Different Conceptions in Software Project Risk Assessment

Höst, Martin; Lindholm, Christin

Published in:
proceedings of Conference on Software Engineering Research and Practise in Sweden, SERPS

2005

Different Conceptions in Software Project Risk Assessment

Martin Höst, Christin Lindholm
Department of Communication Systems
Lund University, Lund Institute of Technology
P.O. Box. 118
SE-221 00 LUND, Sweden
(martin.host, christin.lindholm)@telecom.lth.se

ABSTRACT

During software project risk management, a number of decisions are taken based on discussions and subjective opinions about the importance of identified risks. In this paper, different people's opinions about the importance of identified risks are investigated in a controlled experiment through the use of utility functions. Engineering students participated as subjects in the experiment. Differences have been found with respect to the perceived importance, although the experiment could not explain the differences based on study program or undertaken role in a development course. The results and experiences from this experiment can be used when a larger experiment is planned.

1. INTRODUCTION

During project planning and management, procedures for risk management are crucial. This is, for example, acknowledged by the presence of risk management issues at level 3 in the Software Engineering Institute's Capability Maturity Model (e.g. [1]). The objective of risk management is to identify relevant risks as early as possible in a project, in order to avoid or limit the effect of potential problems, such as project delays and cost overruns. More formally, risk management can be defined as "an organized process for identifying and handling risk factors; including initial identification and handling of risk factors as well as continuous risk management" [2].

Risk management is often carried out in a number of steps, e.g.: risk identification, risk analysis, risk planning, and risk monitoring [6]. During risk identification, risks are identified by relevant people, e.g. by using checklists and brainstorming-techniques. The identified risks are prioritized with respect to their probability of actually occurring in the project and their potential impact. The risks that are expected to have both high probability and large unwanted effects are the most important risks to continue to work with in the process. In the risk-planning step, plans are made in order to either lower the effects of the prioritized risk, lower their probability, or to prepare for what to do if they actually occur. In the monitoring step, the risks are monitored during the course of the project.

There are, of course, no clear and objective rules available for how to prioritize the identified risks in the second step. This is instead carried out through discussions and subjective evaluations, where participants have different values and see the risks in different ways [5]. This means that there is a need to investigate methods that can help decision-makers in this discussion. In this paper the usage of utility functions, as described below, are used in order to investigate the different values. It is, for example, important to know if different people have various opinions about how important different risks are for a project.

Utility functions (e.g. [8]) describe how different people value a property. For example, a utility function could describe how people value the expected life-duration after different alternative medical treatments. If the utility function is linear, a life-duration of $2x$ years would be perceived as twice as good as a life-duration of x years. The utility function does, however, not have to be linear, which affects how people make decisions when choosing different treatments. In software engineering relevant properties to study include, for example, the expected delay of a project and the number of faults that remain in the project after delivery.

Based on the shape of the utility function it is possible to discuss whether different individuals act as risk-averse, i.e. they tend to avoid risks and choose a lower safe gain instead of an uncertain high gain, or risk-seeking, i.e. seeking a possible high gain instead of a more certain lower gain.

Safety critical projects include, as all other projects, a large amount of software. In all projects, risk management is important and especially typical project-related risks play an important role. When it comes to risks that are more related to the product, e.g., the number of persistent faults in the product, they are very important in safety critical systems for two reasons. One reason is obviously that it is important to identify these risks as early as possible in order to secure the quality of the developed product. The second reason is that it is important to limit the number of problems during the project even if the quality of the product with respect to the number of dormant faults is acceptable when the product is delivered. This is because a large amount of changes during a project decreases the maintainability of the code, which may result in later lowered quality, which results in new faults later on.

The outline of the paper is as follows. In Section 2, the Trade-off method for deriving utilities is presented and the usage of utility functions in software risk assessment is discussed. In Section 3 the research method and the research questions are presented, and the results are presented and discussed in Section 4. In Section 5 conclusions are presented.

