Electronic version of an article published in Empirical Software Engineering,
Vol. 14, No. 6, 2009, pp. 644-684
[doi: 10.1007/s10664-009-9105-0]

Empir Software Eng (2009) 14:644–684
DOI 10.1007/s10664-009-9105-0

Practical challenges of requirements prioritization based on risk estimation

Andrea Herrmann · Barbara Paech

Published online: 17 February 2009

Editor: Daniel M. Berry

Abstract Requirements prioritization and risk estimation are known to be difficult. However, so far, risk-based requirements prioritization has not been investigated empirically and quantitatively. In two quantitative experiments, we explored practical challenges and needs of risk estimations in general and of our method MOQARE specifically. In the first experiment, ten students made individual estimations. In the second one, twenty-four students estimated risks in seven moderated groups. The students prioritized the same requirements with different methods (risk estimation and ranking). During the first experiment, we identified factors which influence the quality of the prioritization. In the second experiment, the results of the risk estimation could be improved by discussing risk estimations in a group of experts, gathering risk statistics, and defining requirements, risks and prioritization criteria more tangibly. This first quantitative study on risk-based requirements prioritization helps to understand the practical challenges of this task and thus can serve as a basis for further research on this topic.

Keywords Requirements prioritization · Risk · Risk analysis · Risk estimation · Risk prediction

1 Introduction

Requirements prioritization is known to be difficult to perform: “Requirements decisions are hard because of the uncertainty and incompleteness of the information available.” (Ngo-The and Ruhe 2005) There are even more factors which amplify this difficulty, e.g.

A. Herrmann (✉) · B. Paech
Faculty of Mathematics and Computer Science, Software Engineering Group, University of Heidelberg, 69120 Heidelberg, Germany
e-mail: herrmann@informatik.uni-heidelberg.de

B. Paech
e-mail: paech@informatik.uni-heidelberg.de

differing perspectives of the stakeholders and dependencies among requirements. One way of identifying priorities of requirements is to assess the risks involved in case a certain requirement is not realized (Berander 2004; Park et al. 1999). This principle of prioritizing requirements based on risk estimation is being used especially in the context of security (as in (Arora et al. 2004)), but makes sense also with other non-functional requirements (NFR) (Feather et al. 2006) when prioritizing countermeasures. Countermeasures are a special type of requirements which are defined in order to reduce risks (e.g. security-related risks) and thus improve software quality. We have applied this principle to countermeasures that are specified with MOQARE (Misuse-oriented Requirements Engineering) (Herrmann and Paech 2005; Herrmann et al. 2006; Herrmann and Paech 2008a, b).

There are no publications of quantitative empirical studies of risk-based requirements prioritization. Therefore, we performed two student experiments to learn about the practical needs of this requirements prioritization principle, such as preparation, material, knowledge and time consumption. We wanted to understand the factors which influence the quality of the outcome. The insights gained during the first experiment helped to improve the outcome of the second experiment.

Before performing these experiments, our prioritization approach had been tested in examples and in one case study. However, the resulting impressions of the quality of the method were subjective and depended on the estimator and the context. Therefore, we decided to undertake a systematic quantitative empirical investigation with a considerable number of participants who all perform the same task, who have a comparable educational background, level of knowledge of the method and of the example system used in the experiment. The participants had to be open for experiments and for using different methods to solve the same problem just for the sake of testing them. The experiment was performed during a lecture. So, several parameters could be controlled, e.g. which material is used and whether the participants perform the tasks in the defined order. We did not expect to find such participants and conditions in a real software project. Therefore, conducting a student experiment seemed ideal for our purpose of performing a pilot study.

The remainder of this work is structured as follows: In Section 2, we summarize some basics of requirements prioritization and risk estimation, including published empirical investigations about prioritization. Section 3 describes the requirements prioritization methods tested, the research questions and the variables which were observed in the experiments. Section 4 describes the preparation and execution of Experiment 1. Section 5 describes Experiment 2 in the same form. In Section 6, the results of both experiments are being discussed and compared, and lessons learned derived. In Section 7, we discuss our conclusions and future work. The annex summarizes data and data analysis for both experiments.

2 Basics of Requirements Prioritization and Risk Estimation

In this section, we list approaches of requirements prioritization, especially those that use risk estimation. Then we present former empirical work on requirements prioritization. These form the basis of the design of our own risk-based prioritization method and of the experiment.

2.1 Requirements Prioritization Based on Risk

Typical criteria for the prioritization of requirements are their benefit (e.g. business value) for the stakeholder, the dissatisfaction if not implemented, their urgency, volatility, risk,

their implementation cost or system impact. They can be estimated in cardinal values (=absolute values) or ordinal values (=relative values, also called ratio scale). There are several methods for determining ordinal rankings, often based on pair-wise comparison, mainly varying in the way the pairs are being combined. Karlsson et al. (1998) describe and compare six such methods: analytic hierarchy process (AHP), hierarchy AHP, minimal spanning tree matrix, bubble sort, binary search tree, priority groups. Another prioritization method is the 100$ method (also called cumulative voting) (Leffingwell and Widrig 2000). Only few of these methods can be used for the determination of cardinal values. Risk estimation is such an approach. When risks and priorities are quantified on a cardinal scale, an existing list of prioritized requirements is more easily scalable and extensible than when ordinal values are used. New requirements can easily be inserted in the list, without the need to compare each one to the whole list of the other requirements.

Misuse cases describe the course of risk events (e.g. attacks, user errors, accidents) which may happen with a certain probability and have a usually negative impact. Traditionally, misuse cases are used to elicit security requirements (Sindre and Opdahl 2000; Sindre and Opdahl 2001). However, MOQARE (Herrmann and Paech 2008a, b) applies misuse cases to all types of NFR. Then, these misuse cases describe unwanted scenarios as well, where the misuser is not a malicious attacker, but might be a user who by mistake impairs data integrity, or a developer who by negligence threatens the maintainability of a software. Misuse cases are used to identify countermeasures, i.e. requirements which, if satisfied, prevent, mitigate or detect misuse cases and thus support the satisfaction of security and of other NFR. Countermeasures can be requirements on the IT system, on its design, its development process, operation environment or personnel.

Risk events have an effect on the benefit and cost of the system. In the security community, the risk of misuse cases is quantified by the product of *probability* and *caused damage* (see for instance in (Kontio 1996; ISO 2002; Xie et al. 2004)). This risk is influenced by which countermeasures are realized or not, but also depends on environmental factors.

Regularly, risk is proposed to not only quantify the importance of misuse cases, but also the benefit of a single requirement. In (Park et al. 1999), WinWin is described to "assign each item [=requirement] a difficulty and importance (or a probability and loss)". Berander (2004) explicitly uses risk estimations as a prioritization criterion. Mayer et al. (2005) propose to integrate requirements engineering and risk analysis "for focusing […] on the most critical parts of the IS." They use the quantitative risks assessment, business criticality, budget and the countermeasure cost as a basis for the requirements elicitation and prioritization.

The Failure Mode and Effects Analysis (FMEA) (Stamatis 2003) also prioritizes failures according to their risk, which in FMEA is defined as the product of the importance of the failure effect, the probability of occurrence of the failure cause and the inability of controls to detect the failure effect or failure cause. Each of these three aspects is rated by a number between one and ten, which results in a risk between 1 and 1,000. (Other approaches estimate probabilities in percent and damages in $.)

Feather et al. (2006) measure the benefit of a countermeasure (there called "mitigation") by the difference between the risk without any countermeasures being implemented (worst case) and the risk with the chosen countermeasure. Arora et al. (2004) also explicitly set the benefit of a countermeasure equal to "the reduced expected loss due to security failure incidents (i.e. reduction in risk)". To determine this risk reduction, they define two types of risk: the baseline risk and the residual risk. The baseline risk is calculated from total incident risks, if no countermeasures were in place. The residual risk is the expected value

of damages, if only one countermeasure was installed. Then, the benefit of this countermeasure equals baseline risk minus residual risk. These authors do not refer to any real case study, only to examples, where supposed values are used.

Although the estimations of risks and risk reduction have often been proposed, we found only two detailed experience reports of authors who actually applied risk estimations. In the SQUARE project, Xie et al. (2004) applied the quantitative principles of Arora et al. (2004) in small companies, but met several practical challenges. They remark that this approach has a practical limitation: It requires high volumes of incident data, ideally from the same company. While big companies generate their own historical security statistics, small companies must rely on public statistics. They report: "detailed attack data are simply not available to be used as references". As public statistics are available only on a high level of granularity, Xie et al. (2004) subsume misuse cases in categories of threat, such as denial of service, system penetration, or sabotage of data. Similarly, countermeasures are summarized in categories. Xie et al. define the baseline risk like Arora et al. (2004), but the residual risk as "incident risk to the organization if security solutions are properly installed, utilized, and monitored". They initially used estimated cost figures from nationally surveyed losses for each category of threats. Later on, they worked with a company and their estimations for their environment. They found that lower ends of nationally surveyed losses may be used as estimations for tangible losses (productivity loss, fixing cost, etc.), but cannot sufficiently account for intangible losses (loss of reputation, loss of confidential data, etc.), since these values are highly company and project specific.

The second group of experience reports stems from the NASA. Feather et al. (2006) performed a high number of risk estimations as well, but they discuss practical challenges only qualitatively. They emphasize the importance of tool support, e.g. for visualization (see also in Feather et al. 2000a, Feather et al. 2000b) and the involvement of experts who cover a wide spectrum of knowledge (Feather and Cornford 2003): "Typical DDP applications have involved 10–20 experts drawn from the disciplines of mission science, project planning, software and hardware engineering, quality assurance, testing, risk management, etc." Their positive experiences from many case studies (e.g. in terms of achieved cost saving during subsequent software development) show that risk estimation makes sense and can be applied to realistically large requirements specifications successfully.

While there is only little practical experience with risk estimation in the field of requirements prioritization, it is different in the decision theory community. Many biases are known (Raiffa et al. 2002) which lead to bad estimations of probabilities, frequencies and values, or as Tversky and Kahneman (1974) put it: "intuitive predictions and judgement under uncertainty do not follow the laws of probability or the principle of statistics. Instead, people appear to rely on a limited number of heuristics and evaluate the likelihood of an uncertain event by the degree to which it is representative of the data generating process, or by the degree to which its instances or causes come readily to mind."

Another challenge of requirements prioritization are the complex dependencies among the benefits of requirements. Such dependencies are critical in practice (Ryan and Karlsson 1997). In one of our publications (Herrmann and Paech 2006), we discuss how such dependencies complicate estimations. For instance, countermeasures can replace each other partly, when they mitigate the same misuse case. The benefit of implementing two countermeasures with dependencies is not twice as high as the benefit of only one, but less. Also, two or more countermeasures may need each other for being effective against a misuse case. In this case, the implementation of one of these countermeasures alone does not add much benefit, only the implementation of all of them. Due to such dependencies, the benefit of a requirement cannot be described by one fixed value only (as is usually done

by prioritization methods) and several benefits cannot be added up. There are two ways of treating these dependencies:

1. One estimates the risk and benefit as a function of which countermeasures are implemented and which are not. This means to estimate these values for all possible combinations of these variables, in the N dimensional space for N countermeasures. When this benefit function is known, the requirements can be prioritized by optimizing this function numerically (Ruhe et al. 2003, van den Akker et al. 2004 und 2006) or with methods from artificial intelligence (Menzies et al. 2003; Jalali et al. 2008).
2. If one does not want to estimate the benefit function completely or if such an estimation is not feasible, one can work with approximations, as most prioritization methods do. Daneva and Herrmann (2008) have identified six approximations commonly applied with prioritization methods, which can be combined with each other.

