Generality

7.6.1 Generality: problems
To be able to handle a risk there must be some level of generality to it. Formulating a risk which states “there is a risk that the project will fail” doesn’t provide you with a good foundation for creating plans how to tackle it. This is according to us an example of a composite risk. We will get back to the reasoning about cause and effect and dependencies between risks further on. This problem includes as well categorisation as formulation of risks. The risk mentioned earlier in this paragraph is far too large to be manageable and it might also be seen as the ultimate risk in every project and it might also be seen as a consequence (effect) of other risks that materialised in such a way that the project failed. In our project we found out that the level of generality was very varying. Some of the risks were very loosely specified (mostly relational issues as communication) and some of the risks were very precise (mostly technical development issues). The degree of generality gave us problems because it was almost impossible to calculate the aggregate impact and possibility when the risk was too undefined. The risks that were highly specified were rather easy to handle in terms of generating plans and actions but specifying a risk into atomicity doesn’t always help in terms of predictions and calculations. The question here is where to stop decomposing risks.

7.6.2 Generality: solutions
The CTC structure forces us to create risk descriptions as thoroughly thought through statements decomposed into sufficient level of generality. Our suggestion of knowing when to stop decomposing risks is when you have a condition that has its direct possible consequence on one of the four ultimate project constraints: time, quality, function and money. [Down 94], [Norris 00] This could also be expressed as three project constraints concerning technical risks as well; cost, time and customer acceptability.[van Scoy 92] These parameters are the ultimate project constraint because every negative event finally can be predecessors to negative impact on one or several of these. For a project to fail, one or more of these constraints must be that negatively affected that they are irreparable. Time refers to the timeframe given for the project within it must be completed to be a success. Quality refers to the total quality of the delivered product. If quality isn’t acceptable, the customer won’t accept the product and thereby the project would be a failure. Function describes the completeness of all features of the final product that are demanded by the customer and money refers to the economic constraints in form of project budget. Function could also be seen as part of quality in a broader sense.

When creating a risk chain we suggest that you stop at these boundaries mentioned above because they can be seen as the time and space limits of the project. When overrun or not fulfilled the project will be a failure on delivery day. If all four constraints are intact at the day of delivery, the time scope of risks transforming into problems is passed and because the termination of the successful project, impact of all risks will be 0 (zero). As for an example we could take a risk and transform it into CTC risks linked to each other to an ultimate constraint shown in the figure below, picture 7.6.2.
The chain of connected risks are further described and discussed in the cause and effect session below.

7.7 Cause and effect

7.7.1 Cause and effect: problems

When constructing the risk management document for this master thesis project, we tried to do this according to the basics of risk management as possible. This included: identification, categorisation, analysis of probability and impact. All according to the basic foundation of risk management found in Shari Lawrence Pfleeger’s book [Pfleeger 98]. We soon found out that according to the formulation of the risk we could manipulate the values of impact and probability quite a lot. One finding here was that some of the risks that we found were really not risks themselves, but rather effects of other risks. As one can see, the risk “the product will not be delivered on time” is really the result of a vast collection of other risks such as: “a person getting ill”, “not approved on final delivery” etc. etc. The risk “not approved on final delivery” can in turn be seen as reformulation of the ultimate risk that “the product will not be delivered on time” itself or it might be one of many sub risks leading to non-approval. Also the risk “not approved on final delivery” consists of a lot of risks, such as: “poor quality”, “not appropriate size” and so on. To further emphasise this problem, you can choose to break down the risk of “poor quality” into even more risks, like: “not following requirements”, “bad language” and so forth. This problem is something you have to deal with before telling people to use the risk management process.

Some risks in our project were dependent on other risks and this relationship was never handled. As for the risk “activity estimation failure” is strongly connected to the risk that “If time starts to run out it could be difficult to do a good work.” The latter could be said to be dependent on the previous one in the way that if estimations were wrong it could lead to time pressure in the end. This was certainly a “common sense knowledge” in our project but the dependencies were not described anywhere and therefore you might have omitted to notice the connection between those risks. It gets even worse when you have a lot of technical risks. These dependencies show up especially obvious when talking about interface connections and integration problems. If you in some way alter an interface to reduce/avoid a risk on one side and the other
The group decides to solve the risk in another way you have big problems correcting the triggered action plan.

### 7.7.2 Cause and effect: solutions

To solve this problem by formulating the risk statement in a proper way, we once again go to the CTC risk representation. The suggestion made by [Gluch 94] is to take a snapshot of the current situation in the project and use this as input for risk analysis and identification. The current situation forms the foundation for concerns about future uncertain events with probable negative impact. When using this view identifying the risks, you automatically identify the real issue and thereby the situation to alter to prevent the risk from materialising into a problem.

The problem that we encountered when using the traditional risk representation without a standardised format was that some risks were formulated as conditions and some of the risks were formulated as consequences. This fact led to confusion about where to attack the risk when preventing it from occurring. A risk can have multiple co-occurring conditions leading to the same consequences and this was in our project mixed together into one and potential cascading events were not separated. There is also a possibility that one single condition might lead to several consequences. The suggestion here is to split all conditions leading to a specific consequence into separate risk entities according to the figure below, picture 7.7.2.a [Gluch 94] The definition of the current conditions might also be the foundation for action plans, preventive actions and nevertheless the foundation for software process improvement once the non-functional conditions are revealed. [Zahran 98]

![Picture 7.7.2.a, Decomposition of a risk with several conditions]

This figure describes the decomposition of conditions that lead to the same consequence in the same risk entity. The decomposition of risks leads to an easier creation of preventive plans because of the atomic conditions could be handled one at a time and thereby facilitating the creation of specific prevention plans. The alternative would be to let the general (composite) condition be handled without decomposition and thereby, according to us, enabling the possibility to leave some conditions without plans. The negative aspect of decomposing the risk conditions into atomic pieces is that there will probably be a large amount and this will be hard and time consuming to handle, but to be sure that every condition leading to concern about possible negative future consequences are being taken care of, we find no other way than to perform this procedure. The same decomposition should be used the other way around as well. A
condition leading to multiple consequences should also be broken down into separate risk entities according to [Gluch 94]. To be able to separate a risk from a problem Gluch suggests the following definitions:

- “A problem involves a value judgement made upon the merits of the current condition. It is a condition that exists and is undesirable.
- A risk involves a value judgement made upon the potential implications of the current condition. It suggests a possible, future undesirable condition (consequence).”

This implies that the conditions described in the CTC structure could be recognised as problems in that they might lead to a negative consequence depending on uncertain events. This also implies that there might be an overlap between risk entities whereas the consequence of one risk might become the future condition of another risk entity. Present problems also have the possibility to be risks themselves in that they might lead to other problems in the future. The overlap is depicted in the figure below, picture 7.7.2.b. [Gluch 94]

![Problems & Risks Overlap](image)

**Picture 7.7.2.b, Intersection of risks and problems**

The solution to the problem of identifying what is cause (condition) and what is effect (consequence) is to use the CTC construction when describing risks. Using this format in connection with thorough risk decomposition allows for controlling what is condition and what is effect of the risk. As we’ve said before, the condition clause in the CTC structure is a snapshot of the current situation and therefore we suggest that only the current condition are thoroughly analysed and that the cascading risks are documented but not handled right away. This would, according to us, provide for not missing risks in the future and saving time because of the possibility to handle the documented risks as they occur. By accepting the fact that a risk might be a problem at the same time and that a problem could lead to other risks and problems in the future, a logical time frame could be set up to restrict the handling of future possible conditions. Those are not to be handled right away, but documented for future possible use.

As described in the experiment result part of this thesis, the SEI Taxonomy Based Categorisation along with the CTC format help a lot when categorising risks because of its natural dividing of cause and effect. This combination makes it possible to sort the CTC formulated risks on their conditions. When rewriting the non-CTC formatted risks to CTC format, the conformance in risk categorisation increased dramatically. This fact gives support to that a proper way to solve this problem is to divide conditions and consequences. This is done by using the CTC format.