2. ESTIMATION AND USAGE OF THE UTILITY FUNCTION

2.1. The Trade-off method

The objective of the Trade-off (TO) method is to estimate the utility function for one person, i.e. decide how this individual perceives different values of a factor. First the TO method is explained, and then the usage of utility functions is further discussed.

According to the TO method [8] the subject is iteratively asked to compare different "lotteries". A lottery is shown graphically in Figure 1.

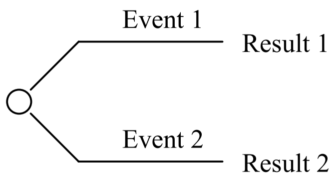

Figure 1. A lottery

Figure 1 shows that one of two events (event 1 and event 2) will occur. I.e., if the probability of event 1 is p, then the probability of event 2 is $1-p$. If event 1 occurs this will result in result 1 and if event 2 occurs this will result in result 2.

An example of possible values for the lottery is shown in Table 1.

Table 1. An example of a lottery.

Property	Meaning
Event 1	Design expert NN is unable to follow the project
Event 2	Design expert NN is able to follow the project
Result 1	The revenue of the project will be 100 KEUR
Result 2	The revenue of the project will be 200 KEUR

The meaning of this example is that either the design expert will be able to follow the project or not. He/she may, for example, be ill during the project or transferred to another project. If the design expert is able to participate in the project, the expected revenue is 200 KEUR and if he/she is unable to participate in the project the expected revenue is 100 KEUR.

In the TO method participants should iteratively compare pairs of lotteries. An example of a pair of lotteries is shown in Figure 2.

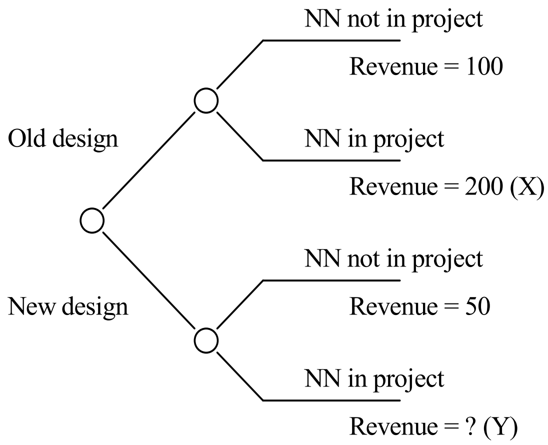

Figure 2. A pair of lotteries.

The upper lottery in Figure 2 shows what could happen if one condition is true (an old design is chosen) and the lower shows what could happen if another condition is true (a new design is chosen). The probabilities of the events are assumed to be independents of the conditions, i.e. the probability that design expert NN will be able to participate in the project is the same in the two lotteries. An advantage of the TO-method compared to other methods for eliciting utility functions is that the value of the probability need not be explained to the person using the method.

In the TO method the subject is first asked to select a value of the revenue in the second lottery (Y in Figure 2) that makes the two lotteries equally attractable. When this has been done the subject is asked to compare two new lotteries. These two lotteries are similar to the first two lotteries, but with value X (see Figure 2) changed to the value that the subject chose in the last question. The subject is now asked to give a new value of Y that makes these two lotteries equally attractable. This process is iterated in order to give values of the utility function for the result factor (i.e. revenue in Figure 2).

If the X-value in the first comparison is called x_0, the first Y-value is called x_1, the second Y-value called x_2, etc., then it can be shown that the utility function u, can be estimated as [8]

$$u(x_i) = i \times u(x_1)$$

which can be normalized to

$$u(x_i) = i \times a$$

where $a = 1/n$, and n is the number of Y-values given by the subject. The proof for this is not provided in this paper; instead the interested reader is referred to [8].

In Figure 3 a hypothetical example of a utility function is shown.