We do not expect that for larger numbers of requirements, it is practically feasible to estimate a complete benefit function or to model all dependencies among the requirements. Therefore, when prioritizing countermeasures based on risk estimation, we apply two of the six approximations: the reference system and the bundling.

Risk estimations as well as benefit estimations are comparable to each other only when they are made with respect to the same reference system (Herrmann and Paech 2006). A reference system is an idea of a set of requirements which are imagined to be implemented. It is important that the reference system is clearly defined, easy to imagine for the estimators and near to the system that is finally to be implemented. If perfect quality is the goal or the benchmark, the perfect system is the reference, i.e. the system in which all countermeasures are implemented, as in (Xie et al. 2004). The reference system also can be the ensemble of all mandatory requirements, as used by (Ruhe et al. 2003), the former system version, a competitor's product or all FR without any countermeasures (Arora et al. 2004; Feather et al. 2006). The "reference risk" denotes the risk in this reference system. In order to determine the risk reduction effected by the implementation of each countermeasure relative to the reference system, we estimate a "varied risk" in a system identical to the reference system, except for one countermeasure only.

In many prioritization methods, it is common to bundle those requirements which depend on each other most to relatively independent bundles. These bundles have the name feature (Regnell et al. 2001; Wiegers 1999), feature group (Regnell et al. 2001), super-requirement (Davis 2003), class of requirements (Ruhe et al. 2003), bundle of requirements (Papadacci et al. 2004), category (Xie et al. 2004), User Story (Beck 2000), super attribute (Stylianou et al. 1997) or Minimum Marketable Feature (Denne and Cleland-Huang 2003). Bundles are applied as an efficient way of reducing the complexity, time need and effort of prioritization.

These two principles–reference system and bundling—can easily be integrated in any requirements prioritization method, but usually are not applied. Usually, prioritization methods accept as input requirements on an arbitrary level of granularity and attribute a fixed priority to each requirement, without defining any reference system. Such methods are the analytic hierarchy process (AHP) (Saaty 1980; Karlsson et al. 1998), numeral assignment (Karlsson 1996) or cumulative voting (CV) (Leffingwell and Widrig 2000; Berander and Jönsson 2006).

In our approach, the countermeasure benefit is finally calculated from the risk estimations. The risk reduction respectively benefit achieved by a countermeasure in relation to a misuse case equals the difference between the two misuse case risks: the reference risk and the varied risk, like in (Arora et al. 2004). The estimation is complicated

by n-m-relationships between misuse cases and countermeasures. If one countermeasure counteracts several misuse cases, then the benefit of the countermeasure can be the sum of the risk reductions produced for each of these misuse cases, but not necessarily, if dependencies exist among the misuse cases. To account for such dependencies, a misuse case group can be estimated together. If vice versa for one misuse case several countermeasures are defined, the estimation of the risk reduction is done for each misuse case–countermeasure pair individually, each time in a system which differs from the reference system by variation of this single countermeasure alone. If the countermeasures depend on each other strongly (either by being able to replace each other partly or totally, or by being effective only if implemented together), the countermeasures are bundled and treated as one countermeasure. Bundling can avoid effects like this: When several countermeasures can replace each other, then the benefit achieved by each of them relative to the perfect system is low (and consequently its priority), although the whole bundle may be highly beneficial. Consequently, these countermeasures would be prioritized too low.

2.2 Empirical Studies of Requirements Prioritization

So far, there have been no quantitative empirical studies of risk-based requirements prioritization. Therefore, in this section, we describe four empirical studies based on other requirements prioritization methods. We built the design of our own experiments on these studies. We could not find other experiments which were as thoroughly designed, executed and analyzed. Other publications on empirical experience in requirements prioritization instead refer to qualitative industry case studies and treat mostly organizational aspects of requirements prioritization, like information flow, stakeholders involved or negotiation of different opinions. However, we are highly interested in systematic and controlled experiments about the estimation process.

Karlsson (1996) performed an empirical comparison of the pair-wise comparison technique and a numeral assignment technique with five participants, applying them on 14 requirements. Criteria for the comparison were time consumption, number of comparisons to execute, standard deviation of the priorities for the same requirement, perceived trustworthiness of the method. They found that relative prioritization by pair-wise comparison of requirements and judging which is more important relative to the other tends to be more accurate and informative than attributing absolute numbers to the requirements. Relative values were also found to be easier to estimate than absolute values.

Karlsson et al. (1998) compared six prioritization methods in a self-experiment. Each of the three authors prioritized the same 13 quality requirements. Their criteria for the comparison were: number of decisions, time consumption total and per decision, ease of use, subjective reliability of results, fault tolerance. They concluded that AHP (Analytical Hierarchy Process) is best, because it produced the most trustworthy results, is fault tolerant, includes a consistency check, and the distance between requirements becomes tangible. Its main problem is scalability: the time consumption grows with the square of the number of requirements.

Karlsson et al. (2004) describe an experiment aimed at comparing the Planning Game PG with Pair-Wise Comparison. They measured the average time consumption and assessed the ease of use by asking: “Which technique did you find easiest to use?”. The accuracy was measured in a post-test a few weeks after the experiment. The subjects were asked which of the two resulting priority orders reflects their opinion best. The experiment was performed with 15 Ph.D. students and one professor as subjects. They prioritized features of mobile phones with respect to both prize and value. The results indicate that PG

is less time-consuming and a majority of the subjects found it easier to use. Most subjects also found the results from PG more accurate, i.e. they said that they reflect their views more accurately, which was unexpected. To find out whether order effects occurred, the two techniques were performed in varied order, the aspects prize and value were treated in different order, and also the requirements were presented in different order. However, no statistically significant order effect was observed.

Karlsson et al. (2007) performed a further experiment for comparing tool supported pair-wise comparison with the PG. They observed the same variables and used the same requirements as above. Half of the subjects were asked to prioritize eight requirements, while the other half prioritized 16 requirements. There were no statistically significant differences between the results for 8 or 16 requirements, for instance no fatigue effect. The order in which the techniques were used affected the mean consistency ratio, but not to a statistically significant degree. When being tool-supported, Pair-Wise Comparison required less time than PG. PG seemed to be less easy to use and its results to be slightly more accurate, but the differences were not statistically significant.

From these experiments, we conclude for our own experiment, that

- a number of 8 to 16 requirements can allow conclusions on the properties of prioritization methods.
- interesting variables for comparing risk estimation methods are: the number of estimations to be done, the time consumption, the standard deviation of priorities for the same requirement, the subjectively perceived ease of use, and the accuracy, i.e. whether the participants think that the resulting priorities reflect their view; fault tolerance does not apply to our method as no redundant estimations are foreseen.
- we rather make the participants estimate relative than absolute values wherever possible.
- we have to discuss or vary the order in which different methods are performed.

3 The Research Questions, and the Requirements Prioritization Methods Used in the Experiments

As there are very few publications on empirical studies about estimations of risk and risk reduction, we performed two student experiments in order to empirically investigate risk estimation and its practical needs: How much time does it take, what knowledge is needed (e.g.: How are estimations influenced by statistics provided?), how much method training and what material do the estimators need, how do group decisions influence the process and results, what is the influence of transparency? (Transparency in the context of our methods means: While estimating misuse case probabilities and damages, the estimator can see their effect on the resulting requirements benefits.) We also investigated the advantages and disadvantages of risk estimation compared to a simpler prioritization method.

These research questions were investigated by evaluating qualitative comments of the participants and observations of the moderators as well as by comparing the quality of the risk estimations under different conditions (e.g. when using different methods, with or without risk statistics provided, with or without transparency, with or without group discussions).

To investigate the advantages and disadvantages of requirements prioritization on the basis of risk estimations, we executed the same task with two prioritization methods: with risk estimation and with a traditional prioritization method. As a reference method to which to **compare risk estimation**, we chose a very simple prioritization method: ranking in two steps. First, each requirement is attributed to one of the groups “high/ average/ low benefit”,

and then a total ranking is performed, attributing the number 1 to the most beneficial requirement and the highest number to the least beneficial one. This ranking is called "Method 1" in this work. We did not choose AHP as reference, although according to the experiments mentioned in Section 2.2, it seems to be the best prioritization method available. However, our purpose was to investigate the benefits which practitioners would experience by performing risk estimation. Assuming that they usually do not choose between one sophisticated method and the other, but rather between a sophisticated method (like risk estimation) and a simpler one, we wanted to simulate this comparison.

Method 2 is based on risk estimations as described in Section 2.1: The **reference risks** (p_{ref} x d_{ref}) and **varied risks** (p_{var} x d_{var}) **of the misuse cases** are estimated, and from these the benefit of each countermeasure is calculated as $p_{var} \cdot d_{var} - p_{ref} \cdot d_{ref}$. The requirement with the highest benefit gets the highest priority 1.

In the Sections 4 and 5, both experiments will be described in more detail, and at the end of Section 5, the parameters varied in the experiments are summarized in Table 4. The influence of these factors is analyzed. We expected effects with respect to the time need, the quality of the resulting estimations and priorities, as well as to the participants' subjective perception of this process and its results. Below, we present the variables used to measure these effects.

The **time need** of the methods is measured by the average duration in minutes the experiment participants need for their execution. As in practice, time need means cost, this variable is relevant for practitioners.

We measure the **quality of the priorities** resulting from the methods with the following variables (all subjective, except for the first one):

a. low **standard deviation** of the priorities of each single requirement (calculated over all participants' or all group results), averaged over all requirements. The standard deviation measures how well the participants agree with each other.
b. the participants indicate in a questionnaire that the method was **easy to use**
c. directly after having done the risk estimations, the participants expect that the resulting priorities will be reasonable, **realistic** and useful (in Experiment 1 without knowing them yet; in Experiment 2 they know the resulting priorities)
d. **accuracy**, i.e. the priorities resulting from the risk estimations reflect the participants' opinion
e. a low **frequency** with which a requirement or misuse case is named by the participants when being asked where they believe that their estimations were especially uncertain
f. directly after the estimations, the participants **feel certain** about their estimations

Additionally to these quantitative results, we also explicitly asked the participants to comment on the methods and the influence of group discussions in free-text fields and we gathered observations made by the experiment moderators.

4 Experiment No. 1

During Experiment 1, nine requirements were prioritized in individual estimations with the methods 1 and 2.

4.1 Experiment No. 1: Preparation

Sample population Ten master students participating in our university course "Knowledge management and decisions in software engineering" in the winter term 2006/07 performed

this experiment in a 3 h session. They had been taught prioritization methods and how to prioritize requirements within MOQARE in the lecture before, 3 weeks ago.

Requirements The experiment uses a case study discussed in the lecture and in preceding homework: an Internet flea market to sell used goods from individuals to individuals. During the lecture, the business goal, business damages, quality deficiencies and quality goals had been identified, during their homework, the students described functional requirements, misuse cases and countermeasures. A consolidated list of all these requirements was discussed in the subsequent lecture. The homework had been done by all participants. Five participants had been participating in the discussion of the case study as well as the homework results, and five missed one or both. This known case study was chosen in order to reduce misunderstandings about the software system and its environment. We expect that the discussions in the lecture and the homework had almost no influence on the experiment results as risks, benefits and priorities had not been discussed there. Nine of the countermeasures defined during the homework were chosen for the experiment. The criteria for this choice were: They tackle a tangible risk which is easy to imagine, and they belong to different quality attributes, i.e. security requirements as well as usability requirements and maintainability.