The essential way of thinking is according to us, to focus on the present conditions and their possible consequences. By doing this you will effectively decrease the amount of risks to handle right away and you will at the same time be able to monitor the whole chain of possible emerging risks. The standardised formulation with clear cause and effect also facilitates the categorisation procedure which is discussed under the experiment paragraph. The degree of decomposition and the length of the risk chain...
will be discussed in a forthcoming paragraph. Nevertheless the building of risk chains is important in that way that you might use materialising risks predicting coming ones. In this way you naturally identify early predecessors (symptoms) to risks further along the chain. [DeMarco 97]

7.8 Format

7.8.1 Format: problems

In our project some risks were formulated as questions and other as statements. This created some confusion when managing the risks. The non-unified format invited to free interpretation of risks and therefore, depending on who did the management that time, the values of impact and probability varied over time despite the fact that no known related factors had changed. This could lead to overhead time in managing and values that are not accurate or valid enough to make decisions from. Some risks identified were handled in the software development model already – Some risks and solutions are already presented in the software development model (if existing in the company) and there is a question if those risks are to be handled within the risk management domain. A risk with not handling those issues from the software development model as risks is that they might be forgotten if not present in the risk management document.

7.8.2 Format: solutions

In order to get a consistent format for all your risk statements, we simply suggest that you use the CTC format for all risks. This solves the problem with having a lot of different formats on all risks in the risk document. This in turn allows you to easier overview the entire risk document and to understand the risk statements. [Kotonya 98] It also makes the risk set more complete. [Kotonya 98] It also allows, as shown in the experiments result part, for easier categorisation of risks because of the fact that you can easily see what the condition is and what the consequence of the risk is. Regarding the interpretation of the risk statement, the format provides you with a distinction of condition and consequence and therefore no one will confuse these two essentials regarding risks. A thorough analysis of the previously performed identification should also have been done so that everybody knows what the risk statement means. Also the condition clause of the expanded CTC format allows for further explanation and clarifying of the risk condition and its circumstances.

Our conclusion, there is none to find in the literature, is to address all identified risks even if they are handled in the software process model. Our thoughts of a risk is that if someone experiences a current condition stressful or uncertain in terms of outcome, then it is a risk and therefore it should be included in the risk document. Even the fact that the entire software process model could be seen as a condition leading to concern, there might be parts of it doing the same. This should according to us be reason enough to address the risk even if it is represented in, or even if it consists of, the software process model.

By collecting identified risks over a larger collection of projects, you might recognise common problems that ought to be added to the standard development process model as a form of software process improvement (SPI). [Zahran 98]

The level of risk driven project management should also initially be defined to clarify what parts that are incorporated in the software development model should be used and duplicated (read redundancy) in the risk management document. If all risks identified are present in the risk management document, we efficiently eliminate the risk of risks not handled because of being already present in the software development process.
7.9  Exemplification

7.9.1  Exemplification: problems
All risks identified were not exemplified and therefore providing the opportunity of misinterpretations. There is also a risk that exemplifications of risk categories might narrow the category down in an unwanted way and thereby excluding risks from being sorted in on the right place. You could also imagine the risk manager thinking in a broader sense when constructing the risk statement than what he/she gives the impression of in the exemplification. This could lead to that the one getting the risk delegated excludes some aspects of the risk that should (according to the risk manager) have been included by the formulation.

7.9.2  Exemplification: solutions
By using the expanded CTC format you get to give the reader a clear picture of where to attack the risk or emerging problem. This is described by the condition clause in the CTC format. [Gluch 94] By further making use of the context clause, you can give a wider picture of the condition leading to concern. While constructing the context, it is fairly important to analyse the parameters affecting the condition probably leading to transition. It is also important to make clarifications of expressions and abbreviations not mentioned in the risk statement itself. A good way of writing this part of the expanded CTC format is to imagine the reader as a non-technical, external person. With this in mind, we think that the formulation will be easier to read for everyone. The context part of the expanded CTC structure is of uttermost importance because of “the transient nature of people’s memories.”[Williams 99] This implies that a statement written some days ago and sounded crystal clear to everyone involved, might become totally useless and non-understandable a couple of days later. This gives that the context of a risk is there to facilitate for the reader what affects the condition, what the risk means and clarifying abbreviations and technical terms. When making reports about progress and setbacks to upper management, you could easily use the risk statements to describe the current condition and the possible consequences thereof. Of course this kind of reporting requires an open, non-hostile and risk aware environment because of fear of reprimands reporting risks would decrease the willingness to report setbacks and concerns, [Humphrey 97] but nevertheless you could easily transform your prioritised risk document into an external progress report. Given the CTC format when doing this, you are not only sure to give the receiver of the report a picture of what problems might occur, you also bring forth the condition that gives cause to concern and from this you might easier create an appropriate and more understandable action plan.

7.10  Impact and probability

7.10.1  Impact and probability: problems
As we have mentioned before, all identified risks aren’t possible to predict values on because of sloppy formulations. This has according to us lead to the most of the confusions and conflicts during the risk management process. Here the question is whether it is the formulation or the evaluation system that has been erroneous. Before going in to this discussion we will try to sort out the concepts of impact and probability. Probability is simply the possibility that a risk materialises into a problem. The impact is a measurement of how badly the problem could hurt the project. These terms are often graded into 5-level scales. To take a real-life example from the QUB-project: “There is a risk that a project member gets ill”. This is an exemplification of a non-measurable risk according to us because of its formulation. The impact 5 (on a 5-level scale) would probably mean that a project member gets so ill that he can’t return to work in the project at all or at least that it is of such severe kind that the project will
almost certainly fail. The risk could possibly be interpreted in other ways, but to simplify the exemplification we choose this interpretation for the entire paragraph. The impact can be seen as connected to the probability in a relational way in such manner that the longer period of illness, the lower probability. This was probably the issue that lead to most problems and disputes in our risk management process in the QUB-project.

7.10.2 Impact and probability: solutions
To solve this problem according the CTC and the decomposition of risks, we would end up with a non-manageable amount of risks. The illness period could easily be divided into measurable, discrete sub-elements of the original risk such as: illness period 1 day, illness period 2 days and so on. We soon realise that this is not a proper way of handling this kind of problem. Maybe it really is so that we cannot solve this problem with formulation alone?

According to our experience from software project risk management we find that the numbers of the impact scale alone are not enough to predict the probability of a certain consequence to occur. Trying to reformulate and rephrase the risk in several ways we still end up with a risk where the impact is dependent on the impact and vice versa. This forces us to attack the problem the other way round. Let’s take a look at the measures of a risk; the probability of it to occur and its impact on the project if it does. Our suggestion is that, for a risk to be valid the risk should be measurable regarding impact and probability and these two parameters should preferably be separately measurable. In the example that “There is a risk that a project member gets ill” we feel that there is a strong connection between impact and probability because the longer time (higher impact), the less probability of the unwanted consequence to occur.

From own experience (both from software projects and health care (author working as registered nurse)) we draw the conclusion that there is a higher risk that someone gets ill for a day or two than for six months. Given this connection between impact and probability we feel that using the CTC format when formulating the risk statements is still adequate but it doesn’t solve the problem. [Patterson 98] has solved this problem by assigning a specific value to each impact step of the impact scale, picture 7.10.2.a. In this reference there is an example of how to create an impact scale of cost for some arbitrary risk with effect on this parameter. The impact scale used here is a five-level scale numbered from 1 to 5. Each of the steps has been given a certain calculated value related to the total project budget as depicted in the chart below. The values of the impact steps may also be written in absolute figures and in this case in real money. The cost percentages in the figure are calculated on the entire project budget.

<table>
<thead>
<tr>
<th>IMPACT LEVEL</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Minimal or no impact</td>
</tr>
<tr>
<td>2</td>
<td>&lt; 5 %</td>
</tr>
<tr>
<td>3</td>
<td>5 – 7 %</td>
</tr>
<tr>
<td>4</td>
<td>&gt; 7 – 10 %</td>
</tr>
<tr>
<td>5</td>
<td>&gt; 10 %</td>
</tr>
</tbody>
</table>

Picture 7.10.2.a, General exemplification of impact levels

This helps everyone to understand what the impact level really means to the project. You may optionally add a textual description further explaining the consequences if the problem should occur. As in the example from the QUB-project regarding the risk of someone getting ill, you could formulate your impact scale in the following way, picture 7.10.2.b.
Additionally we must clarify that the scale can be modified to fit the current situation as suitable. It could actually be that way that it would mean disaster to the project to lose one project member for 3 days. Then of course the scale top value should reflect this value instead of the > 1 month-value stated in the chart above, picture 7.10.2.b. When it comes to assign values to the impact scale, the most suitable person to do this ought to be the person that should handle the risks in the future. In this master thesis we don’t state any demands on the scale to be equally scaled or measured in absolute or relative values. The issue solved by assigning specific values to the impact scale is to go round the problem with risks formulated in a way that makes you ask “what is the possibility of impact 4?” for example, and where the answer would be: “It depends…” By structuring the impact scale in this way you could actually historically verify the probability of each level by retrieving data from the company’s database describing the amount of sick days of your staff.