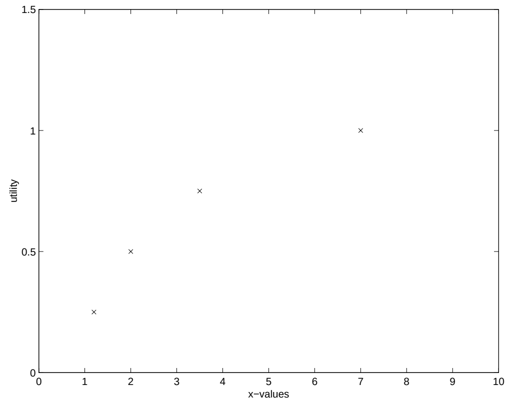

Figure 3. An example of a (concave) utility function.

This example shows a concave curve. If this example denotes the utility of a gain, e.g. in monetary terms, it means that the subject values lower values relatively higher than higher values. Often people with these values are denoted risk-averse, i.e., they prefer lower values with low risk before higher values with higher risk.

The above discussion concerning the meaning of a concave utility function is relevant when the utility function denotes something positive, such as revenue. Since the concept of utility functions, at least as we see it, is easier to understand for results where a large amount is better than a small amount (e.g. revenue) the presentation in this section is based on the revenue as an example. However, in the experiment that is presented in the sequel of the paper, the subjects compared e.g. different values of the number of remaining faults in a program. The value of this parameter should of course be as low as possible.

2.2. Tool

In the TO-method the questions that are asked to the subject should, as it is described in Section 2.1, be based on the previous answer given by the subject. For example, if the subject answered "250" in the last round, then "250" should be one of the results that the subject should compare to in the next round. This means that it is hard to use the TO-method based on completely pre-developed and parameterized instrumentation, e.g. paper forms.

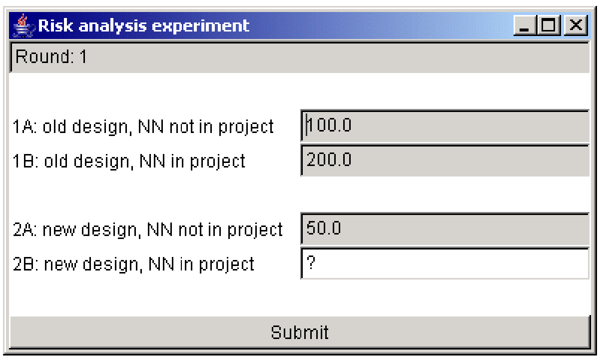

Figure 4. A simple tool (round 1).

For the purpose of the research presented in this paper, a very simple tool was developed. A screen-shot from the tool is presented in Figure 4. From the screen-shot it can be seen that the appearance of the tool was not identical to the questionnaire that is described in [8]. In [8] the decision tree (e.g. Figure 2) was graphically presented to the subjects.

In Figure 4, the user is asked to answer the same question as in Figure 2. If the user answers "250", the next question will be as presented in Figure 5.

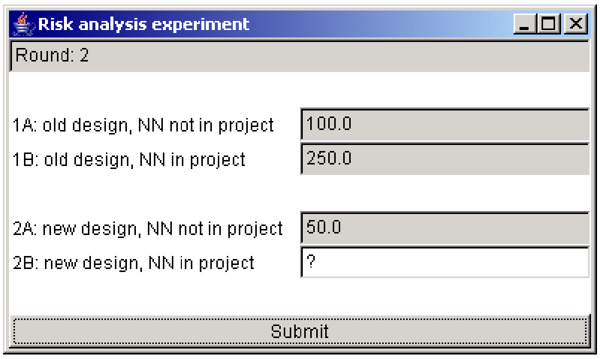

Figure 5. Round 2 after answering "250" in round 1.