Each of the methods was tested on the same nine countermeasures of the case study. These were:

- **R1:** clear and intuitive user interface design
- **R2:** user support via several media (phone, email)
- **R3:** similarity with a real life flea market
- **R4:** inspection of the specification documents
- **R5:** encrypted storage of the customer data
- **R6:** fast hardware and software
- **R7:** standard compliance during its implementation and of the user interface design
- **R8:** automated notification of service staff in case of a system breakdown
- **R9:** backup server

(Although some of these countermeasures sound like design elements, we treat them as requirements, because it is unclear whether they will be satisfied by the system; they might not.) These countermeasures are quite fuzzy and not measurable. This in fact was a disadvantage during the estimations and different interpretations were observed to lead to differing estimation results.

In Experiment 1, we define the reference system to be the system in which all countermeasures are implemented. Other reference systems could have been used, but this one is easy to imagine and easier to handle than a reference system in which some countermeasures are implemented and some are not. In this specific reference system, the varied risk is the risk in a system where all countermeasures are implemented, except for one.

Material The material was pre-tested in a self-experiment by the two authors and by a colleague. The pre-tests had led to simplifications of the questionnaire and to an improved presentation of the methods.

During the experiment, the students individually estimated risks on paper questionnaires. They were led by step-by-step instructions and templates. We reduced the number of estimations to be done to the necessary minimum. All calculations and the derivation of benefits and priorities from the risk estimations were done afterwards by us. This was

supposed to reduce the time consumption of the experiment, and also to allow to test the accuracy without transparency, i.e. how well the priorities resulting from risk estimations correspond to the participants' view when (s)he cannot predict which effect her/ his probability and damage estimation(s) have on the resulting countermeasure priorities. The participants worked with the following six documents:

- The introduction which describes the objective of the experiment and the case study, including information about competitors, the company, project execution and staff, and the flea market's functional requirements. This introduction also includes the upper levels of the MOQARE analysis as shown in Fig. 1, which starts with business goals to be achieved. From these, MOQARE derives business damages (i.e. unwanted effects on the business) and quality deficiencies (which describe deficiencies of the software system), from which quality goals are derived. These quality goals describe quality attributes which are to be satisfied by the whole software system or by parts of it. The dependencies between these concepts are visualized in a so called misuse tree. The twelve misuse cases are defined by asking which misuse cases can threaten the satisfaction of these quality goals. For reasons of readability, the misuse tree presentation of the MOQARE results in Fig. 1 does not contain the misuse cases and countermeasures specific to the case study. The misuse cases are presented in Table 2 below. The introduction's content was the result from the lecture and the homework.
- Questionnaire Q1 supports Method 1 by a table which allows attributing a group and a priority to each requirement, see Table 1.
- Questionnaire Q2 supports Method 2. It first states the expected total revenue and cost of the system and informs that all estimations are to be done for a period of 2 years. The reference system is defined. Method 2 is supported by two tables: one for the reference risk (see Table 2) and the other for the varied risk, each containing a column for the probability and for the relative damage estimation. Neither the misuse case risk nor the countermeasure benefit are calculated here. After the risk estimation, the participants are asked to mention those misuse cases where they were especially uncertain (Variable e).

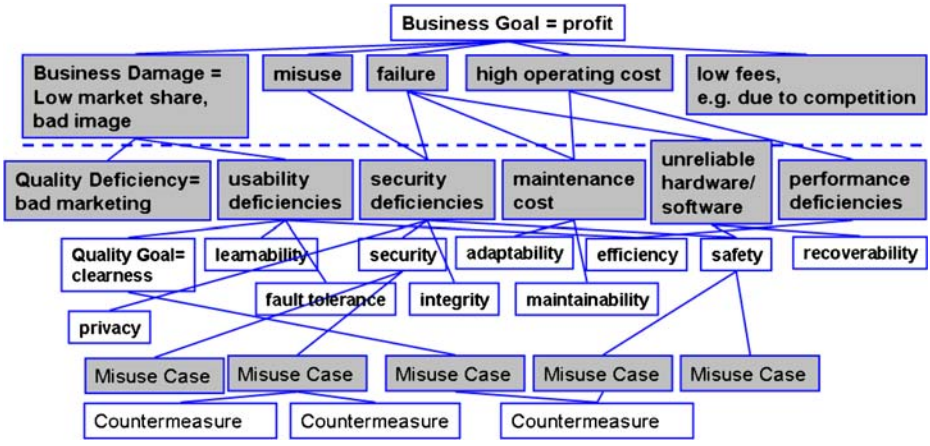

Fig. 1 Upper part of the MOQARE analysis of the Internet Flea Market for Experiment 1, without misuse cases and countermeasures

Table 1 The table on Questionnaire Q1, supporting method 1 in experiment 1

Requirement	Group: "high benefit", "medium benefit", "low benefit"	Priority
R1		
R2		
…		

The table for the varied risk estimation also contains a column for the misuse case probability and damage. Damage is estimated relative to the damage of the reference risk in %. There is one line for each pair of misuse case and related countermeasure.

- Questionnaire Q4 asks the participants to rate the methods in terms of ease of use (Variable b) and whether they expect reasonable, realistic and useful results (Variable c). This is done directly after the estimations.
- Questionnaire Q5: 1 week after the experiment, during the post-test session, each participant receives a table with his/ her priorities resulting from each method. They are asked to comment on the results (e.g. the results of which method reflect their opinion best, Variable d) and on the methods. To test the effect of risk statistics, the participants are offered four statistics on the frequencies of security incidents and the sources of attack from the CSI/FBI Computer Crime and Security Survey (Richardson 2003). On this basis, the reference risk probabilities of two security misuse cases are re-estimated (the participants could look up their former estimations if they wanted to).

These questionnaires—which originally are in German—can be found in our technical report (Herrmann and Paech 2008a, b)

Table 2 Table on questionnaire Q2 for reference risk estimation in method 2, in experiment 1

Misuse case	Probability p of the misuse case in %	Damage, relative to the benefit of the business goal (550,000€), in %
MUC1: user error impedes planned purchase		
MUC2: neglect of intuitiveness requirements during software development leads to loss of customers		
MUC3: seller gives up trying to offer an item after some time		
MUC4: users without technical background do not understand technical terms and user interface -> long learning phase and loss of customers		
MUC5: web site does not help users when they enter faulty input		
MUC6: customer data are read by unauthorized person		
MUC7: hacker manipulate the web site including the data		
MUC8: inefficiency by long waiting times caused by the system		
MUC9: complexity of the system leads to high maintenance cost		
MUC10: low user efficiency because information is hard to find and essential information can only badly be recognized		
MUC11: code can only be reused poorly or not at all when changes in the system environment occur		
MUC12: The only server fails, and after hours the service staff notices the failure by chance; a backup server is not available; the users cannot access the web site for days		

4.2 Experiment No. 1: Execution

The methods 1 and 2 were performed in this order. We did not vary the order, because we expected that performing the more sophisticated method first and then the more primitive one would influence the results of the latter, i.e. that for the simple ranking in Method 1, risks would be taken into account to a higher degree then if Method 1 was applied first. In the first experiment, we wanted to avoid such an effect. (Such order effects were tested in the second experiment.)

Before the first questionnaire was distributed, there was an introductory presentation which explained the objective of the experiment and the time plan, as well as the case example. After Questionnaire Q1 and before Q2, the principles of requirements prioritization by risk estimation and of the reference system were recapitulated.

The experiment was performed in a 3 h session, with 10 participants. Concerning the risk estimations in Q2, a discussion arose about how to interpret percentages in the probability estimations. For instance, for some misuse cases the probability means "What ratio of the users…?", for others: "How long during 2 years…?" Some information was found to be missing in the material, like: When a potential buyer does not buy an article, how high is the probability to find another buyer?

In two cases, in Q2, there evidently had been misunderstandings during the risk estimations: When estimating the damage for the varied risk, a value relative to the reference risk damage was to be given, not relative to the value of the business goal, as in the estimation of the reference risk before. This misunderstanding had the consequence that the risk with the countermeasure being implemented was higher than without. These two participants received their results in the form of excel spread sheets per email and corrected their varied risk damage estimations before the post-test session.

One week after the experiment session, a post-test was performed where Questionnaire Q5 was answered by the participants individually and afterwards a concluding discussion of the experiment took place in the group.

5 Experiment No. 2

During Experiment 2, seven countermeasures were prioritized in moderated group estimations with methods 1 and 2.

5.1 Experiment No. 2: Preparation

Sample population Twenty-four students took part in this experiment in the summer term 2007. They were divided into seven groups. Three groups had four members and four groups had three members. These groups were identical to the teams which had formed 5 weeks earlier and worked on a programming project together in the software engineering course. In these teams, almost all members were bachelor students of the "Software Engineering I" course. Except for one 3-member-team, all groups included a project manager, i.e. a master student (a bachelor student in one case) who took part in the course "Software engineering II: Requirements Engineering and Project Management". One project manager did not take part in the experiment, because this student had before been a participant in Experiment 1 and we therefore expected a strong influence of his risk estimation experience in the discussion. The moderators were advised to treat all team members equally.

The students had not been taught prioritization methods or how to prioritize countermeasures in MOQARE before, but received a 20 min introduction at the beginning of the experiment.

The experiment was performed in sessions of one and a half hours.

In this second experiment, some unwanted factors from the former experiment were eliminated. These factors are discussed in Section 5.2. To observe their effect was one of the objectives of this Experiment 2.

Requirements The software we used as an example, was Sysiphus (Dutoit and Paech 2002; Dutoit and Paech 2003; Wolf and Dutoit 2004; Sysiphus 2007), a software engineering tool which is being developed at the Technical University of Munich and at the University of Heidelberg. The tool Sysiphus is used to teach software engineering and to document the results of software projects. It serves two purposes: It supports the whole software engineering process in student projects and it serves as a realistic example software which students modify to learn programming. Five weeks before the experiment, as a homework, the students had derived misuse cases and countermeasures for the quality goals "conformance of user interface to user expectations", "performance", "availability", "compliance to Java Code Conventions" and "clear code structure". Some of the misuse cases and countermeasures were then used in the experiment. The students consequently knew how to elicit and to specify countermeasures with MOQARE and had thought about misuse cases, but had not yet learned how to prioritize countermeasures or thought about it.

The countermeasures prioritized in this experiment were:

- **R1:** usability tests with future users
- **R2:** high-performance system (more main memory & faster processors & more efficient algorithms)
- **R3:** limiting the number of simultaneously allowed users to one fourth of the number of students
- **R4:** monitoring and automatic restart of the server
- **R5:** maintenance activities are done exclusively in the mornings between 7 and 9 am
- **R6:** user errors are caught and do not lead to system failure
- **R7:** backup every 3 days

We paid attention that these countermeasures were less fuzzy than those in Experiment 1, and some are even measurable.

In Experiment 2, we dealt with an existing system which is known to and used by the participants. Here, it was easiest to define the status quo to be the reference system. In such a reference system, estimations of the varied risk are to be performed differently for countermeasures that are implemented in the reference system and for those which are not. For those countermeasures not realized in the reference system, the varied risk is estimated by estimating the misuse case risk in a system which differs from the reference system by this one countermeasure being implemented additionally. The difference between these two risks for each misuse case is the benefit achieved by each countermeasure with respect to this misuse case.

Material In this experiment, two methods were applied: Method 1 and Method 2. During the experiment, the moderator entered the values estimated by the group into spreadsheets which were projected to the wall, so that they were visible for all participants. These spreadsheets calculated risks and benefits automatically.

The table supporting Method 1 this time also presented the misuse case to which each requirement/ countermeasure refers to, see Table 3.

The tables for Method 2 are like those in Experiment 1 (see Table 2), but have an additional column on the right side, where probability and damage are multiplied automatically, to result in risk.