The solution to this problem is to assign specified values to an impact level in order to be able to predict its probability to occur. [Patterson 98] Our suggestion to the problem of getting two or more impact levels with the same probability is simply to reconstruct the scale to make it adapt to the probability scale so that each of the impact values takes one of the defined probability values. Also the probability levels could be further explained in text and/or percents to facilitate the assessment procedure. The textual description could consist of words as: “Not likely”, “Low likelihood”, “Likely”, “Highly Likely” and “Near certainty”. [Patterson 98] The important things here are to be sure, and agree on, that everybody concerned understand the meaning of the textual representation of the used words. There is a survey showing that the word “PROBABLE” for example was interpreted to mean likelihood from 25 % to 90 % for a problem to occur. [Boehm 93]

The figure below, picture 7.10.2.c, depicts a possible way of transforming a 5 step numbered exposure scale to a three step textual scale. [Boehm 93] Please note the difference in exposure for 3*2 and 2*3 respectively. The dark grey area represents “Low” exposure. The white area represents “Medium” exposure and the light grey area represents “High” exposure.

<table>
<thead>
<tr>
<th>IMPACT LEVEL</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Illness &lt; 1 day</td>
</tr>
<tr>
<td>2</td>
<td>1 - 2 days</td>
</tr>
<tr>
<td>3</td>
<td>2 days - 1 week</td>
</tr>
<tr>
<td>4</td>
<td>&gt; 1 week - 1 month</td>
</tr>
<tr>
<td>5</td>
<td>&gt; 1 month - whole project</td>
</tr>
</tbody>
</table>

Picture 7.10.2.b, Modified impact levels regarding illness
The reason for bringing this figure up is that we in our project made no difference between risks with the same exposure but different degrees of impact and probability. We do agree with this representation though. The criterion for a risk is that it materialises one time or else it is a problem. The question of interest here is; what risk would you sign up for insurance on? The risk with an impact of level 4 and a probability of level 2 or a risk with the probability of level 4 and an impact of level 2? The issue is to show the reader that it is more to risk management than just finding and documenting risks. You have to prioritise them and act upon them as well.

### 7.11 Naming

#### 7.11.1 Naming: problems

Naming and tagging of risks is according to us essential in terms of traceability, possibility to get a quick overview and understanding. When it comes to traceability, as mentioned before, risks do appear in chains and they can be categorised in different ways. In the QUB-project we used tagging of the risks, but we never appreciated the possibility to find early predecessors or general problem domains containing linked or clustered risks. Regarding the overview of risks, the important issue would be to get a quick overview of all risks in the risk document and get a fair picture of what risks that is in the project. The problem when it comes to understanding is that by naming the risk in an erroneous or ambiguous way you might have to read the whole risk description to find out what it is about. The paragraph below will try to exemplify the solutions to these problems by using the following CTC-formatted risk statement gathered from the QUB-project.

Given (that) <The customer has indicated that he will be very busy during the project> then there is concern that (possibly) <it will be difficult to meet with the customer>

#### 7.11.2 Naming: solutions

In alignment with our belief that unique identification of risks is needed, we turn to the requirements engineering literature for answers. [Kotonya 98] states that to uphold traceability of requirements, they need to be named with a unique label. We would here like to transfer this thought to risks as well. According to us a risk needs to be uniquely identified to get traceability through the chains of connected risks. The naming of the risks should also give a hint of where in the categorisation it is located. Our suggestion is to let the tag reflect the absolute search path through the tree of categories. If you use the SEI Taxonomy Based categorisation structure, and you have the risk stated above sorted in under “Program Constraint/Program Interfaces/Customer” then you should write the tag: “PC.PI.Customer” followed by a unique numerical identifier for the risks in this category and finally the chosen textual name. The textual representation of the name of the risk should according to us be

<table>
<thead>
<tr>
<th>Probability</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 5 10 15 20 25</td>
<td></td>
</tr>
<tr>
<td>4 4 8 12 16 20</td>
<td></td>
</tr>
<tr>
<td>3 3 6 9 12 15</td>
<td></td>
</tr>
<tr>
<td>2 2 4 6 8 10</td>
<td></td>
</tr>
<tr>
<td>1 1 2 3 4 5</td>
<td></td>
</tr>
</tbody>
</table>

![Probability/impact scale](image)
divided into two parts. Because of that there might be risks sharing the same condition or the same consequence, there is a need to include parts of both to ensure unique names. The name should according to us be short to get an overview when sorting the risks and it should be divided visually into two parts showing the condition and the consequence. Our suggestion for the textual risk name representation will therefore be:

- PC.PI.Customer.1 Busy customer; meeting difficult,
whereas the unified format for risks would be:
- <Category level 1 abbreviation>.<Category level 2 abbreviation>.
  <Category level 3 full name>.<Unique increasing number> <Short condition description>;<Short consequence description>

This format would according to us provide the reader with a notion of the content of the risk when looking at the name, it will be unique within its category because of the number [Connolly 99], it will be unique in the entire risk document because of the existing names of the categorisation levels and it will be easy to sort in to get and overview [Connolly 99] and also being able to prioritise and monitor the top N risks.[Monarch 95] Because of not removing risks from the document (as well as with requirements) there also is a guaranteed traceability when linking risks that emerges sequentially. There is though a chance that risks may be identified that look rather alike. The only three possibilities that should lead to consolidation are:

- manifestation of the same risk statement; that is, identical in every way except in the wording of the statements
- fragmentation due to minor variations or different aspects of the same risk statement
- differences in granularity; for example, a minor risk statement which is covered in the context of another risk statement of larger magnitude.[Higuera 96]

7.12 Language

7.12.1 Language: problems

As in all specifications and descriptions, the least you can demand is that the risks are written in a readable and clear language. An example here can be taken from our project. As a definition of the only risk found under the category negotiations we read: “We find this as a risk regarding to whether we are successfully negotiating requirements or not.” This can hardly be seen as a natural statement in the English language. The project members were not fully acquainted with this language because all of none of them having English as native language. Especially when using a foreign language, you must be careful with how you spell and construct statements.

7.12.2 Language: solutions

This problem might seem a little bit naïve to bring up in a software engineering master thesis, but nevertheless the problem occurred in the QUB-project. Due to lack of information regarding this specific issue, the content of this paragraph will be based on our experiences alone. In our case, the sometime lousy language made it difficult to read the risk statements and it sure didn’t help in increasing the status of the risk document in our project. Because of us thinking that the formulation of risk statements is a very important issue, we would suggest that the statements are written by someone that really recognises the value of writing risks right and preferably, the risk document should be reviewed by someone else except from the author. As we’ve mentioned before, a review would add more effort and time loss to the project and that would be a contradictory to the implicit wants of the project managers interviewed in the survey used as a foundation for this master thesis. What they wanted was a swift and intuitive
process and as we have written before, a review of such a, probably extensive, document sure would take a lot of time. The suggestion made by the authors of this thesis to decrease the impact of this problem is to at least use a spell checking and grammar checking tool to facilitate the creation of risk statements. You should also assure that the person formulating the risk is skilled in expressing him/herself using the chosen language.

7.13 Summary

The most overall important thing about risk statement formulation is to provide the reader with a fully understandable format and formulation of the statement. The format suggested in this thesis is the CTC format: “Given (that) <Condition> there is concern that (possibly) <Consequence>”. This gives a unified format that separates cause from effect, which is very useful when categorising risks. The question that needs to be answered is if the reader can understand the statement. The person best fit to use the statements is the one that wrote them.

When it comes to completeness, we don’t talk about identifying and managing all risk. We mean to talk about the completeness of a certain risk statement. By asking yourself two subjective, binary questions: “Do I have enough information to understand and act upon this risk?” and “Can impact and probability be measured based upon this formulation?” you get a measure of whether the statements are fully adequately formulated or not. Using the context paragraph of the CTC format, you can include all important aspects affecting the risk that you are to manage. The negative about the CTC format here is that it doesn’t ensure you that the context paragraph is accurately written.

Ambiguity is another aspect of formulation problems. You need to use words and abbreviations that either are common to all readers or you have to explain them further in, for example, a wordlist. The use of comparative words such as small, unsatisfying etc. shall preferably be avoided. By checking readability from an external reader’s point of view and ensuring that the statement is measurable in terms of impact and probability the risk of ambiguity will decrease. To keep in mind is that every reader makes assumptions and interpretations of the written word, so the oral part of risk communication is not to be forgotten.