2.3. Interpretation of utility functions in software engineering risk assessment

The factors that are considered in software risk assessment often refer to negative aspects and not to positive aspects. For example, factors such as number of remaining faults, delay, etc. are analysed instead of positive factors such as revenue, life-duration, etc. This means that the interpretation of the utility function cannot be carried out exactly as described in Section 2.1.

In [3] the typical shape of utility functions for losses, e.g. in monetary terms, is discussed. In this paper the focus is on determining the shapes and the differences between different people's shape of the utility functions. The focus is not that much on the interpretation of the utility functions. However, a short attempt to explain the meaning of different shapes is given.

If the utility function e.g. for the remaining number of faults is concave (i.e. as in Figure 3) this means that relatively the effect of every fault is higher if there are few faults than if there are many faults. This means that a person with this interpretation thinks that $2x$ faults is less than twice as serious than if there are x faults. If this person would choose between a fixed value x and a lottery with value 0 with probability 1/2 and value $2x$ with probability 1/2, this person would probably choose the lottery since the expected utility value of the lottery is lower than for the fixed value x. Since this person chooses the lottery instead of the fixed value, we say that a person with a concave utility function is risk seeking.

If the function is convex, the value of every fault is higher if there are many faults compared to if there are few faults. We say that a person with a convex utility function is risk averse.

Imagine a situation where a person should compare two different alternative ways of handling a risk in a project. Based on subjective evaluations it might be estimated that one of the alternatives will results in a certain expected amount of remaining faults and the other alternative will results in a higher amount of expected faults. In this case a person with a concave utility function would probably not see the second alternative as negative as a person with a convex utility function. This will of course affect how different people act during discussion on risk evaluation during risk management. It is therefore interesting to investigate how similar utility functions for this type of properties are.

3. RESEARCH DESIGN

3.1. Introduction

The objective of the research presented in this paper is to investigate the shape of utility functions for factors that are relevant in software project risk management. More specifically, the research questions are as follows:

- RQ1: What is the distribution between convex, concave and linear utility functions for properties that are relevant in software project risk assessment?
- RQ2: Is there any difference between different roles in a project with respect to the shape of the utility functions?
- RQ3: Is there any difference between the backgrounds of people with respect to the shape of the utility functions?
- RQ4: Is there any difference between the shapes of the utility functions for normal projects and projects developing safety-critical products.

The research questions are investigated in an experiment where students act as subjects.

3.2. The experiment

The experiment was conducted as part of a software engineering project course given at LTH Campus Helsingborg during the spring of 2005. The students followed programmes in Computer Science, Software Engineering, Electrical Engineering, and Multimedia. The course is attended in the 2:nd year of their university studies.

The course where the experiment was conducted is a project-course where the students work in projects of typically 17 persons in each project. All projects are given the assignment of implementing a number of services for a very basic telephone switching system. In the beginning of the course the students are given a basic version of the system where only basic functions such as providing simple telephone calls, managing what happens if the called party is already involved in a telephone call, etc. are provided. Their assignment is to develop more advanced services such as call forwarding, billing, etc.

The project group should follow a software development process based on the waterfall model with steps such as project planning, requirements engineering, implementation, and testing. This experiment was conducted during the test-phase of the project, i.e. after the project planning was carried out.

In every project groups the students are divided into the following roles:

- Project leaders
- Technical responsibility
- Developers
- Testers

The experiment was conducted during a seminar where all students participated. At the seminar the seminar-leader first held a lecture on risk management, and then the students carried out the tasks of the experiment.

In the experiment the utility function of every student was elicited with the TO-method. The factor of interest was the remaining number of faults after delivery of a software system. In the assignment the students were presented with two scenarios (scenario 1 and scenario 2).

Scenario 1 is based on the project assignment that they were involved in. The scenario was presented as follows (slightly modified and translated from Swedish to English):

Assume that there in your project was a design expert (NN) that could decide the design. NN is part of the "technical responsibility"-group of your project and NN has some new ideas about the design that are not exactly as the teachers in the course has thought about the design. The design proposed by NN is called "new design" and the ordinary design, as proposed by the teachers is called "old design". Based on experience data, the project leaders estimate that there will be a certain amount of faults remaining in the product at the acceptance test.