Additionally to performing the estimations in the group, the participants individually completed three questionnaires:

- Questionnaire Q1 evaluates Method 1 (with respect to the Variables c and e and in free-text comments) and was filled out immediately after the execution of Method 1.
- Questionnaire Q2 evaluates Method 2 (with respect to Variables c and e and in comments) and was filled out immediately after the execution of Method 2.
- Questionnaire Q3 compared Method 1 with Method 2 and was filled out after the execution of both methods and after Q1 and Q2. It asked questions about the Variables b and d and requested comments on the effect of group discussions and the (dis) advantages of quantitative risk estimation.

This material can be found in our technical report (Herrmann and Paech 2008a, b).

Great care was given to the wording of the misuse cases and countermeasures, the design and content of the experiment material and the instruction of the four moderators. The task of the moderators was to organize the group discussions, to type the estimations into the spreadsheets, to control the time and to avoid misunderstandings concerning the misuse cases, countermeasures, the definitions of probability and damage and the use of the method. The moderators were not allowed to propose any values. They guided the process by asking questions.

There were two preparatory sessions with the moderators: in the first, the concept and material of the experiment were discussed, in the second, a test run was executed where the future moderators were estimators and the author of the method was the moderator.

Table 3 Table supporting method 1 in experiment 2

Requirement	Group: "high benefit", "medium benefit", "low benefit"	Priority	Misuse case
R1: usability tests with future users			MUC1: use is expensive due to confusing user interface
R2: high-performance system (more main memory & faster processors & more efficient algorithms)			MUC2: high load before delivery deadline turns it difficult to deliver homework in time
R3: limiting the number of simultaneous users to one fourth of the number of students			MUC2: high load before delivery deadline turns it difficult to deliver homework in time
R4: monitoring and automatic restart of the server			MUC3: failure due to instable system
R5: maintenance activities are done exclusively in the mornings between 7 and 9 am			MUC4: frequent maintenance activities lead to unavailability of Sysiphus
R6: user errors are identified and do not lead to system failure			MUC5: user errors lead to system crash
R7: backup every 3 days			MUC6: data loss due to system crash (software or hardware failure)

5.2 Influencing Factors in Experiment 1 and 2

In both experiments, Method 1 and Method 2 were executed. In the first experiment, 9 countermeasures and 12 misuse cases were treated, in the second experiment there were 7 countermeasures and 6 misuse cases. In addition to the number of countermeasures and misuse cases, the factors presented in Table 4 had been varied between the experiments. This was done taking into account what we had learned from the first experiment, with the objective to improve the risk estimation in the second experiment. To observe the effects of these variations was one of the objectives of Experiment 2. These factors are discussed in what follows.

System: In Experiment 2, the students estimated risks for a system which they knew well as they had been using it in their course for at least 4 weeks and were enhancing its code. Consequently, their estimations were based on user knowledge and programmer knowledge of a real system. Furthermore, during the experiment no detailed description of the system and its environment needed to be provided by us and to be understood by the participants, as was necessary in Experiment 1, where a fictitious system was used. The countermeasures were less fuzzy in Experiment 2 than they had been in Experiment 1. They were quantified where possible.

Group discussions: In Experiment 1, each participant did estimations individually on a paper questionnaire. In Experiment 2, groups of three or four persons discussed jointly until they agreed on one value. We expected that these discussions would reduce the probability of misunderstandings.

Transparency: In Experiment 1, with Method 2 the participants estimated probabilities and damages of misuse cases, but did not know which misuse cases risks, countermeasure benefits and countermeasure priorities would result from these estimations. In Experiment 2, the estimated probabilities and damages were put into a spreadsheet which automatically calculated these values immediately and which were visible to all group members. It was technically possible to test different values and to check their influence on the result, but was not done often due to time limits.

Prioritization criteria: For Method 1, in Experiment 1 the participants were asked to rank the countermeasures according to their benefit. When asked which criteria they had used, a large variety of criteria was named. Not two participants shared the same criteria, like competitiveness, the cost of changing the implementation, specific needs during the initial

Table 4 Influencing factors varied in the experiments

	Experiment 1		Experiment 2	
Factor	Method 1	Method 2	Method 1	Method 2
Number of participants	10	10	24	24
Number of resulting data sets	10	10	7	7
Number of requirements	9	9	7	7
Example system	Fictitious	Fictitious	Real	Real
Group discussion & moderation	No	No	Yes	Yes
Transparency	Yes	No	Yes	Yes
Order of execution	1st	2nd	1st or 2nd	2nd or 1st
Prioritization criteria	Benefit	Relative damages in %, probabilities in %	Benefit with respect to misuse case	damages in lost calendar hours, probabilities in times per month

phase of the online flea market, the cost of non-implementation, or the quality of service. (Note: Evidently, the estimators in Method 1 already considered risk, although they had been asked to use benefit as prioritization criterion. They had been influenced by the experiment's introductory material which announced that risk would be used for requirements prioritization.) As the prioritization criteria were found to be misunderstood and diverse among the participants, in Experiment 2, we defined more clearly that the countermeasures had to be ranked according to their benefit with respect to a misuse case (without estimating the risk quantitatively). The requirements were less fuzzy in Experiment 2 than they had been in Experiment 1. They were quantified where possible. The measures with which probabilities and damages were estimated, were defined in a more tangible way. For Method 2, in the first experiment we preferred estimating damages in relative values in %, because relative values had been found to be easier to estimate in former experiments by other authors. In the second experiment, we measured damages in lost calendar hours. This was more tangible and could be deduced from the participants' everyday user experience with the system. Probabilities were estimated in % in Experiment 1, whereas in Experiment 2, they were measured by the average number of times per month. This metric was also more tangible und less ambiguous than giving a probability in percent. A percentage can mean something different with respect to each misuse case.

5.3 Experiment No. 2: Execution

At the beginning of the estimation workshops, the students got an introductory presentation of 20 min about requirements prioritization in general and the two methods, to risk estimation and the significance of the reference system. The agenda of the experiment and the meaning of the misuse cases and countermeasures were explained. It was also defined that estimation of probabilities refers to "number per month" in a normal month during lecture time and that damages are estimated in calendar hours lost in a student project (not working hours: for instance, if Sysiphus breaks down at midnight and is re-started at 9 o'clock the next morning, then 9 h are lost in which no one could work, although the work time lost may only be 20 min in all). Consequently, misuse case risk quantifies the calendar hours which are lost per month due to a misuse case. Immediately before executing Method 2, another five-minute introduction to requirements prioritization was given to each group. Based on our experience from the test run, the agenda allowed 20 min for Method 1 and 40 min for Method 2.

This time, the order in which the two methods were executed, was varied. The first four groups started with Method 1 and then executed Method 2, the other three groups proceeded in the reversed order.

Questionnaires 1–3 were used to ask the participants about their opinion about the methods. Immediately after each method, a questionnaire asked to evaluate the method, and at the end of the experiment, Questionnaire Q3 requested to compare the two methods.

After the experiment, the moderators were asked about the process of the group discussions and these observations were evaluated qualitatively.

6 Results and Lessons Learned

In this section, the results of Experiment 1 and 2 are analyzed together. These results include quantitative data (time need and Variables a–f, summarized in the annex), the results of the data

analysis and lessons learned as concluded from the free-text answers in the questionnaires, from discussions with the participants and from observations by the moderators. We here analyze how the factors given in Table 4 influence the variables defined in Section 3 and the qualitative results of the experiments. Where averages from different samples are compared, we tested the statistical significance of the difference by testing the hypothesis that the expectation values of both samples are equal. If this hypothesis can be accepted with a certainty of 90% or more, we assume that the difference is not statistically significant.

6.1 Influence of Statistics

We expected the following influence of providing **statistics** about frequencies and damages of specific misuse cases. When the participants are given several statistics, we expect that

- this significantly influences the value of the estimated probabilities of misuse cases,
- the standard deviation (relative to the average) of their estimations is lower (Variable a),
- they feel more confident about their estimations.

In Experiment 1, the participants were provided four publicly available statistics about security incident frequencies and were asked to re-estimate the probabilities of two security misuse cases. The results are presented in Annex A.a. As can be seen from these results, providing statistics to the estimators did (statistically significantly) influence the value of their probability estimations and also lowered their relative standard deviation (Variable a). However the participants were not sure whether the statistics really facilitated their estimations. Half of the participants in their free-text comments were still sceptical whether the estimated values based on statistics are more exact than those estimated without. Similar doubts had been uttered by Xie et al. (2004) (see Section 2.1).

6.2 Improvements from Experiment 1 to Experiment 2

From our experience in Experiment 1 we learned about the challenges and needs of risk estimation and therefore varied some influencing factors in order to improve the results. These factors are discussed in Section 5.2. For example, that instead of a fictitious system, a real system known to the estimators was chosen (because the estimators felt very uncertain about their risk estimations), and estimations were not given individually but in moderated group discussions (to avoid misunderstandings about the method and to bring different experiences and knowledge into the estimation), also the prioritization criteria were defined more clearly and tangibly, requirements were less fuzzy and quantified where possible, and transparency was now introduced into the risk estimation by tool support.

All these changes were expected to improve the results of the experiment with respect to at least some of the criteria defined above, which in fact they did. In this section we present these improvements. In the following sections, we discuss which of the improvements is due to which factor. As so far we have performed only two experiments, the effects can not easily be attributed to one factor and most of them very likely have more than one cause. However, when for instance one improvement is more pronounced for Method 2 than for Method 1, this is a hint that transparency plays a major role here because transparency was improved for Method 2, but was the same for Method 1 in both experiments.

We observe that **time need** was much higher in Experimental 2 than in Experimental 1 (see Tables 5 and 6), what we mainly attribute to the fact that in Experiment 2, estimations were discussed in the groups-as will be discussed below. The standard deviations of the priorities for the same countermeasure (Variable a, see Table 7 and Table 8) were compared.

Table 5 Time need in minutes per countermeasure

	Experiment 1	Experiment 2
Method 1	0.73	2.50
Method 2	3.53	5.33

In Table 9, the standard deviations are normalized, i.e. divided by the average priority, which is equal to $(n+1)/2$ when n is the number of countermeasures. The normalized standard deviations—also called coefficient of variation—of the priorities have indeed decreased. This means that some sources of uncertainty and different perception which were probably caused by the setup of the experiment could be reduced. The difference is large: For Method 1, this reduction was 25% and for Method 2, it was 17%. We must remark that the countermeasures and the participants were not the same in the two experiments.

In terms of **ease of use** (Variable b, see Table 10), both methods score better in the second experiment. With Method 2, this effect is more pronounced. (However, the differences between both experiments are statistically not significant, only with a certainty of 50%.) The perceived higher ease of use—if it really exists—can be assumed to be a combined effect of several of the factors varied between the experiments.

The participants in the second experiment believed that their results were more **realistic** (Variable c, see Table 11). This is true with both methods, but the effect is more pronounced with Method 2. When we compare Tables 9, 10, 11, it is interesting to remark that the coefficient of variation is lower, when more realistic results were expected by the participants. This indicates that the participants are able to judge which values were uncertain to estimate and where the other participants might obtain similar results. (However, these differences are not statistically significant due to high standard deviations of these variables among participants.)

Table 12 presents the **frequency** with which the participants named a countermeasure or misuse case as being especially **uncertain** (Variable e, see Annex A.e). These results indicate that there were statistically significantly fewer uncertainties in Experiment 2 than in Experiment 1 (Table 13 and Table 14).

(Remark: Overall, a countermeasure was named with a frequency of 0.3 in average. For the three countermeasures R3, R5 and R7, which are defined quantitatively, this average is only 0.15. It seems that the metric made the countermeasure benefit easier to judge, but this question should be re-investigated with more than seven countermeasures.)

Variable d was not comparable as it was determined qualitatively in Experiment 1 and quantitatively in Experiment 2.