Using standard templates and formulation formats (CTC in our case) will increase understandability, reduce redundancy and left out risk and will on the same time increase the status of the risk management document as an internal work product. The risk templates shall preferably be made within the company to support assimilation with other documents. By using these standards you facilitate categorisation of risks as well.

In order to get the risk statements on a proper level of generality, we suggest that you form chains of cascading risks leading forth to one of the ultimate project constraints: “Time”, “Function”, “Quality” and “Money”. By doing this you provide yourself with a cascading symptom chain. This chain can be used to identify emerging problems early. Breaking down risks with multiple conditions and/or consequences can lead to an unmanageable amount of risks and the judgment of the risk manager should be consulted when to do a breakdown or not.

As said before the CTC format separates condition and consequence of a risk. This facilitates categorisation and makes the risks easier to overview. It also points out where to prevent the problem from occurring and where to let the contingency plans take effect.
When finding a risk that is part of the software process model, or the process itself, we recommend that you include this risk in the risk document even though it has been addressed before. This to avoid risks left out and also giving the company a chance to improve the software process model with solutions to risks found in projects.

When using exemplifications in risk statements, the exemplification needs to be very thorough and should preferably be written for an external, non-technical person. Ambiguity must be avoided, and this can be done with wordlists and so on. Because people tend to forget parts of what they hear and later on re-interpret things, it is of great importance that the context of the risk statements are written down as agreed in close connection to the identification session. An oral agreement between people connected to the risk about the validity of the risk is to prefer. A proper risk statement with thorough context can easily be transformed to an external progress report and thereby facilitating this process.

Impact and probability is the natural way of measuring the damage a materialising risk has to a project. These measures are not always easy to define if not using proper risk statements. The problem is not solved by the CTC format or formulation only. The scale of impact and probability can (should) be assigned discrete interval values to ensure measurability. Using numbers on the scale instead of words increases the common notion of how dangerous a risk is. Even though two risks get the same value when multiplying impact and probability to get the exposure of a risk there could be a difference between a risk, with an impact of three and a possibility of two, and another with the opposite condition.

Naming of risks when categorising them should provide for uniqueness, understandability and condition source. Our suggestion is to name the risks in the following manner:

- `<Category level 1 abbreviation>.<Category level 2 abbreviation>.<Category level 3 full name>.<Unique increasing number> <Short condition description>;<Short consequence description>`

Finally the language used when constructing the statements need to be fully understandable. If a foreign language is used, stick to the common knowledge. If time allows, which it certainly does not, perform a review to correct the statements. In any case a word processor with grammar and spell check should be used. You also need to assure the competence of the risk statement creator regarding the language used.
8 HOW TO CATEGORISE RISKS

We begin this chapter discussing problems and solutions with categorising risks, emphasizing the great need for a proper categorization process and system.

8.1 Motivation for risk categorisation

According to most of the literature on the subject the main reason for categorisation should be to get a more manageable risk-document with greater possibility to overview risks. [Down 94] [Pfleeger 98] It is not easy to overview hundreds of risks if they are just put into one long list. It is also in the nature of the human to have a need for categorising things. Peter Gerzon [Gerzon] also mentions other benefits of categorising risks. Among the benefits he mentions identification of gaps that might been overseen, easier identification of further risks, easier identification of as well double counting and dependencies as correlations.

- **Aids understanding** - Without categories the risk register would seem like a big mess and it should be difficult to understand it. The search within a great register would be nearly impossible and it would be hard to overview the risks. By categorising you also record the thought process you had at the time for risk identification for every risk. [Gerzon] This because of that you can see which aspects of a risk you consider by looking at the category you have placed it in. If you didn’t have any categorisation you should have to add more describing information to each risk. [Down 94]

- **Identifying gaps** - With a proper and well thought through risk categorisation you will have assist in identifying gaps between risks that individuals or subgroup-units may not recognise. [Gerzon] This will lead to better possibility to come up with more thorough action plans against the identified risks, which in turn makes the whole risk management process more useful for the project.

- **Easier risk identification** - Risk categorisation will also aid you in the identification work since you easily can use the categories as a checklist for where to find the risks that are connected to the present project. Category checklists are a very commonly used risk identification technique. [Gerzon] [Down 94] [Nicholas 01] [Pfleeger 98]

- **Identifying double counting, dependencies and correlations** - By having the risks that relates to each other grouped together you more easily find double counting. It will also be easier to recognise the dependencies that are between risks and in this way create more efficient plans to avoid the risk or decrease the impact of the risk. Another advantage is that you can recognise how the different risks correlate to each other. This is important if you want to track a risk to its origin. [Gerzon]

8.2 Problems with risk categorisation

The main reason to make use of categorisation should be to get a more manageable risk-document. It is not easy to overview hundreds of risks if they are just put into one long list.

During the QUB-project we faced some problems with the categorising of risks that has to be solved before you can implement the entire EbRM model. The same conclusion was drawn while making the risk analysis for this master thesis project. These problems are explained below. The advantages of categorisation and organisation regarding risks are described by Peter Gerzon [Gerzon]. He states that
there are a lot of benefits in organising risks. Among the benefits he mentions the possibility to overview risks, identification of gaps that might been overseen, easier identification of further risks, easier identification of as well double counting and dependencies as correlations.

8.2.1 A risk fits into more than one category
Often when you have a set of categories you face the problem that a risk easily could fit into more than one of the categories. Which category should the risk be put into then, or is this an evidence of bad choice of categories? An example of this from the QUB-project could be that the risk “faulty use of competence” could be placed both under the category “organisation” and “competence”.

Another risk with this is that you easily can miss some risks because you think that the risk is put under another category. In the same way it could also be redundancy with the same risk in more than one category. Kotonya and Sommerville [Kotonya 98] define this as unnecessarily repetition of information. Regarding our risk document, we had multiple instances of a risk specifying misunderstandings in communication. As we have been taught, communication is one of the more problem filled domains in software engineering and of course we had to bring it up, but due to an inconsistent system of categorisation, this risk occurred as well under customer relations as internal communication. The risks were mainly the same but the action plans and the preventions differed. Besides the fact that redundancy actually demands more time to handle it also means the risk of missing (skipping) the monitoring thereof. This because of it giving the impression that you have already taken care of this risk during this monitoring session (because of the similar formulation of the different instances of the risk).

8.2.2 A risk does not fit well into any category
This is quite hard connected with the previous problem. The reason is the same; you do not have the right set of risk categories. What do you do if you identify a risk that does not fit into any of your risk-categories? Create another category? Perhaps you can let every project have their own categories, perhaps not. It might be easier if there were some standardised categories in advance, and the project could focus on these. Maybe it should be free for the projects to add some subcategories on a certain level? In the QUB-project we solved this problem in a kind of naive way; we just added a “various” category where every risk that didn’t fit into any other category was placed.

Is there a finite/manageable set of risk categories that allows a risk to always fit into one, and only one category? And in that case, which are those categories?

8.2.3 Lack of intuitivism of the category names
Often when you read through a list of category names you have problems with placing your risks under the right category because you do not know exactly what they mean. An example is the risk-grouping categories suggested by Down [Down 94]

- Schedule
- Function
- Quality
- Process
- Business
- Resource
- Third parties/vendors
- Dependencies
- Constraints

We think that it is quite hard to, only by the name of the categories, group the identified risks. Sure, you can add a description of the category name but the
intuitiveness of the name lacks anyway. A drawback here is that most of us want to use just one word for a category name. Perhaps it would be easier if we let that go and let us use perhaps a whole sentence. In this way it should be much easier to understand what kind of risks that should be put under that category.

8.2.4 Lack of organisation and storage of risks
We as well as the authors of the requirements engineering book understand the importance of organising risks in the storage media. This media could of course be of any form (paper, word processor documents or databases and so on). What we experienced was that organisation in categories is a neat way of handling a large number of risks, but this requires a categorisation that is natural, easy to overview, intuitive and with an underlying process that helps you to organise your risks into those categories. In the QUB-project we, for example, experienced some redundancy which we blame on the not that well structured categories. The risk describing insufficient communication was inserted under each of the main actors in the project: the customer, the line organisation, between the project members and between subgroups in the project. All these risks can in some way be seen to belong to the same main category; "communication" and thereby we could have reduced a lot of redundancy and overhead time. We would in this way also have seen if we had a general communication problem, something that wasn't easy to do with our categorisation, and through this indication rearranged our entire way of communication. Now these communication risks had no logical connection to each other and therefore were treated as independent risks.