Consider the following four cases:

Case 1A: The old design is used and NN is able to participate in the project. Then there will be 5 faults at the acceptance test.

Case 1B: The old design is used and NN is unable to participate in the project due to illness. Then there will be 6 faults at the acceptance test.

Case 2A: The new design is used and NN is able to participate in the project. Then there will be 2 faults at the acceptance test.

Case 2B: The new design is used and NN is unable to participate in the project due to illness. How many faults can there be at the acceptance meeting if the two designs should be equally attractable?

Scenario 2 is based on another system than they worked with in the course. It describes instead a safety critical system. Scenario 2 was presented as follows (slightly modified and translated from Swedish to English):

In an intensive care unit you have surveillance equipment connected to the patient that monitors the patient condition. Different values is continuously registered, such as patient's absorption of oxygen, cardiac activity etc. The values are analysed by software in the surveillance equipment. The surveillance equipment sends an alarm, if the analysed values in any way diverge form the normal values. If no attention is taken to the abnormal values (i.e. absence of alarm), it can cause severe injury to the patient and in some cases even death. There is a great risk for serious damage if the alarm fails. The personnel need proper training to be able to connect and manage the surveillance equipment correct. Most of the personnel have this type of training, but some times they do not have the training, due to lack of time. If the surveillance equipment is connected the wrong way there is a risk for absence of alarm and the patient are exposed to danger. Now there is a desire to try new software in the surveillance equipment. Consider the following four cases:

Case 1A: Present software is used. The personal are trained on the surveillance equipment. At 7 occasions in a three-month period, there was absence of alarm from the surveillance equipment, despite the fact that there should have been *alarms.*

Case 1B: Present software is used. In this case personnel who have not received proper training on the equipment use the equipment. At 9 occasions in a three-month period, there was absence of alarm from the surveillance equipment, despite the fact that there should have been alarms.

Case 2A: New software is used. The personal are trained on the surveillance equipment. At 4 occasions in a three-month period, there was absence of alarm from the surveillance equipment, despite the fact that there should have been alarms.

Case 2B: New software is used. In this case personnel who have not received proper training on the equipment use the equipment. How many alarms can be missed if the new software should be equally attractable?

The students were also given instructions on how to use the tool that is presented in Section 2.2. They used the tool when they answered questions iteratively according to the TO-method.

All students first worked with scenario 1 and then with scenario 2. In the analysis the results from each student is characterized as concave, convex, linear or "other". A curve is classified as "other" if it has not the same shape (convex or concave) for all x-values, e.g. the first half of the curve is convex and the second half is concave.

In order to investigate research question RQ1 the data from all students are pooled and the number of curves of each shape is compared.

In order to investigate research question RQ2 the following independent and dependent variables [9] have been defined for the experiment:

- Independent variable: role in project
- Dependent variable: number of curves of each shape

In order to investigate research question RQ3 the following independent and dependent variables have been defined for the experiment:

- Independent variable: study program
- Dependent variable: number of curves of each shape (i.e. the same independent variable as for RQ2)

In order to investigate research question RQ4 the following independent and dependent variables have been defined:

- Independent variable: Scenario
- Dependent variable: number of curves of each shape (i.e. the same independent variable as for RQ2 and RQ3)

That is, for all four research questions, the number of people with a certain shape of the utility function was chosen. The analysis is presented in Section 4.

3.3. Validity

In order to evaluate the validity of the study, a checklist from [9] is used. Validity threats may be classified into the following four classes: conclusion validity, construct validity, internal validity, and external validity.