Statistically relevant improvements in Experiment 2 compared to Experiment 1 were found with respect to the coefficient of variation, and less misuse cases and countermeasures were named as being difficult to estimate. Statistically not relevant improvements were found with respect to their ease of use and the participants' expectation of how realistic the results are.

Table 6 Time need in minutes per estimation

	Experiment 1	Experiment 2
Method 1	0.37	1.25
Method 2	0.61	1.43

Table 7 Variable a: The standard deviations s found among the priorities of the ten participants in Experiment 1 for each single countermeasure: $\bar{s}$ denotes the average of s over all participants, s_{min} the minimum value found for any countermeasure, and s_{max} the maximum

	$\bar{s}$	s_{min}	s_{max}
Method 1	2.35	2.01	2.84
Method 2	2.38	2.04	2.88

6.3 Influence of Moderated Group Discussion

The group discussions compared to individual estimations were expected to not only demand more time, but also to improve the quality of the results and the participants' perception of the process. The results of a group should be better than those of each individual. We expect this because during the discussion the knowledge of several persons is added, misunderstandings with respect to the method or the case are discovered, missing information investigated or common assumptions are made. In addition to comparing Variables a–f in both experiments, we ask the participants explicitly how they perceive the discussions.

One very strong influence was observed on **time need**. We expected the time need (see Table 6) to be proportional to the number of estimations to perform (which differs a lot between the methods) and to depend on whether the estimations were done alone or discussed in a group. In Method 1, the estimation of one value with 3-4 participants took 3.4 times as much time as individual estimations. In Method 2, this factor was 2.3. (All these ratios are statistically significant.) While the group discussions took more time than individual estimations, the group discussions presumably contributed to the improvements discussed in Section 6.2. Whether and to which degree this is true, should be investigated in additional experiments.

We wondered whether all participants perceived the same countermeasures or misuse cases as **difficult to estimate** (Variable e, see Annex A.e). And whether in Experiment 2 the members of a group share this perception. Such effects might have been caused by discussions in the groups having focused on the same misuse case for a long time. Some single countermeasures or misuse cases were especially certain or uncertain for many participants, but no evident correlation was visible among the opinions of the members of the same group.

The other variables observed allow no conclusions specific to group discussion. Yet the estimators were asked some qualitative questions with respect to the group discussion and the replies are discussed in the following.

In Experiment 2, we asked: Do you think that you have been **involved adequately** in the discussions and that your proposals and objections have been considered sufficiently? Concerning Method 1, an average of 1.58 points was attained (from an interval of [−2, +2]) and concerning Method 2 it was 1.25 points. This difference is statistically significant. This

Table 8 Variable a: standard deviation of the priorities of the seven groups in Experiment 2 (calculated for each countermeasure, then averaged over all countermeasures) (* was 0 for Countermeasure R7)

	$\bar{s}$	s_{min}	s_{max}
Method 1	1.40528	1.25357*	1.951800
Method 2	1.58414	0.78680 (for R7), 1.0965 without R7	2.64575

Table 9 Variable a: coefficient of variation (=standard deviation/average = $\overline{s}/\overline{p}$) of the priorities p

	Experiment 1	Experiment 2
Method 1	0.470	0.351
Method 2	0.476	0.396

means that with Method 1, it was easier to make the participants feel involved in the discussion.

In Questionnaire Q3, we asked the participants explicitly how they **perceived the group discussions**. We were interested in the qualitative content of their replies, but also whether more advantages or disadvantages were named. The participants named more advantages than disadvantages (20 advantages versus 6 disadvantages). As advantages they repeatedly named: discussion led to a better understanding of the meaning of countermeasures, misuse cases and the method, clarification of misunderstandings, taking into account different experience (by calculating averages, by eliminating exceptional experiences, by discussing new aspects and solutions). Disadvantages were: individual opinions are neglected, different roles of the participants influence the group consensus (here: project managers dominated), higher time need.

In **free-text comments**, they expected that estimations would have been difficult when done alone, because knowledge exchange was useful (named twice), one participant emphasized that moderation was important, one felt that estimations would have been easier alone because no discussion would have taken place, one would have needed a clearer specification of the countermeasures, and three participants explicitly expected no difference.

We also asked quantitatively how difficult the methods would presumably have been, if executed alone. We compared the answers to the ease of use judgement made before. The participants expected that doing the estimations alone would have been somewhat more difficult or less easy than during the experiment. Method 1 received 1.00 instead of 1.13 points, Method 2 got −1.13 instead of −0.92 points. The difference is statistically significant for Method 1, but not for Method 2, because of a higher variance of the data.

The moderators observed that the student who had the **role** of project manager within the team in almost all the groups organized the finding of the group consensus. This was probably not only due to the project manager role, but also because the project managers had more experience with the example software. They had started using Sysiphus several weeks before the other students and most of the project managers also knew the software from earlier courses.

The effect of moderated group discussions compared to individual estimations was difficult to isolate quantitatively in these two experiments. The participants' answers indicate that they perceive the group discussion as positive. However, the group discussions demanded more time and the roles that the persons have within the group influence the discussion.

Table 10 Variable b: The ease of use as assessed by the participants. Given are the average values in points, averaged over all participants

	Experiment 1	Experiment 2
Method 1	1.0	1.12 Points
Method 2	−1.2	−0.92 Points

Table 11 Variable c: Do the participants expect the priorities to be realistic? Given are the average values, averaged over all participants, in points

	Experiment 1	Experiment 2
Method 1	1.00	1.04
Method 2	0.10	0.17

6.4 Influence of Transparency

Transparency here means that when doing the probability and damage estimations, the participants can see (and control) their influence on the resulting countermeasures priorities. In Experiment 1, Method 2 was not transparent, while in Experiment 2 we performed it in a transparent way. In Experiment 2, the estimated probabilities and damages were put in by the moderator in a spreadsheet which automatically calculated these values immediately and which were visible to all group members. It was technically possible to test different values and to check their influence on the result, but this was not done often due to time limits. Method 1 was transparent in both experiments. We expected that such a transparency leads to corrections of the estimations during a plausibility check of the priorities. We also expected the participants to perceive the quality of the resulting estimations and priorities as better and to feel a higher trust in the method.

In Experiment 1, we tested whether "blind" estimations of probability and damage which are done without knowing their effect on the resulting priorities lead to good results. However, in Q5, 9 out of 10 participants marked "because the estimations were split up in single steps and the result of one's estimation is not predictable" as a reason why his/ her results were so different with different methods. In the subsequent Experiment 2, we hoped that transparency would reduce problems like the inconsistency problems observed in Experiment 1, where 6 out of the 10 participants obtained at least one negative countermeasure benefit. Such a negative value signifies that the implementation of a countermeasure did not lead to a risk reduction but to a risk augmentation. In fact, no such errors were observed in Experiment 2. We suppose that this is caused by the fact that they could easily be detected by the estimators.

Transparency, as was expected, was observed to lead to corrections of the estimations during Experiment 2. The participants could compare the resulting risks to those of other misuse cases and checked for plausibility. Due to time constraints in the experiment, this was done in only few cases, but sometimes the probability or damage estimations were corrected.

As has been discussed above, there have been several improvements observed in Experiment 2, compared to Experiment 1. If this effect is more pronounced for Method 2 than for Method 1, this can be a hint that the higher transparency causes part of this

Table 12 Variable e: average frequency with which a certain countermeasure or misuse case was named as being difficult to estimate per participant (* marks the results we obtain when the six participants who said that they were uncertain for all misuse cases are taken literally)

	Experiment 1*	Experiment 1	Experiment 2
Method 1		0.27	0.20
Method 2, probability estimation	0.73	0.24	0.28
Method 2, damage estimation	0.67	0.20	0.31

Table 13 Probability estimations for misuse case 6 ("Customer data are read by an unauthorized person")

	Without statistics (Q2)	With statistics (Q5d)
Average over all participants	7.64%	45.3%
Standard deviation	17.2%	20.3%
Coefficient of variation = standard dev./ average	2.25	0.45

improvement, because for Method 1, the transparency was the same during both experiments. The improvement in fact was higher for the **ease of use** (Variable b, see Table 10: For Method 1, the improvement was 0.12 points, for Method 2 it was 0.28.) and the results are expected to be more **realistic** (Variable c, see Table 11, where Method 1 gets a plus of 0.04 more points, but Method 2 of 0.07). The statistical significance of these differences for Variables b and c however is low (50% certainty).

However, transparency seems to have no major effect on the **standard deviations** (Variable a). The improvement was more pronounced in Method 1 than in Method 2.

6.5 Prioritization Criteria and Metrics

In Experiment 1 for Method 1 the results of different persons differ widely from each other. One reason for this might be that Method 1 here did not define clear prioritization criteria, but asked the participants to rank the countermeasures concerning their benefit. When being asked about their criteria, each of the participants named different criteria, usually quality attributes like security or usability, but also user benefit or administrator benefit, risk or to surpass competitors. We hoped that using clearly defined prioritization criteria in Method 1 would lower the standard deviation of the results. We therefore defined in Experiment 2 that the countermeasures were to be ranked according to their benefit relative to a specified misuse case. In Method 2 we had clear criteria (probability and damage) and nevertheless the results of the participants differed a lot.

In Experiment 2, more tangible measures were used (explained in Section 5.2). We hoped to reduce the influence of misunderstandings and consequently the standard deviation of the resulting priorities. Additionally, in Experiment 2 we took care to define countermeasures that are less fuzzy than in Experiment 1 and which are quantified where possible. This was expected to reduce the standard deviation, as this fuzzyness according to the participant comments was one source of uncertainty during estimations.

Indeed a reduction of the **standard deviations** in Experiment 2 was observed for methods 1 and 2. For Method 1, this effect was stronger than for Method 2.

Table 12 presents the **frequency** with which the participants named a countermeasure or misuse case as being especially **uncertain** (Variable e, see Annex A.e). In Experiment 1, damage estimation has been statistically significantly more difficult (in Method 2) than

Table 14 Probability estimations for misuse case 7 ("Hackers manipulate the flea market including its content")

	Without statistics (Q2)	With statistics (Q5d)
Average over all participants	15.0%	33.1%
Standard deviation	21.8%	21.7%
Coefficient of variation = standard dev./ average	1.45	0.66

probability estimations, whereas in Experiment 2, no statistically significant difference is found. This can be an effect provoked by the different metrics used in the two experiments.

The prioritization criteria, damage metrics and probability measures have to be chosen carefully. There were hints that the risks referring to measurable countermeasures are easier to estimate.

6.6 Order Effects and Learning Effects

In Experiment 2 we tested order effects: Four groups (number 1–4; 13 participants) executed Method 1 first and then Method 2, the other three groups (number 5–7; 11 participants) in a second shift (supported by the same moderators) proceeded vice versa. There were differences observed between the results of group 1–4 and 5–7.

The method which was executed second in order **took less time**. From the data in Experiment 2, we estimate that about 12% of the total time is needed for general explanations and clarifications.

Differences were also found with respect to the Variables c, d, and e:

Variable c: The participants of groups 5–7 expected their estimations to be more **realistic** than groups 1–4 did: Concerning Method 1, groups 1–4 gave an average of 0.85 points versus 1.27 given by group 5–7. With respect to Method 2, the probability and damage estimations were expected to be realistic with −0.23 versus 0.55 points, and the resulting priorities with −0.08 versus 0.45. This means that groups 5–7 considered their results to be more realistic in general, for both methods. All these differences have been found to be statistically significant.

The findings are similar for accuracy (Variable d): Method 1 received 1.23 versus 1.36 points, and Method 2 received 0.00 versus 0.27. For Method 1, the difference is statistically not significant, but for Method 2 it is.