8.3 The results for the categorisation experiment
Our experiment showed us that the Taxonomy Based Categorisation system, described earlier, could handle these problems for our sample risks. 100% of the risks could be categorised and the two independent experiment groups had a concordance in nearly 87% which means that the system is quite intuitive to use. It would in the future be very interesting to see if the TBC system manages this for a greater amount of risks in a sharp environment.

8.4 How to create your own categorisation system
We tend to quite often implement our own categorisation system for things in our daily life. For example you categorise the food in your refrigerator, your stamp collection and you even try to put people you meet into different categories. These categories work quite well as long as it is not too many or too complex things you are about to categorise. The problems starts when you are about to use a “home-made” categorisation system in cooperation with another person, then it has to be quite intuitive in order for you not to make misinterpretations of the categorisation rules that the creator had in mind when he created the system. We have through our experiments and experience found three steps that are good to go through in order to succeed with your categorisation building.

8.4.1 Categorisation pre-study
First of all you have to think-through the “data” that is about to be categorised. It is now good if you know if there are about to be 20 entities or 10 000 entities that will be categorised in your categorisation system. It is also good to think through if there are someone else that are about to use or search in “your” categorisation system. You also have to know if it is the same grade of generality on the entities that are about to be categorised, otherwise there will be great problems to come up with categories. The reason for this is explained in the next paragraph.
8.4.2 Decide what preference to categorise on

If you have 1000 stamps that you should categorise you could do this in a lot of ways. One way is to categorise them on which country the stamp is created, but you could also categorise them after motive, year, value and so forth. The important thing is that you have to decide. It is very difficult to categorise them on more than one of their preferences.

The same thought works for risks as well. You can either categorise them after the physical entity that is the condition for the risk, or the physical entity that is the consequence of the risk. [Down 94] You could also categorise the risks after in which phase, or period, of the project they tend to materialise. [Nicholas 01]

8.4.3 Decide if you need a subcategory system

If you have many entities in your categorisation system it could be good to split up every category in some subcategories. This is quite useful and it can even help up the interpretation of the risk if you have followed the categorisation system down to it. But there are some fallbacks with subcategories, for example there is a risk that when you create subcategories you tend to increase the scope for that head-category. If you place an entity in, let’s say a level three subcategory, which entity must also fit perfectly into the category that is the father to that subcategory and so on. Our experiences tell us that the use of subcategory system also tends to increase the problem with risks wanting to fit into more than one category. This is because you tend to make broader interpretations of the categories than they were thought to, this means that more categories give more overlap.

8.4.4 Problems with categorisation systems for risks

Our experiments and studies show us that it is quite hard to come up with categorisation systems for risks. There are too many variables to take care of in order to make it perfect, but perhaps it doesn’t have to be perfect? You will have to adjust this to every situation, the size of the project and the amount of risks and so on. [Down 94] In some cases you can manage without categorisation systems, in other you maybe just need a couple of categories to have a structure on the risk document. The nature of the EbRM model makes it necessary to have a well structured categorisation system because this is to be used by the entire organisation.

8.5 Summary

We first mentioned why there are great needs for categorising the risks. The advantages are that a well thought through categorisation system:

- Aids understanding for the risks
- Identifying gaps between risks
- Makes the risk identification easier
- Identifies redundancy and dependencies as well as correlations between risks.

According to most of the literature the main reason for categorisation should be to get a more manageable risk-document with greater possibility to overview risks.

The problems that could occur with risk category systems are for example that

- A risk fits into more than one category
- A risk does not fit well into any category
- Lack of intuitivism of the category names
- Lack of organisation and storage of risks

The first two problems could to 100 % be handled by the TBC-system. We also think that the third problem is solved by the TBC-system since we had a concordance in 87 % of the categorised risks. The fourth problem wasn’t really tested by our experiment.
We ended this chapter by mentioning how to create a categorisation system and what to look out for when doing this.


9 **FUTURE WORKS**

This chapter describes things that would be of use to expand the results and findings of this master thesis. The issues described below are left out from this thesis because of lack of time or because of the complexity of the issue.

9.1 **Future works with formulation and categorisation of risks**

- Redo the experiment conducted in this thesis with a larger control group, real industry data and thereafter as a case study in a real target environment.
- Develop a software tool that can make semantic analysis of the contents of the risks and show dependencies and correlations between risks, actors and events.
- Try to find a risk that is not possible to categorise with the SEI Taxonomy Based Categorisation system which would be the only way to prove that the system is not entirely sufficient.
- Try to come up with an own categorisation system that should be able to handle the problems that we mentions in chapter 8.

9.2 **Future works with the EbRM model**

- Investigate and implement all parts of the model, such as; identification, monitoring and so on.
- Create a process for incorporating the EbRM into industry.
- Create a database system to uphold the information gathered from the projects.
- Create a process for incorporating experiences into the software development model.
- Evaluate the outcome of the incorporation of the EbRM in industry.
- Experiments on whether risk management is useful or not in terms of gaining knowledge from others’ mistakes.
10 CONCLUSIONS

The EbRM model is to us a way of not only increasing the effectiveness of risk management and emphasising its importance. It is also a way of incorporate experience reuse (or knowledge management if you like) into an already accepted software development activity. By letting risks and their actions become parts of successor projects, i.e. continuous software process improvement, we believe that the often made mistakes could be reduced and thereby saving a lot of time and money for the software industry.

Regarding the formulation part of this thesis, we would like to recommend risk managers to use the CTC format when managing risks. As we have not been able to find another formulation system, we cannot guarantee though that there is no better system than this but as the experiments show the confidence is rather high. This format separates a risk’s condition and its consequences and by doing this, the CTC format is very suitable when categorising risks. It also makes it possible to create chains of risks which allow the user to early identify predecessors of other risks and thereby making the risk management process even more proactive. As the experiment shows us, the correlation between independent groups’ categorisation increases when using the CTC format and we see this as an indication of higher understandability of risk statements. One thing to keep in mind though is that only using the CTC format does not prevent from bad formulations, misinterpretations or even redundancy. Here a thorough risk statement creation process is needed. Things like cause and effect, measurability and so on is heavily dependent on a risk’s formulation. One must also remember that oral communication is a very important medium when sharing information. Remember that people tend to interpret things in their own ways. Below is a short description of important findings regarding each formulation chapter.

The most overall important thing about risk statement formulation is to provide the reader with a fully understandable format and formulation of the statement. The format suggested in this thesis is the CTC format: “Given (that) <Condition> there is concern that (possibly) <Consequence>.” This gives a unified format that separates cause from effect, which is very useful when categorising risks. The question that needs to be answered is if the reader can understand the statement. The person best fit to use the statements is the one that wrote them.

When it comes to completeness, we don’t talk about identifying and managing all risk. We mean to talk about the completeness of a certain risk statement. By asking yourself two subjective, binary questions:” Do I have enough information to understand and act upon this risk?” and “Can impact and probability be measured based upon this formulation?” you get a measure of whether the statements are fully adequately formulated or not. Using the context paragraph of the CTC format, you can include all important aspects affecting the risk that you are to manage. The negative about the CTC format here is that it doesn’t ensure you that the context paragraph is accurately written.

Ambiguity is another aspect of formulation problems. You need to use words and abbreviations that either are common to all readers or you have to explain them further in, for example, a wordlist. The use of comparative words such as small, unsatisfying etc. shall preferably be avoided. By checking readability from an external reader’s point of view and ensuring that the statement is measurable in terms of impact and probability the risk of ambiguity will decrease. To keep in mind is that every reader makes assumptions and interpretations of the written word, so the oral part of risk communication is not to be forgotten.
Using standard templates and formulation formats (CTC in our case) will increase understandability, reduce redundancy and left out risk and will on the same time increase the status of the risk management document as an internal work product. The risk templates shall preferably be made within the company to support assimilation with other documents. By using these standards you facilitate categorisation of risks as well.

In order to get the risk statements on a proper level of generality, we suggest that you form chains of cascading risks leading forth to one of the ultimate project constraints: “Time”, “Function”, “Quality” and “Money”. By doing this you provide yourself with a cascading symptom chain. This chain can be used to identify emerging problems early. Breaking down risks with multiple conditions and/or consequences can lead to an unmanageable amount of risks and the judgment of the risk manager should be consulted when to do a breakdown or not.