The *conclusion validity* is related to the possibilities to draw correct conclusions about relations between the independent and dependent variables of the experiment. Typical threats of this type are, for example, to use wrong statistical tests, to use statistical tests with too low power, or to obtain significant differences by measuring too many dependent variables ("fishing and the error rate"). Since, as it is shown in Section 4, there in most analyses, is no possibility to determine any differences with statistical significance, there is no such risk. However, when it is discussed whether this means that there are no different it is important to remember that this also may be due to few data points. This is further discussed in Section 4.

The *internal validity* is affected by confounding factors that affect the measured values outside the control, or knowledge, of the researcher. This may, for example, be that the groups of subjects carried out their assignments under different conditions, or maturation of participants. In order to lower the internal threats in this experiment all students carried out the assignment the same time during a 90 minutes seminar when one of the researchers were present. One threat to this study is that the two scenarios were analysed in the same order by all students. This should be taken into account when the difference between the scenarios is analysed, i.e. when RQ4 is analysed. The reason for letting every participant work with the scenarios in the same order was that it was seen as positive that the students started with a scenario that presents a familiar project and system, i.e. a situation that is related to the course.

Threats to *construct validity* denote the relation between the concepts and theories behind the experiment, and the measurements and treatments that were analyzed. We have not identified any serious threats of this kind.

The *external validity* reflects primarily how general the results are with respect to the subject population and the experiment object. The intention is that the subjects in this experiment should be representative of engineers working with this type of estimation in live projects. As we see it, the largest threat to validity is of this kind. It cannot be concluded with any large validity that the students that participated in this experiment are representative of professional practitioners. Scenario 2 is not in any way related to the students' course work, but scenario 1 was based on the projects that the students participated in the course. However, the scenario was still a hypothetical scenario and it was studied in the testing phase of the project, i.e. after the risk assessment in a real project. According to [4], controlled experiments can be classified according to two dimensions as displayed in Table 2. According to this classification, the experiment could be classified as (I2, E1) with respect to scenario 1 and as (I1, E1) for scenario 2.

In an experiment that is classified as this, it may be important to reflect over how valid the results are. In this case we believe that the results primarily could serve as a basis for continued experiments in the area. It is important to include more experienced people in continued experiments. The results from this experiment are however important when these experiments are planned.

Table 2. Classification scheme

Incentive	Experience
I1: Isolated artefact	E1: Undergraduate student with less than 3 months recent industrial experience
I2: Artificial project	E2: Graduate student with less than 3 months recent industrial experience
I3: Project with short-term commitment	E3: Academic with less than 3 months recent industrial experience
I4: Project with long-term commitment	E4: Any person with industrial experience, between 3 months and 2 years
	E5: Any person with industrial experience for more than 2 years

4. RESULTS

4.1. Results from the empirical study

The experiment was conducted with 47 students, but one of the them did not hand in any results, which means that there were 46 students that completed the tasks of the experiment. The number of subjects that completed scenario 1 was 44, since 2 of the subjects were discarded because the scenario was only iterated three times. The minimum of iterations was set to four times. In scenario 2, 3 subjects were discarded for the same reason so the number of subjects that retained for further analysis was 43 in scenario 2.

In order to investigate research question RQ1 the distribution of utility functions were analysed. The distribution between concave, convex, linier and other utility functions for the two scenarios are displayed in Table 3. The result is presented in percent of the total number of subject for each scenario, and in absolute figures in parenthesis. In scenario 2 the linier utility functions dominate and there are only a few concave functions. Scenario 1 does not show this type of dominance.

Table 3. Distribution utility functions

	Concave	Convex	Linear	Other
Scen.1	20 % (9)	32 % (14)	30 % (13)	18 % (8)
Scen.2	5 % (2)	23 % (10)	58 % (25)	14 % (6)

The students had different roles in their project groups. There is a difference in the amount of students connected to the various roles. The largest group were developers (18 students) and the smallest group were project leaders (6 students). The values for each role and type of utility function are presented in Table 4. It can be seen in scenario 1 that both project leaders and those with technical responsibility show convex utility functions were as most of the developers and testers show linier utility functions. In scenario 2, the linear utility functions dominated for all the roles.