When being asked for misuse cases which were especially uncertain with respect to their probability estimation, groups 1–4 named each misuse case (Variable e) with a frequency of 0.23 and groups 5–7 only with 0.18. With respect to damage estimation, these frequencies were 0.19 (groups 1–4) versus 0.27 (groups 5–7). With respect to Method 1, the frequency in groups 1–4 was higher: 0.20 versus 0.09. These differences are statistically significant. Each participant of groups 1–4 named at least one countermeasure here, but in groups 5–7 only 5 out of 11 participants did.

In summary, groups 5–7 seem to have felt more confident about their results (but not about the ease of use of the method and about damage estimation) than groups 1–4, even for Method 2 which they executed first. In this experiment setting this probably signifies a learning effect of the moderators. In Experiment 2, we proceeded in two shifts for practical reasons: Group 1–4 started and immediately afterwards, the same moderators executed the same experiment with group 5–7. It is possible that during their second run, the moderators could answer to questions better, explain the method better and in general felt more confident, which influenced the participants' perception.

6.7 Comparison of the Methods

In both experiments, methods 1 and 2 were compared with each other with respect to time need and the Variables a–e. We also cite some remarks of the participants about the differences which they experienced between the methods.

Time need: As expected, Method 1 was fastest. This is not only because there were less estimations to do per countermeasure, but also the time need per estimation was lower. In

the prioritizing of the same (number of) countermeasures, in Experiment 1, the relation between Method 1 and 2 concerning the total time need was 1: 4.8 and in Experiment 2, it was 1: 2.1 (see Table 5). The time need per estimation in Experiment 1 was 1: 1.6 and in Experiment 2, it was 1: 1.1 (see Table 6.). This shows that the higher time need mainly results from a higher number of estimations to do and to a smaller part (but statistically significantly) from the fact that damage and probability estimations are more time-consuming than the decisions in Method 1.

Risk estimation takes more time than ranking. The next question was whether the higher effort of the risk estimation is remunerated by a better quality of the resulting countermeasures priorities.

The **standard deviation** (Variable a), see Table 7 and Table 8: The differences between the standard deviations of the methods during the same experiment are not statistically significant. However, Method 1 seems to be slightly better than Method 2. One might wonder whether a higher consistency of the priorities means that they are more realistic. We do believe that the risk estimation in principle predefines the estimators' perspective by the misuse case definition. This can be an advantage, if the misuse cases are properly defined, but can also be a disadvantage, because other perspectives, which are not considered by these misuse cases, might be underrepresented by the countermeasure priorities.

Ease of use (Variable b) See Table 10: In both experiments, Method 1 was judged to be (statistically significantly) easier to use than Method 2. Method 1 was found rather easy to use, Method 2 rather difficult.

Realistic priorities (Variable c) See Table 11: The participants expected Method 1 to deliver the most realistic priorities ("rather realistic"). The differences between Method 1 and 2 are statistically significant.

Accuracy (Variable d) See Annex A.d: We summarize the qualitative answers obtained from Experiment 1 by saying that intuitively the participants felt that Method 1 reflected their subjective priorities best, while they thought that Method 2 must deliver more objective and therefore better results. Method 2 was not clearly perceived as being more accurate. In Experiment 2 (Questionnaire Q3), Method 1 again rated better than Method 2.

In Table 12, we see that in both experiments, the risk estimations in Method 2 led to a (statistically significant) higher number of estimations where the participants felt especially uncertain (Variable e), compared to Method 1.

In Experiment 2, the participants were asked which advantages and disadvantages the quantitative estimation in Method 2 had, compared to the intuitive ranking in Method 1 (free-text question). Among 19 free-text replies, 13 mentioned advantages and 18 disadvantages. We conclude that the quantitative estimation was experienced as being rather inferior. The participants named as advantages: objective measure, own experiences can be contributed, schema, order of magnitude, more details. As disadvantages they saw: difficult estimation, especially when information is missing, high time need for the same result, dependency on many factors, uncertainty of the estimated values, coming to a consensus is more difficult. The latter can be explained by the fact that in Method 1, there are only few values to choose from, compared to Method 2.

Which method is best? Method 1 (ranking) rated better than Method 2 (risk-based prioritization) with respect to time need and the quality Variables b–e, and according to free-text answers. Only with respect to Variable a this was not the case. We will discuss this finding in the conclusion in Section 7.

In Experiment 1, we also tested another risk estimation method based on MOQARE's Misuse Tree (Herrmann and Paech 2005; Herrmann et al. 2006; Herrmann and Paech 2008a, b). In this Method 3, the probability estimation of a misuse case risk is equal as in Method 2, but the damage is derived step-wise from business goals via the intermediate concepts business damage, quality deficiency, quality goal, top-down to the misuse case. Method 3 is not presented and discussed in detail here, as it was abandoned. Compared to Method 2, it demanded many more estimations and consequently more time. The step-by-step estimations of Method 3 were considered to be an advantage by several participants, because they ask simpler questions which can more easily be answered. This was reflected by the fact that Method 3 needed less time per estimation than Method 2. The main argument of abandoning Method 3 against Method 2 was that each of the cascaded estimations must be expected to be flawed with an uncertainty, which raises the total uncertainty of the resulting countermeasure benefit to about eight times the uncertainty of each estimation. In terms of the Variables a–f, Method 3 rated approximately equally to Method 2. Sometimes, it seemed to be slightly better, but the difference was not statistically significant.

6.8 Further Lessons Learned

Additionally to the results discussed above, we gathered lessons learned from free-text feedback of the participants in the questionnaires and in discussions, as well as from the observations made by the discussion moderators.

Risk estimation turned out to be difficult, as was expected. The resulting priorities of the participants in all methods differed a lot. The participants themselves (in Experiment 1, Q5) proposed the following reasons for these deviations (the first three were named by more than one person):

- Different criteria and goals of different estimators
- Different experiences
- Uncertainty of the estimations
- The end result (the priorities) was difficult to foresee for the participants
- Missing information led to differing assumptions
- Missing experience and technical competence
- Misunderstandings concerning the method
- Missing knowledge about market and reality
- No feedback about the other participants' estimations, which might have led to more realistic values
- Time pressure

To estimate risks reliably and to feel certain about one's estimations, one needs a lot of information about the system, the usage, the environment. According to participant answers in the questionnaires and according to our observations during both experiments, one need the following types of information:

- To estimate the reference risk, personal experience with the system is useful.
- The varied risk is estimated on the basis of "What-if" questions. For these estimations, practical experience is required with countermeasures, which have not been implemented so far, in order to come to realistic expectations about their effect. It has to be clearly defined what the system would be like, if a countermeasure was not implemented or implemented additionally.

- Expert knowledge is needed, e.g. management knowledge (about business goals), user knowledge (e.g., from the user perspective about frequent misuse cases and damages caused), and technical knowledge (e.g., from the technical perspective about frequent misuse cases and damages caused, as well as technical possibilities to mitigate misuse cases). An example of where the lack of knowledge caused difficulties, is: In Experiment 2, the participants from their user experience knew how often a user observes a system failure. However, each of three reasons given referred to another misuse case (because each demands specific countermeasures), and the participants could only guess the relative occurrence probabilities of these misuse cases. To have such expert knowledge available is difficult even if the system exists, is well known to the estimators and is regularly used.
- For treating n–m-relationships between countermeasures and misuse cases as described in Section 2.1, dependencies among risks and countermeasures must be known.

In Q5 of Experiment 1, 3 out of 10 participants marked "missing information" as a reason why his/ her results from different methods were so different and 4 marked "misunderstandings". Information which the participants regarded as missing for instance was: the number of system users, the cost for setting up a support centre, the protection of the company's servers by measures other than encryption, the qualification of the personnel, the knowledge of the users. Xie et al. (2004) had also found that risk is highly project and company specific. We wonder whether it would have been practically possible to read and understand all information necessary for a good estimation, during an experiment. We expect that it makes only little sense to estimate risks for a fictitious system during an experiment. Method evaluations must take place in a real project, or at least in a real student project. The persons involved will have a lot of this information available from their experience with their software, the environment etc. In fact, in Experiment 2, the results were better, but it cannot be quantified how much of this improvement is due to the fact that an existing system was used in this experiment.

The participants of a prioritization workshop only need half an hour of **training** to understand the principle, but they need **tool support** and a **moderator** who guides them through the estimations step by step. The prioritization workshop has to be prepared by performing a MOQARE analysis. Because of the n–m-relationships between misuse cases and countermeasures the benefit calculations are not straightforward and spreadsheets supporting them cannot be reused without adaptation.

The **moderator** needs good knowledge of the method and much experience to be able to answer all questions and to guide the group well. He/ she should ideally have a perfect understanding of the theoretical background of the method and a thorough preparation. This preparation includes the decision on which misuse cases to consider, on how to write them down (phrasing), on which statistics and other information to supply.

In general, the participants all expressed that they did not feel sure about their probability and damage estimations. During Experiment 1, a discussion arose, as 20% did not mean the same for all misuse cases, there could be no general and satisfactory rule given for the probability estimation. In Experiment 1 (Q5), 8 out of 10 participants marked "because risk estimations are difficult in general" as a reason why his/ her results from different methods were so different.

While we could provide **statistics** about security incidents, this was not the case for other quality attributes. In these fields risk estimations and therefore also the recording of misuse probabilities and damages are not done as systematically as for security misuses (example: Misuse Case "user error prohibits sale"). Such statistics could support risk

estimation and countermeasure prioritization a lot. Statistics about probabilities and damages caused must be available, relating to a system and environment as similar as possible to the present one, if possible with incident statistics or experiences from the same company. As public statistics rarely apply to the same environment as the system under consideration, there will still be high uncertainties in the results of the estimators due to adaptation.

Granularity of the misuse cases: In Section 2.1, we said that grouping misuse cases and countermeasures is a means of taking into account dependencies among them; it is also a means of saving time. The moderators observed that when misuse cases and countermeasures are too general (e.g. including a whole group of misuse scenarios), they are difficult to estimate because we have to average over many scenarios; when they are too detailed, the time need is increased and dependencies among these fine-grained misuse cases irritate the estimators. In free-text comments, the participants also criticized that the misuse cases were too general.

Sometimes the group value is calculated as the **average** of several estimations. This is the case when group consensus on one value could not be achieved or when several scenarios had to be considered for one misuse case. Then, the result differs depending whether the average is made for probabilities and damages separately, or for the risks first and then for the probability and damage estimation are derived from the average risk. As in the following simple example: one estimation of risk is $1 \times 1 = 1$ and the other $2 \times 3 = 6$ -> the average probability is 1.5, the average damage 2 -> the risk is 3. But the average of the two risks is 3.5.) Such a difference was observed once during Experiment 2, where different types of misuse cases were grouped in one, and consequently probability and damage were correlated. There was one frequent misuse case with low damage and another one which happens rarely, but causes high damage. In this case, the average should not be calculated from probability and damage estimations but from the risk.

It is difficult to estimate the damage of all misuse cases in the same **measure**, because in practice, damages influence different goals and therefore are measured in different units like Euro, calendar time, work time, score received for the homework. Misuse case risks are difficult to compare. Maybe it is even impossible to measure benefit and damage with one and the same measure. Instead, one could choose points as a unit, like in FMEA (Stamatis 2003) and other approaches.

We expect the results of the estimations to be sensitive to the definitions and a clear, unambiguous **wording**. In a prioritization workshop, the estimators should agree on the definitions and wording and rephrase countermeasures and misuse cases if necessary. This signifies an additional coordination effort for them. As a positive side effect, these discussions lead to the quality assurance of the requirements. However, in the experiment setting, wording could not be modified, as this would have endangered the comparability of the estimation results.