As said before the CTC format separates condition and consequence of a risk. This facilitates categorisation and makes the risks easier to overview. It also points out where to prevent the problem from occurring and where to let the contingency plans take effect.

When finding a risk that is part of the software process model, or the process itself, we recommend that you include this risk in the risk document even though it has been addressed before. This to avoid risks left out and also giving the company a chance to improve the software process model with solutions to risks found in projects.

When using exemplifications in risk statements, the exemplification needs to be very thorough and should preferably be written for an external, non-technical person. Ambiguity must be avoided, and this can be done with wordlists and so on. Because people tend to forget parts of what they hear and later on re-interpret things, it is of great importance that the context of the risk statements are written down as agreed in close connection to the identification session. An oral agreement between people connected to the risk about the validity of the risk is to prefer. A proper risk statement with thorough context can easily be transformed to an external progress report and thereby facilitating this process.

Impact and probability is the natural way of measuring the damage a materialising risk has to a project and its risk of occurring. These measures are not always easy to define if not using proper risk statements. The problem is not solved by the CTC format or formulation only. The scale of impact and probability can (should) be assigned discrete interval values to ensure measurability. Using numbers on the scale instead of words increases the common notion of how dangerous a risk is. Even though two risks get the same value when multiplying impact and probability to get the exposure of a risk there could be a difference between a risk, with an impact of three and a possibility of two, and another with the opposite condition.

Naming of risks when categorising them should provide for uniqueness, understandability and condition source. Our suggestion is to name the risks in the following manner:

- PC.P1.Customer.1 Busy customer; meeting difficult,
whereas the unified format for risks would be:

- <Category level 1 abbreviation>.<Category level 2 abbreviation>,
  <Category level 3 full name>.<Unique increasing number> <Short condition description>;<Short consequence description>
Finally the language used when constructing the statements need to be fully understandable. If a foreign language is used, stick to the common knowledge. If time allows, which it certainly does not, perform a review to correct the statements. In any case a word processor with grammar and spell check should be used. You also need to assure the competence of the risk statement creator regarding the language used.

The main findings regarding the categorisation of risks are that the SEI Taxonomy Based Categorisation system can be seen as fully adequate as the system actually could handle 100% of the risks used in our categorisation experiment. The 141 risks sorted into the system were real software project risks and can therefore be compared to industrial data. Of course a larger amount would give higher significance to this finding and in our effort to categorise risks, every risk identified has been possible to categorise with the system. Though the system is actually constructed to function in cooperation with a risk identification questionnaire, the TBQ, our experiments show us that the categorisation works even with risks already identified. A concordance in 87% of the categorised risks between the two independent groups shows that the TBC-system is quite intuitive and only demands very small amount of instructions. The system can also handle different kinds of risk statement formulations. We hereby feel confident to recommend our colleagues to use the SEI TBC system when categorising risks in future software projects, preferably in combination with the CTC format.
11 REFERENCES


http://www.eurolog.co.uk/eurweb/Pages, 2003-05-20


http://www.eurolog.co.uk/euroweb/Pages, 2003-05-20


12 Appendices

12.1 Appendix A, Taxonomic Group Definitions

This appendix is taken from [Carr 93] and provides the definitions of the taxonomic groups in the class, element, and attribute categories of the Software Development Risk Taxonomy. An overview of the taxonomy groups and their hierarchical organization is provided in Picture 5.1.1 and Picture 5.1.2.

A. Product Engineering

Product engineering refers to the system engineering and software engineering activities involved in creating a system that satisfies specified requirements and customer expectations. These activities include system and software requirements analysis and specification, software design and implementation, integration of hardware and software components, and software and system test. The elements of this class cover traditional software engineering activities. They comprise those technical factors associated with the deliverable product itself, independent of the processes or tools used to produce it or the constraints imposed by finite resources or external factors beyond program control. Product engineering risks generally result from requirements that are technically difficult or impossible to implement, often in combination with inability to negotiate relaxed requirements or revised budgets and schedules; from inadequate analysis of requirements or design specification; or from poor quality design or coding specifications.

1. Requirements

Attributes of the requirements element cover both the quality of the requirements specification and also the difficulty of implementing a system that satisfies the requirements. The following attributes characterize the requirements element.

a) Stability

The stability attribute refers to the degree to which the requirements are changing and the possible effect changing requirements and external interfaces will have on the quality, functionality, schedule, design, integration, and testing of the product being built. The attribute also includes issues that arise from the inability to control rapidly changing requirements. For example, impact analyses may be inaccurate because it is impossible to define the baseline against which the changes will be implemented.

b) Completeness

Missing or incompletely specified requirements may appear in many forms, such as a requirements document with many functions or parameters “to be defined”; requirements that are not specified adequately to develop acceptance criteria, or inadvertently omitted requirements. When missing information is not supplied in a timely manner, implementation may be based on contractor assumptions that differ from customer expectations. When customer expectations are not documented in the specification, they are not budgeted into the cost and schedule.

c) Clarity

This attribute refers to ambiguously or imprecisely written individual requirements that are not resolved until late in the development phase. This lack of a mutual contractor and customer understanding may require re-work to meet the customer intent for a requirement.

d) Validity

This attribute refers to whether the aggregate requirements reflect customer intentions for the product. This may be affected by misunderstandings of the written requirements by the contractor or customer, unwritten customer expectations or requirements, or a specification in which the end user did not have inputs. This attribute is affected by the completeness and
clarity attributes of the requirements specifications, but refers to the larger question of the system as a whole meeting customer intent.

e) Feasibility
The feasibility attribute refers to the difficulty of implementing a single technical or operational requirement, or of simultaneously meeting conflicting requirements. Sometimes two requirements by themselves are feasible, but together are not; they cannot both exist in the same product at the same time. Also included is the ability to determine an adequate qualification method for demonstration that the system satisfies the requirement.

f) Precedent
The precedent attribute concerns capabilities that have not been successfully implemented in any existing systems or are beyond the experience of program personnel or of the company. The degree of risk depends on allocation of additional schedule and budget to determine the feasibility of their implementation; contingency plans in case the requirements are not feasible as stated; and flexibility in the contract to allocate implementation budget and schedule based on the outcome of the feasibility study. Even when unprecedented requirements are feasible, there may still be a risk of underestimating the difficulty of implementation and committing to an inadequate budget and schedule.

g) Scale
This attribute covers both technical and management challenges presented by large complex systems development. Technical challenges include satisfaction of timing, scheduling and response requirements, communication among processors, complexity of system integration, analysis of inter-component dependencies, and impact due to changes in requirements. Management of a large number of tasks and people introduces a complexity in such areas as project organization, delegation of responsibilities, communication among management and peers, and configuration management.

2. Design
The attributes of the design element cover the design and feasibility of algorithms, functions or performance requirements, and internal and external product interfaces. Difficulty in testing may begin here with failure to work to testable requirements or to include test features in the design. The following attributes characterize the design element.

a) Functionality
This attribute covers functional requirements that may not submit to a feasible design, or use of specified algorithms or designs without a high degree of certainty that they will satisfy their source requirements. Algorithm and design studies may not have used appropriate investigation techniques or may show marginal feasibility.

b) Difficulty
The difficulty attribute refers to functional or design requirements that may be extremely difficult to realize. Systems engineering may design a system architecture difficult to implement, or requirements analysis may have been based on optimistic design assumptions. The difficulty attribute differs from design feasibility in that it does not proceed from pre-ordained algorithms or designs.

c) Interfaces
This attribute covers all hardware and software interfaces that are within the scope of the development program, including interfaces between configuration items, and the techniques for defining and managing the interfaces. Special note is taken of non-developmental software and developmental hardware interfaces.

d) Performance
The performance attribute refers to time-critical performance: user and real-time response requirements, throughput requirements, performance analyses, and performance modelling throughout the development cycle.

e) Testability
The testability attribute covers the amenability of the design to testing, design of features to facilitate testing, and the inclusion in the design process of people who will design and conduct product tests.

f) Hardware Constraints
This attribute covers target hardware with respect to system and processor architecture, and the dependence on hardware to meet system and software performance requirements. These constraints may include throughput or memory speeds, real-time response capability, database access or capacity limitations, insufficient reliability, unsuitability to system function, or insufficiency in the amount of specified hardware.

g) Non-Developmental Software
Since non-developmental software (NDS) is not designed to system requirements, but selected as a “best fit,” it may not conform precisely to performance, operability or supportability requirements. The customer may not accept vendor or developer test and reliability data to demonstrate satisfaction of the requirements allocated to NDS. It may then be difficult to produce this data to satisfy acceptance criteria and within the estimated NDS test budget. Requirements changes may necessitate re-engineering or reliance on vendors for special purpose upgrades.