Table 4. Roles and utility functions

Role	Scen	Concave	Convex	Linear	Other
Project leaders	1	17% (1)	50% (3)	33% (2)	0% (0)
	2	0% (0)	40% (2)	60% (3)	0% (0)
Technical responsi-bility	1	25% (2)	50% (4)	0% (4)	25%(2)
	2	25% (2)	0% (0)	50% (4)	25%(2)
Developer	1	17% (3)	28% (5)	39% (7)	17%(3)
	2	0% (0)	29% (7)	65%(11)	6%(1)
Tester	1	25% (3)	17% (2)	33% (4)	25%(3)
	2	0% (0)	23% (3)	54% (7)	23%(3)

Not only the roles but also the background of the students are different. So the background (i.e. study program) was looked at as a variable against the number of curves of each shape. The students come from the four different engineering programs Computer Science, CS (12), Software Engineering, SE (8) Electrical Engineering, EE (9), and Multimedia, MM (17). Table 5 shows for scenario 1 that the students from Software Engineering have the same amount of convex and linear utility functions and so have also the students from Multimedia. The students from the Electrical Engineering program have the same amount of concave and convex utility functions while for the computer science students the majority of the utility functions are linear.

Table 5. Programs and utility functions

Program	Scen	Concave	Convex	Linear	Other
SE	1	25% (2)	38% (3)	38% (3)	0% (0)
	2	0% (0)	17% (1)	83% (5)	0% (0)
MM	1	18% (3)	29% (5)	29% (5)	24% (4)
	2	12% (2)	22% (4)	47% (8)	18% (3)
CS	1	10% (1)	30% (3)	40% (4)	20% (2)
	2	0% (0)	27% (3)	45% (5)	27% (3)
EE	1	33% (3)	33% (3)	11% (1)	22% (2)
	2	0% (0)	22% (2)	78% (7)	0% (0)

In the scenario 2, in Table 5 the pattern is not the same as in scenario 1. For all the programs in scenario 2, the linear utility function occurs most frequently.

4.2. Analysis

The data has been analysed with a number of chi-2 tests [7] as summarized in Table 6. In the analysis, data from people with responses other than convex, concave and linear was discarded.

In analysis 1, a chi-2 goodness of fit test was carried out in order to see whether the three shapes were equally probable. Data from both scenarios was pooled (denoted “1+2” in the 4:th column). It was clear that the shapes were not equally probable. The value of this analysis is limited, but it does show that the shape that results from the method is not completely random. Concerning RQ1, the most important contribution lies in the distribution of the different shapes.

Table 6. chi-2 tests

RQ	Analysis	Independent variable	Data from scenario	p	Chi2 rquirements ok
RQ1	1	-	1+2	0.0006***	Yes
RQ2	2	Role	1	0.94	No
	3	Role	2	0.04(*)	No
	4	Role	1+2	0.73	No
	5	PL+TR vs D+T	1+2	0.66	Yes
RQ3	6	Program	1	0.83	No
	7	Program	2	0.60	No
	8	Program	1+2	0.95	No
RQ4	9	Scenario	1+2	0.012*	Yes

*significant at the 5% level
**significant at the 1% level
***significant at the 0.1% level

Concerning RQ2 in analysis 2-4, there are too few data points to be able to carry out a Chi-2 test that compares the shapes of the roles ("No" in 6:th column). Therefore in analysis 5, data from project leaders and "technical responsibility" was pooled and data from developers and testers were pooled, which means that an analysis comparing "management roles" to more developer-oriented roles could be carried out. This analysis is valid with respect to the number of data-points, but it was clear that there is no statistical difference.