The misuse cases define the **perspective** of the estimator (e.g.: user perspective, developer or maintainer perspective). This can have advantages as well as disadvantages. A clearly defined perspective helps the estimators to implicitly consider the business goals underlying the countermeasures like user satisfaction or low maintenance cost. However, it is difficult to estimate from an unfamiliar perspective.

Some participants felt that the benefit of a countermeasure **with respect to one misuse case** does not measure its benefit for the whole system. They also criticized that risks and disadvantages caused by a countermeasure are not taken into account by the method. (Remark: In practice, the estimators should define new misuse cases, if they discover important misuse cases caused by the implementation of a countermeasure and treat these misuse cases like the others.)

Some lessons learned refer to the **experimentation**. In the experiments, the estimators needed and welcomed clear rules and step-by-step instructions for each misuse case (e.g. "Imagine that the misuse case happens 10 times and calculate the average damage.") and they demanded unambiguous wording and definitions of the countermeasures and misuse cases, ideally in a quantified way. The results of the estimations are expected to be sensitive to these definitions, to the wording, and to the granularity of the misuse cases and countermeasures. This means a high preparatory effort for an estimation experiment. In Experiment 2, the text for the instructions was 3 to 4 times the volume than the misuse case and countermeasure descriptions. (These long instructions considered all ambiguities and misunderstandings which occurred during the double pre-test.)

The need of support in the risk estimation by **clear rules** must be emphasized. While in the experiment these rules had to be defined by the moderators in order to produce comparable results, in practice, rules, definitions and assumptions can be defined by the estimators themselves, but should be documented.

6.9 Discussion of Validity

The validity of an experiment means that the experiment measures exactly what should be measured.

Höst et al. (2000), following the classification of Cook and Campbell (1979), distinguish between conclusion validity, internal validity, construct validity, and external validity. Conclusion validity is concerned with the relationship of the treatment and the outcome, i.e. whether there is a statistically significant relationship. Our low sample sizes (low number of participants as well as low number of countermeasures prioritized) are an issue here. The low numbers were problematic in hypothesis tests, because many effects observed were not statistically significant. However, the numbers of countermeasures and participants were limited by practical restrictions. Nevertheless, we chose to perform the experiments, because scalability of the methods was not a topic under investigation. The experiments described here were intended as preliminary investigation, the experiment effort had to be manageable for practical reasons, and the numbers are not lower than in comparable investigations (see Section 2.2). Because many observed relationships were not statistically significant, this publication mainly offers indications on relationships, but no proofs. We believe that the results give useful hints for future experiments and applications of the method in practice.

We also believe that many practical challenges observed during the experiment would not have happened in a real project. In the experiment, the moderator had to define the wording of the countermeasures and misuse cases to obtain comparable results in all groups, whereas in practice, the estimators themselves would define them in a way which seems optimal for them.

As we performed only two experiments so far and experiences from the first experiment were used to improve the execution of Experiment 2, several variables differed in these two experiments. Therefore, the observed effects can be due to several factors and not definitely be attributed to one factor. Further experiments should more reliably test the correlations observed. For instance, for Method 1 the differences in the results of Experiment 1 and 2 can be attributed to the group decisions or to more clearly defined prioritization criteria. For Method 2, the differences lay in group discussions and the transparency of the results. Moreover, not the same number and the same misuse cases and countermeasures were treated in the two experiments; in Experiment 1, we used a fictitious example, in Experiment 2 an existing system, known to the participants. Therefore, further experiments

should be made with the same countermeasures, but new combinations of the influencing factors (summarized in Table 4), like transparent individual estimations or group discussions without transparency.

Internal validity is threatened if a relationship is observed between the treatment and the outcome, although there in fact is none. This may happen when the observed effect is caused by other factors of the experiment execution, which are not under investigation. Such effects can, for instance, be caused by the order in which the methods are applied. In Experiment 1, all participants applied Method 1 and 2 in the same order. In Experiment 2, the order was switched, but there could have been a learning effect on the side of the moderators or any other effect which caused a common difference between those groups who executed Method 1 first and then 2 and the others where it was vice versa. However, all groups used the same material, shared the same introductory training and all participants took part on the same afternoon, so no history effect or opinion exchange between participants was possible.

Construct validity refers to the extent to which the experiment setting actually reflects the construct under study, e.g. the ability of the measure chosen. To avoid such difficulties, we chose several variables to measure what a "good" requirements prioritization method is. With regard to several effects, we asked open questions and did a qualitative rather than a quantitative test, so the participants could express their opinions freely.

One explanation why we received very different estimations from different participants was, that they might have taken different perspectives when estimating, e.g. the perspective of a user, a developer, maintainer or manager. This variety of perspectives is realistic and could also be found in an industry project team, because in the requirements prioritization these different views should be taken into account. Partly, the perspective of the estimator was predefined by the wording of the misuse case. This is no threat to validity, but part of the method.

External validity is associated with generalization. If there is a causal relationship between the construct of the cause and the effect, can the result of the study be generalized beyond the scope of our study? We have discussed before, that there is a difference between an experiment where the participants prioritize requirements in an artificial example or whether they prioritize requirements in a system which exists, which is in operation and which they know from the user and developer perspective.

The external validity is an important issue in student experiments. To find out whether a method can well be used in the software engineering practice, it should ideally be tested in real projects by practitioners. Nevertheless, for practical reasons, methods regularly are tested by subjects who are students. This is called "convenience sampling". Robson (2002) states that: "Convenience sampling is sometimes used as a cheap and dirty way of doing a sample survey. You do not know whether or not the findings are representative. […] Nevertheless, studies with students as subjects have made important contributions to empirical software engineering (Carver et al. 2003)."

To what extent are students representative of real stakeholders in real projects? This is a question regularly discussed and investigated empirically. Some of these studies have found that there are no significant differences compared to professionals, e.g. when estimating the effect of ten factors on time to market (Höst et al. 2000) or with respect to the improvement observed when using a software engineering process (Runeson 2003), while others have found that there are significant differences, e.g. (Remus 1989). "The fact that different studies come up with different results is not very surprising. In some areas it is suitable to use students and in others it is not. However, it is very important to clarify under which circumstances students are useful and not." (Robson 2002)

Tichy (2000) gives eight hints for reviewing empirical work. One of these hints is named "Don't dismiss a paper merely for using students as subjects" where he outlines four different situations where it is acceptable to use students as subjects. These are:

- When the students have been trained well enough to perform the task they are asked for.
- To establish trends: when comparing methods, the trend of the difference if not its magnitude can be expected to be comparable to that of practitioners.
- To eliminate hypotheses: if there is no effect observable in the student experiment, it is very unlikely that an effect is observed with professionals.
- Student experiments as a prerequisite for experiments with professionals.

This means that student experiments are important and helpful for initial studies of a question. Observing trends was a major goal of our experiments.

Tichy (2000) especially argues for experiments with computer science (CS) students: In particular, CS graduate students are so close to professional status that the differences are marginal. If anything, CS graduate students are technically more up to date than the 'average' software developer who may not even have a degree in CS. The 'professional', on the other hand, may be better prepared in the application area and may have learnt to deal with systems and organizations of larger scale than a student.

Studies have found that mere length of professional experience has little to do with competence. In other words, you can't use the argument that professionals with years of experience will necessarily solve a given problem better than appropriately prepared (graduate) students. If scale or application experience matters, then the story may be different. We are confident that our students did as reliable and realistic estimations as possible, especially in Experiment 2, as they are already quite experienced with the system under consideration and also have programming experience in the relevant (university) context.

We believe that in fact professionals would not have had more experience with risk estimation methods than the students. The professionals' advantage would rather be their wider experience with the system and countermeasures under consideration. In practice, professionals will expectedly ask less for guidance and take a more active part in adapting the method and wording the misuse cases to their needs than students can in an experiment. For professionals, the resulting priorities would be more important, while for the students applying the method correctly could be more relevant. However, to professionals, Method 2 would have been as new as to the students. This in fact we wanted to test in the experiments: How usable are these methods for someone with no previous experience. Carver et al. (2003) emphasize one difference between students and professionals: In student experiments, a method 'is being measured in the early stages of the learning curve'. This is true in our experiment. For instance, unlike the experiment participants, the authors of this publication with their experience in risk estimation feel very confident with regard to their own results. Even if the scope of the risk is uncertain, they trust in the relations between the risks.

Although our students said they lack the technical and market knowledge necessary for realistic estimates, such knowledge is not fully available to professionals either (Xie et al. 2004). First of all, we do not claim that our experimental experience with a fictitious case is equally valid for a real project. In an experiment, the case and the system environment cannot be defined in detail due to practical limitations. However, Experiment 2 more realistically simulated the situation in a prioritization workshop in a real IT project team, where the developers have no experience with risk estimation and little practical experience. Nevertheless, we expect different results when experienced estimators perform the same task.

7 Conclusion and Future Work

Tichy (2000): "The reality of even the most rigorous approach to empirical work is that experiments normally constitute only a small step forward. By their very nature, experiments explore the relationships between a few variables only, while the real world is far more complex. Due to their limited scope, experiments merely gather evidence."

In the two experiments described above, requirements prioritization based on risk estimation was investigated quantitatively for the first time. Our present experiments have two main results: (I) they highlight challenges of risk estimations and what is important during practical requirements prioritization based on risk estimation and (II) as an empirical pilot study on this topic, they create ideas for more targeted quantitative experiments.

The experiments provide many insights about the challenges and needs of risk estimation. By learning from the feedback of Experiment 1, the quality of the results of risk estimation and the participants' trust in the method could be improved in Experiment 2. The following lessons learned on risk-based requirements prioritization should be taken into account in future experiments and method developments:

- Group discussions and their moderation have positive effects, although group discussions are time-consuming.
- Risk estimation is difficult and requires a lot of information about the system and its environment. Statistical data or own experience about risk probabilities and damages caused are helpful. In an experiment, a real system should be used which the estimators have practical experience with, if possible from different perspectives.
- Providing statistics to the estimators did influence the value of their probability estimations and also lowered the relative standard deviation. However the participants were not sure whether the statistics really facilitated their estimations.
- Transparency is useful, that means to see the effect which each probability and damage estimation has on the resulting risks (and indirectly on the priorities). Tool support which automatically calculates risk can facilitate this transparency. Transparency was found to be advantageous in terms of ease of use and that results are expected to be more realistic—however, these effects were not statistically significant.
- Prioritization criteria, damage metrics and probability measures should be defined clearly and tangibly. Requirements should be unambiguous and quantified where possible.
- The participants' experience with the method and also the moderator's experience enhance confidence in the results.

These results do not contradict the experiences of other researchers on risk estimation and requirements prioritization. Instead, they are more detailed, as we observe the influence of more factors than others and we do this quantitatively. For instance, Feather and Cornford (2003) also observe that for successful risk and benefit estimation it is important to involve experts and "A facilitator is needed to direct these sessions." They support the experts by providing a knowledge base of known misuse cases and countermeasures for the application domain (i.e., spacecraft and software development).

So far, we performed two experiments, but varied several influencing factors, as is summarized in Table 4: In Experiment 2, compared to Experiment 1, we performed moderated group discussions instead of individual estimations, we provided transparency of the risk estimations, varied the order of method execution, countermeasures were less fuzzy and more often measurable, prioritization criteria more tangible and clearer, and we used a real system instead of a fictitious one. These variations had statistically significant effects.

Some more experiments would help to find out which variation of factors was the main cause of the effect for each effect observed.

We recommend that subsequent experiments use the following setting:

- Risk estimations are discussed in a group and organized by a moderator.
- A tool, which automatically calculates misuse case risk and requirement benefit from probability and damage estimations, is used.
- Prioritization criteria, damage metrics and probability measures are defined clearly and tangibly.
- The moderators have the opportunity to acquire experience with the method before the experiment, not only as participants but also with the moderation task. The experiment participants are given as much time as possible to use the method.
- In order to observe statistically relevant effects, the number of participants and groups, as well as the number of requirements prioritized is as high as possible within the practical constraints.