3. Code and Unit Test
Attributes of this element are associated with the quality and stability of software or interface specifications, and constraints that may present implementation or test difficulties.

a) Feasibility
The feasibility attribute of the code and unit test element addresses possible difficulties that may arise from poor design or design specification or from inherently difficult implementation needs. For example, the design may not have quality attributes such as module cohesiveness or interface minimization; the size of the modules may contribute complexity; the design may not be specified in sufficient detail, requiring the programmer to make assumptions or design decisions during coding; or the design and interface specifications may be changing, perhaps without an approved detailed design baseline; and the use of developmental hardware may make an additional contribution to inadequate or unstable interface specification. Or, the nature of the system itself may aggravate the difficulty and complexity of the coding task.

b) Unit Test
Factors affecting unit test include planning and preparation and also the resources and time allocated for test. Constituents of these factors are: entering unit test with quality code obtained from formal or informal code inspection or verification procedures; pre-planned test cases that have been verified to test unit requirements; a test bed consisting of the necessary hardware or emulators, and software or simulators; test data to satisfy the planned test; and sufficient schedule to plan and carry out the test plan.

c) Coding/Implementation
This attribute addresses the implications of implementation constraints. Some of these are: target hardware that is marginal or inadequate with regard to speed, architecture, memory size or external storage capacity; required implementation languages or methods; or differences between the development and target hardware.

4. Integration and Test
This element covers integration and test planning, execution, and facilities for both the contractual product and for the integration of the product into the system or site environment.

a) Environment
The integration and test environment includes the hardware and software support facilities and adequate test cases reflecting realistic operational scenarios and realistic test data and
conditions. This attribute addresses the adequacy of this environment to enable integration in a realistic environment or to fully test all functional and performance requirements.

b) Product
The product integration attribute refers to integration of the software components to each other and to the target hardware, and testing of the contractually deliverable product. Factors that may affect this are internal interface specifications for either hardware or software, testability of requirements, negotiation of customer agreement on test criteria, adequacy of test specifications, and sufficiency of time for integration and test.

c) System
The system integration attribute refers to integration of the contractual product to interfacing systems or sites. Factors associated with this attribute are external interface specifications, ability to faithfully produce system interface conditions prior to site or system integration, access to the system or site being interfaced to, adequacy of time for testing, and associate contractor relationships.

5. Engineering Specialities
The engineering specialty requirements are treated separately from the general requirements element primarily because they are often addressed by specialists who may not be full time on the program. This taxonomic separation is a device to ensure that these specialists are called in to analyze the risks associated with their areas of expertise.

a) Maintainability
Maintainability may be impaired by poor software architecture, design, code, or documentation resulting from undefined or un-enforced standards, or from neglecting to analyze the system from a maintenance point of view.

b) Reliability
System reliability or availability requirements may be affected by hardware not meeting its reliability specifications or system complexity that aggravates difficulties in meeting recovery timelines. Reliability or availability requirements allocated to software may be stated in absolute terms, rather than as separable from hardware and independently testable.

c) Safety
This attribute addresses the difficulty of implementing allocated safety requirements and also the potential difficulty of demonstrating satisfaction of requirements by faithful simulation of the unsafe conditions and corrective actions. Full demonstration may not be possible until the system is installed and operational.

d) Security
This attribute addresses lack of experience in implementing the required level of system security that may result in underestimation of the effort required for rigorous verification methods, certification and accreditation, and secure or trusted development process logistics; developing to unprecedented requirements; and dependencies on delivery of certified hardware or software.

e) Human Factors
Meeting human factors requirements is dependent on understanding the operational environment of the installed system and agreement with various customer and user factions on a mutual understanding of the expectations embodied in the human factors requirements. It is difficult to convey this understanding in a written specification. Mutual agreement on the human interface may require continuous prototyping and demonstration to various customer factions.

f) Specifications
This attribute addresses specifications for the system, hardware, software, interface, or test requirements or design at any level with respect to feasibility of implementation and the quality attributes of stability, completeness, clarity, and verifiability.
B. Development Environment

The development environment class addresses the project environment and the process used to engineer a software product. This environment includes the development process and system, management methods, and work environment. These environmental elements are characterized below by their component attributes.

1. Development Process

The development process element refers to the process by which the contractor proposes to satisfy the customer's requirements. The process is the sequence of steps—the inputs, outputs, actions, validation criteria, and monitoring activities—leading from the initial requirement specification to the final delivered product. The development process includes such phases as requirements analysis, product definition, product creation, testing, and delivery. It includes both general management processes such as costing, schedule tracking, and personnel assignment, and also project-specific processes such as feasibility studies, design reviews, and regression testing. This element groups risks that result from a development process that is inadequately planned, defined and documented; that is not suited to the activities necessary to accomplish the project goals; and that is poorly communicated to the staff and lacks enforced usage.

a) Formality
Formality of the development process is a function of the degree to which a consistent process is defined, documented, and communicated for all aspects and phases of the development.

b) Suitability
Suitability refers to the adequacy with which the selected development model, process, methods, and tools support the scope and type of activities required for the specific program.

c) Process Control
Process control refers not only to ensuring usage of the defined process by program personnel, but also to the measurement and improvement of the process based on observation with respect to quality and productivity goals. Control may be complicated due to distributed development sites.

d) Familiarity
Familiarity with the development process covers knowledge of, experience in, and comfort with the prescribed process.

e) Product Control
Product control is dependent on traceability of requirements from the source specification through implementation such that the product test will demonstrate the source requirements. The change control process makes use of the traceability mechanism in impact analyses and reflects all resultant document modifications including interface and test documentation.

2. Development System

The development system element addresses the hardware and software tools and supporting equipment used in product development. This includes computer aided software engineering tools, simulators, compilers, test equipment, and host computer systems.

a) Capacity
Risks associated with the capacity of the development system may result from too few workstations, insufficient processing power or database storage, or other inadequacies in equipment to support parallel activities for development, test, and support activities.

b) Suitability
Suitability of the development system is associated with the degree to which it is supportive of the specific development models, processes, methods, procedures, and activities required.
and selected for the program. This includes the development, management, documentation, and configuration management processes.

c) Usability
Usability refers to development system documentation, accessibility and workspace, as well as ease of use.
d) Familiarity
Development system familiarity depends on prior use of the system by the company and by project personnel as well as adequate training for new users.
e) Reliability
Development system reliability is a measure of whether the needed components of the development system are available and working properly whenever required by any program personnel.
f) System Support
Development system support involves training in use of the system, access to expert users or consultants, and repair or resolution of problems by vendors.
g) Deliverability
Some contracts require delivery of the development system. Risks may result from neglecting to bid and allocate resources to ensure that the development system meets all deliverable requirements.

3. Management Process
The management process element pertains to risks associated with planning, monitoring, and controlling budget and schedule; with controlling factors involved in defining, implementing, and testing the product; with managing project personnel; and with handling external organizations including the customer, senior management, matrix management, and other contractors.

a) Planning
The planning attribute addresses risks associated with developing a well-defined plan that is responsive to contingencies as well as long-range goals and that was formulated with the input and acquiescence of those affected by it. Also addressed are managing according to the plan and formally modifying the plan when changes are necessary.
b) Project Organization
This attribute addresses the effectiveness of the program organization, the effective definition of roles and responsibilities, and the assurance that these roles and lines of authority are understood by program personnel.
c) Management Experience
This attribute refers to the experience of all levels of managers with respect to management, software development management, the application domain, the scale and complexity of the system and program, the selected development process, and hands-on development of software.
d) Program Interfaces
This attribute refers to the interactions of managers at all levels with program personnel at all levels, and with external personnel such as the customer, senior management, and peer managers.

4. Management Methods
This element refers to methods for managing both the development of the product and program personnel. These include quality assurance, configuration management, staff development with respect to program needs, and maintaining communication about program status and needs.

a) Monitoring
The monitoring includes the activities of obtaining and acting upon status reports, allocating status information to the appropriate program organizations, and maintaining and using progress metrics.

b) Personnel Management

Personnel management refers to selection and training of program members and ensuring that they: take part in planning and customer interaction for their areas of responsibility; work according to plan; and receive the help they need or ask for to carry out their responsibilities.

c) Quality Assurance

The quality assurance attribute refers to the procedures instituted for ensuring both that contractual processes and standards are implemented properly for all program activities, and that the quality assurance function is adequately staffed to perform its duties.

d) Configuration Management

The configuration management (CM) attribute addresses both staffing and tools for the CM function as well as the complexity of the required CM process with respect to such factors as multiple development and installation sites and product coordination with existing, possibly changing, systems.