Concerning RQ3, there were not enough data points to carry out a chi-2 test, and no natural way to pool data. The data did not show any clear difference between the programmes. In the analysis of RQ4 it was found that there is a clear difference between the two scenarios. This is, however, confounded with the order in which the two scenarios were analysed by the participants.

4.3. Discussion

The empirical study has shown that the distribution of utility functions varies for different kinds of scenarios. The result from scenario 2 shows dominance of linear utility functions but this type of dominance does not exist in scenario 1. In scenario 2 you might expect more risk-averse tendencies because the scenario concerns severe injury to patients or even death but this is not the case. One factor to consider is the fact that the subjects are used to the tool and know how it works during scenario 2, see Section 3.3.

Based on this study, it has not been possible to state that any role is more risk seeking than any other role. Maybe there are too few subjects in the study and the students have not had their roles long enough to connect to it. There is a difference between the two scenarios. In scenario 1 there is an indisputable majority of convex utility functions for project leaders and those with technical responsibility but that is not the case in scenario 2. This has, however, not been further analysed.

If we look at scenario 2 and the students' background we see the same pattern here that for all the study programs the linear utility function dominate. In scenario 1 you have no obvious connection between study programs and shape of utility function. It can be discussed if the students' backgrounds are so different. All four programs belongs to the IT-programs, they have the same admission criteria and results in a bachelor degree. If we had a group of students with entirely different education and background to compare with we might see a clear difference in risk tendency.

5. CONCLUSIONS

From this study it is possible to conclude that different study participants have different opinions about the faults remaining after testing. Some of the participants are more risk seeking than others. There are, however, no clear connection between the projects-roles and the shape of the utility functions in this study.

There are, as described in Section 3.3, some threats to the validity of this study and future studies will, based on the experiences from this study be changed in the following ways:

- People with more experience in general and with more experience from their project-roles should be involved in the study. If students should be involved, they could probably be sampled from populations with larger differences than students from study programs that to some extent are quite similar.
- If a similar experiment design is chosen, it should be adapted so that all subjects do not work with both scenarios in the same order. There were reasons for choosing this design in this research, but in further studies it is probably better not to have the same order for all participants.

Additional further work includes risk management in general, e.g. building a tool for risk assessment and follow-up, and software development in safety critical development projects.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to the students that participated in the study. The authors would also like to acknowledge Gyllenstiernska Krapperupstiftelsen for funding the research studies of Christin Lindholm.

REFERENCES

[1] CMMI Product Team, "Capability Maturity Model® Integration (CMMI^{SM})", Version 1.1, Staged representation, Software Engineering Institute, Technical report CMU/SEI-2002-TR-029, 2002.

[2] R.E. Fairley, "Software Risk Mangement", IEEE Software, May/June 2005, p. 101.

[3] H. Fennema, M. van Assen, "Measuring the Utility of Losses by Means of the Tradeoff Method", Journal of Risk and Uncertainty, Vol. 17, No. 3, 1999, pp. 277-295.

[4] M. Höst, C. Wohlin, T. Thelin, "Experimental Context Classification: Incentives and Experience of Subjects", in proceedings of International Conference on Software

Engineering, 2005, pp. 470-478.

[5] S.L. Pfleeger, “Risky Business: What We Have Yet to Learn About Risk Management”, Journal of Systems and Software, Vol. 53, 2000, pp. 265-273.

[6] I. Sommerville, “Software Engineering”, 7:th edition, Addison Wesley, 2004.

[7] S. Sidney, N.J. Castellan, “Non-Parametric Statistics for the Behavioral Science”, McGraw-Hill, 1988.

[8] P. Wakker, D. Deneffe, “Eliciting von Neumann-Morgenstern Utilities when Probabilities are Distorted or Unknown”, Management Science, Vol. 42, No. 8, August 1996, pp. 1131-1150.

[9] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, “Experimentation in Software Engineering: An Introduction”, Kluwer Academic Publishers, 1999.