5. Work Environment

The work environment element refers to subjective aspects of the environment such as the amount of care given to ensuring that people are kept informed of program goals and information, the way people work together, responsiveness to staff inputs, and the attitude and morale of the program personnel.

a) Quality Attitude

This attribute refers to the tendency of program personnel to do quality work in general and to conform to specific quality standards for the program and product.

b) Cooperation

The cooperation attribute addresses lack of team spirit among development staff both within and across work groups and the failure of all management levels to demonstrate that best efforts are being made to remove barriers to efficient accomplishment of work.

c) Communication

Risks that result from poor communication are due to lack of knowledge of the system mission, requirements, and design goals and methods, or to lack of information about the importance of program goals to the company or the project.

d) Morale

Risks that result from low morale range across low levels of enthusiasm and thus low performance, productivity or creativity; anger that may result in intentional damage to the project or the product; mass exodus of staff from the project; and a reputation within the company that makes it difficult to recruit.

C. Program Constraints

Program constraints refer to the “externals” of the project. These are factors that may be outside the control of the project but can still have major effects on its success or constitute sources of substantial risk.

1. Resources

This element addresses resources for which the program is dependent on factors outside program control to obtain and maintain. These include schedule, staff, budget, and facilities.

a) Schedule
This attribute refers to the stability of the schedule with respect to internal and external events or dependencies and the viability of estimates and planning for all phases and aspects of the program.

b) Staff
This attribute refers to the stability and adequacy of the staff in terms of numbers and skill levels, their experience and skills in the required technical areas and application domain, and their availability when needed.

c) Budget
This attribute refers to the stability of the budget with respect to internal and external events or dependencies and the viability of estimates and planning for all phases and aspects of the program.

d) Facilities
This attribute refers to the adequacy of the program facilities for development, integration, and testing of the product.

2. Contract
Risks associated with the program contract are classified according to contract type, restrictions, and dependencies.

a) Type of Contract
This attribute covers the payment terms (cost plus award fee, cost plus fixed fee, etc.) and the contractual requirements associated with such items as the Statement of Work, Contract Data Requirements List, and the amount and conditions of customer involvement.

b) Restrictions
Contract restrictions and restraints refer to contractual directives to, for example, use specific development methods or equipment and the resultant complications such as acquisition of data rights for use of non-developmental software.

c) Dependencies
This attribute refers to the possible contractual dependencies on outside contractors or vendors, customer-furnished equipment or software, or other outside products and services.

3. Program Interfaces
This element consists of the various interfaces with entities and organizations outside the development program itself.

a) Customer
The customer attribute refers to the customer’s level of skill and experience in the technical or application domain of the program as well as difficult working relationships or poor mechanisms for attaining customer agreement and approvals, not having access to certain customer factions, or not being able to communicate with the customer in a forthright manner.

b) Associate Contractors
The presence of associate contractors may introduce risks due to conflicting political agendas, problems of interfaces to systems being developed by outside organizations, or lack of cooperation in coordinating schedules and configuration changes.

c) Subcontractors
The presence of subcontractors may introduce risks due to inadequate task definitions and subcontractor management mechanisms, or to not transferring subcontractor technology and knowledge to the program or corporation.

d) Prime Contractor
When the program is a subcontract, risks may arise from poorly defined task definitions, complex reporting arrangements, or dependencies on technical or programmatic information.

e) Corporate Management
Risks in the corporate management area include poor communication and direction from senior management as well as non-optimum levels of support.

f) Vendors
Vendor risks may present themselves in the forms of dependencies on deliveries and support for critical system components.

g) Politics
Political risks may accrue from relationships with the company, customer, associate contractors or subcontractors, and may affect technical decisions.

12.2 Appendix B, Experiment instructions
The experiment instructions where used both by us as well as the other group that also made the experiment. These were originally written in Swedish in order to increase the readability for the two Swedish groups, but are here translated into English. There are two instruction sets, one for each experiment part.

12.2.1 Experiment 1

TBC

Taxonomy based categorisation is divided in a tree-structure with three levels, see the example below:

<table>
<thead>
<tr>
<th>Class</th>
<th>Product Engineering</th>
<th>Development Environment</th>
<th>Program Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>Requirements</td>
<td>Engineering Specialties</td>
<td>Development Process</td>
</tr>
<tr>
<td></td>
<td>Stability</td>
<td>Formality</td>
<td>Work Environment Resources</td>
</tr>
<tr>
<td>Attribute</td>
<td>Scale</td>
<td>Product Control</td>
<td>Schedule</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Facilities</td>
</tr>
</tbody>
</table>

Instructions

In order to facilitate the categorisation we have constructed a html-system that represents the different levels with the description for each category.

1. This is started by index.html.
2. Open "Experiment1.doc" that contains the 15 experiment risks. Go through step 3-7 for each risk.
3. Read through the present risk. Decide which of the head-categories (Class) the risk fits under and click on that category.
4. Then decide which second-level-category (Element) the risk fits under and click on that category.
5. Then choose suitable end-level-category (Attribute) and click on that. Accept the questions that pops up and a file will automatically be created with the correct
name and showed in “notepad”. If the file already exist it opens and is showed in “notepad”, then you place the description of the new risk under those already there. It is allowed to jump back in the tree-structure if you feel you are on the wrong place, but this should then be documented.

6. In “notepad” you write the number and the name on the risk as well as make a note if the risk was easy or hard to categorise and if you feel that there are more places that you could place the present risk on. If you discover that an earlier placed risk has been faulty categories it is ok to move it, but this must be documented.

7. Close and chose to save the document. Go back to the head-categories in the browser. Repeat step 3-7 until all risks are categorised.

8. When finish, send the text-files to pt99eoh@student.bth.se.

12.2.2 Experiment 2

TBC

Taxonomy based categorisation is divided in a tree-structure with three levels, see the example below:

<table>
<thead>
<tr>
<th>Class</th>
<th>Product Engineering</th>
<th>Development Environment</th>
<th>Program Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>Requirements</td>
<td>Engineering Specialties</td>
<td>Development Process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Work Environment Resources</td>
<td>Environment Resources</td>
</tr>
<tr>
<td>Attribute</td>
<td>Stability</td>
<td>Product Control</td>
<td>Schedule</td>
</tr>
<tr>
<td></td>
<td>Scale</td>
<td></td>
<td>Facilities</td>
</tr>
</tbody>
</table>

CTC

We have found a universal format for formulating risks as is called CTC (Condition Transition Consequence) and that is developed by SEI. It looks like follows:

**Given (that) <condition> then there is concern that (possibly) <consequence>.

The risks that are attached to this experiment are formulated in accordance to this format. We now want you to categories the risks after condition. Besides this the categorisation will be done on the same way as experiment 1. You do not need to consider the consequence part when categorising the risks.

In order not to affect the result, you should not compare or check on how you categorised the risks in experiment 1. This includes that you should delete all files and notes from last experiment.
Instructions

In order to facilitate the categorisation we have constructed a html-system that represents the different levels with the description for each category.

1. This is started by index.html.
2. Open “Experiment1.doc” that contains the 15 experiment risks. Go through step 3-7 for each risk.
3. Read through the present risk. Decide which of the head-categories (Class) the risk fits under and click on that category.
4. Then decide which second-level-category (Element) the risk fits under and click on that category.
5. Then choose suitable end-level-category (Attribute) and click on that. Accept the questions that pops up and a file will automatically be created with the correct name and showed in “notepad”. If the file already exist it opens and is showed in “notepad”, then you place the description for the new risk under those already there. It is allowed to jump back in the tree-structure if you feel you are on the wrong place, but this should then be documented.
6. In “notepad” you write the number and the name on the risk as well as make a note if the risk was easy or hard to categorise and if you feel that there are more places that you could place the present risk on. If you discover that an earlier placed risk has been faulty categories it is ok to move it, but this must be documented.
7. Close and chose to save the document. Go back to the head-categories in the browser. Repeat step 3-7 until all risks are categorised.
8. When finish, send the text-files to pt99eoh@student.bth.se.