The effects of group discussion, of transparency and of the method used can be re-tested while keeping all other factors fixed. Our observations during our experiments also caused us to ask new questions which we believe to be worth being investigated in more depth. Such questions are:

- Is there an ideal group size for risk estimation? One can imagine that when the group has too many members, the discussion becomes too time-consuming and not all members are heard. Therefore, further experiments might test the effects of group size on time need and satisfaction of the participants.
- Are the results achieved for measurable requirements better than those for fuzzy requirements? For instance, is the standard deviation lower? Do participants feel more certain about their estimates?
- Are damage and probability easier to estimate in points than in percent and Euro, as is done in FMEA (Stamatis 2003)?
- How well can risk estimation predict risk? In order to answer this question, one should have the chance to measure the actual risk later in the project in order to compare it to the values which have been estimated during requirements prioritization. This question is related to the preceding one, as it is possible that estimators can well predict which risk is higher than the other, but no absolute values. In such an experiment, one should also investigate, whether providing statistics to the estimators helps to achieve risk estimates, which are closer to the real values or not.
- As the estimations of different persons vary a lot, one should test the reliability of the results by repeating the same estimation with the same persons. Learning effects and intermediate discussions among participants however, could bias such a re-test.
- It could be interesting to investigate learning effects—of the moderator as well as the estimators–during several subsequent estimation tasks.

Method 1 (the ranking of requirements in two steps) rated better than Method 2 (the risk-based prioritization) with respect to time need and almost all quality variables, except for the standard deviation. One might wonder whether this superiority of the ranking method to the risk-based prioritization in our experiments is due to the fact that in the experiment the subjects were students. We believe that the participants' sensation of uncertainty during our two experiments and other disadvantages of risk estimation observed can partly be explained as a beginner's problem of someone with no experience in risk estimation. Some experience is probably necessary to gain confidence in the method and its results. In fact,

the participants of an unpublished industry case study using the same method for prioritizing requirements in a real software project, said that the method is easy to use and leads to results which are realistic and useful. Feather et al. (Feather and Cornford 2003; Feather et al. 2006) also use risk estimation, even for high numbers of requirements, successfully. Therefore, we do not conclude from our experiments that risk-based prioritization must be abandoned, but that the participants must be carefully chosen and prepared. The influence of the estimators' expertise cannot be underestimated. Feather and Cornford (2003) observe that their "combined expertise" must encompass goals, requirements and constraints, misuse cases, as well as preventative, detecting or alleviating countermeasures.

An approach which supports probability estimation is offered by Bayesian Belief Networks. It has been applied to software cost estimation (Chulani et al. 1999; Devnani-Chulani 1999), as a way to combine historical data and expert judgement. For risk estimation, they offer a way of systematic reasoning on probabilities and damages and for modelling dependencies of the pre—and post-conditions of several misuse case scenarios. However, we have not applied them so far, because we expect them to enhance some of the problems we encountered during our experiment: time need, but also perceived complexity of the method. Modelling different alternative scenarios in Bayesian Networks would complicate the estimation task to a degree that turns it too time-consuming for being used in an estimation workshop. However, we expect that it makes sense that those, who prepare the estimation workshop by deciding on the most relevant misuse cases which are to be considered, use Bayesian Networks for supporting their choice. Such systematic reasoning aided by Bayesian Networks can be done in the style of: "We want to know how probable it is that user error impedes a planned purchase (Misuse Case 1 of Experiment 1). What does this probability depend on? On the type of user. We know that all users are individuals which have not obtained any training about the use of the software. They know how to navigate on web sites and office applications. Knowing this, how high is the a posteriori probability of Misuse Case 1?" However, this has been done intuitively anyhow. Bayes' formulas are more interesting when alternative cases or scenarios are discussed, such as two types of users and their corresponding probabilities. Bayes' approach is also a way of using statistical data which do not describe a misuse case's probability, but of a factor leading to it, like the user type. However, considering several scenarios demands more time than considering only the most probable scenario and Bayes' theory demands the estimation of several conditional probabilities per misuse case probability, which enhances time need further and also the uncertainty of the results. Apart from these difficulties, we expect that when using Bayesian Networks, similar observations would be made as during our experiment without the Bayesian Network.

The high time need and the high degree of uncertainty of the risk and benefit estimations impact the usefulness of quantitative risk estimations. There is a saying that in project management it is not the project plan that is important, but the process of planning. Ambler (Ambler 2002) remarks: "Modeling is similar to planning—most of the value is in the activity of modeling, not in the model itself." We would say that the same is true for risk-based requirements prioritization. Despite all challenges met in the experiments, we believe that risk estimation is a good means of discussing priorities of requirements. Risk–among other criteria—is an important prioritization criterion. As a side effect, this process forces us to phrase the requirements comprehensibly and to identify open questions and missing knowledge, and it requires stakeholders with different experiences to communicate.

As a compromise and based on our current knowledge, we recommend to invest the effort of risk estimation only in the most critical requirements. It probably is most efficient

to first prioritize the requirements with a simpler method and then to use risk estimation for analyzing some especially important requirements in more detail.

Annex A: Data and Data Analysis

This annex for Experiment 1 and 2 describes the quantitative analysis of the variables defined in Section 3. For each variable, the results from both experiments and all methods are presented together. Their interpretation, especially how we believe that these variables have been influenced by the influencing factors, is discussed in Section 6.

A.1 Time Consumption

Method 1 demands to determine only two values (group and priority) for each of the countermeasures. Method 2 demands the estimation of two probabilities and two damages per countermeasure.

In Experiment 1, the time needed for Method 2 (risk estimation) is significantly higher than in Method 1 (ranking). The average time needed for Q1 was 6.6 min for those 7 participants who noted it. Q2 took an average of 31.8 min. In Experiment 2, the time need averaged over those groups who performed this method first, was 17.5 min for Method 1 and 37.3 min for Method 2. (We count only these groups, because of the learning effect observed.)

From these numbers, we calculated the time need per countermeasure (Table 5) and also the time need per estimation (Table 6.), as the number of estimations per countermeasure in Method 2 depends on the number of misuse cases.

A.2 Priorities

The resulting **priorities** of the countermeasures varied widely among the participants and groups in both experiments, for both methods, as can be seen from Tables 15, 16, 17, 18. This means that they differ greatly about the importance of the countermeasures. The same countermeasure could have the highest priority (1) for one participant/ group and the lowest

Table 15 Priorities resulting from Experiment 1 with Method 1 ("1" standing for the most important one): The "1" is row "R1" and column "1" means that according to the results of participant 1, Countermeasure R1 is the most important one

Participant:	1	2	3	4	5	6	7	8	9	10	Average	Priority	Standard deviation
R1	1	4	1	4	1	6	8	3	1	1	3.00	1	2.49443826
R2	5	9	4	7	2	8	7	5	5	4	5.60	5	2.11869981
R3	2	1	6	2	3	9	2	1	4	3	3.30	2	2.49666444
R4	4	2	9	8	8	7	9	2	3	7	5.90	6	2.84604989
R5	3	3	3	1	4	1	6	7	6	2	3.60	3	2.11869981
R6	9	8	8	3	5	5	5	4	9	5	6.10	7	2.18326972
R7	6	6	5	9	7	3	4	9	7	8	6.40	9	2.01108042
R8	7	7	2	5	6	2	3	8	2	6	4.80	4	2.34757558
R9	8	5	7	6	9	4	1	6	8	9	6.30	8	2.49666444

Column "Average" is the average priority of a countermeasure, averaged over all participants, and column "Priority" shows the order of priority for these averages

Table 16 Priorities resulting from Experiment 1 with Method 2 ("1" indicating the most important one)

Participant:	1	2	3	4	5	6	7	8	9	10	Average	Priority	Standard deviation
R1	3	1	2	3	9	6	4.5	3	3	2	3.65	2	2.333928496
R2	9	5	5	5	3	1	8	5	6	5	5.20	5	2.250925735
R3	7	6	9	9	6.5	7	2	8	8	9	7.15	9	2.108843812
R4	8	7	8	4	2	3	4.5	9	4	7	5.65	6	2.427275565
R5	5	2	1	1	1	2	7	2	9	1	3.10	1	2.884826203
R6	1	8	6	8	6.5	8	3	7	5	5	5.75	7	2.324387613
R7	4	3	7	2	4	9	1	6	7	3	4.60	4	2.547329757
R8	6	9	3.5	6.5	6.5	4.5	9	4	1	8	5.80	8	2.573367875
R9	2	4	3.5	6.5	6.5	4.5	6	1	2	6	4.20	3	2.043961296

for another. This was the case even in Method 1, where the results were transparent to and manipulable by the estimators, while in Method 2 the lacking transparency and indirect manipulability of the priorities could possibly lead to results which are unexpected by the estimators.

In Method 1, the averages of the priorities lie between 3.0 and 6.4 for the individual countermeasures. In Method 2, the averages are between 3.1 and 7.2. In Experiment 2, all groups agreed that R3 is one of the least important: In Method 1, all seven groups gave R3 the lowest priority 7, while in Method 2, this was the case for five groups, once it received priority 6 and once 5. The priority averages in Method 1 (not counting R3), vary from 2.57 to 5.00, and in Method 2 from 2.29 to 4.21.

We are sure that these wide ranges are not caused by the misunderstanding whether "1" stands for the highest priority or the lowest. In Method 1, the priority 1 countermeasure for all participants was found in the "high benefit" group. In Method 2, the priorities were determined by us, based on the calculated benefits.

A.a Standard Deviation of Priorities

The **standard deviations** s found among the ten estimations of the nine priorities in Experiment 1 are shown in Table 7 and also the seven estimations found among the seven priorities in Experiment 2 in Table 8. The differences of the standard deviations between the methods in the same experiment are very low and statistically not significant. The standard deviations in Experiment 2 are lower than in Experiment 1 because fewer countermeasures were prioritized, but also when divided by the average priority (which is $(n+1)/2$), in Experiment 2 the standard deviation is lower (see Table 9).

Table 17 Priorities resulting from Experiment 2 with Method 1 ("1" indicating the most important one)

Team	1	2	3	4	5	7	8	Average	Priority	Standard deviation
R1	5	2	5	5	4	1	1	3.29	3-4	1.88982237
R2	4	4	3	2	3	4	6	3.71	5	1.25356634
R3	7	7	7	7	7	7	7	7.00	7	0
R4	1	5	2	1	2	2	5	2.57	1	1.71824939
R5	2	6	6	6	5	6	4	5.00	6	1.52752523
R6	3	1	4	3	6	3	3	3.29	3-4	1.49602648
R7	6	3	1	4	1	5	2	3.14	2	1.95180015

Table 18 Priorities resulting from Experiment 2 with Method 2 ("1" indicating the most important one)

Team	1	2	3	4	5	7	8	Average	Priority	Standard deviation
R1	2.5	5	4	4	5	3	6	4.21	6	1.219875091
R2	6	2	3	6	4	4	2	3.86	3	1.676163420
R3	7	7	6	7	7	7	5	6.57	7	0.786795792
R4	4	1	2	1	2	2	4	2.29	1	1.253566341
R5	2.5	4	5	5	3	5	3	3.93	4	1.096531328
R6	1	3	7	3	6	1	7	4.00	5	2.645751311
R7	5	6	1	2	1	6	1	3.14	2	2.410295378

A.b Ease of Use

The **ease of use** of each method as rated by the participants is shown in Table 10. We attribute points to the answers: Very easy =2 points, Easy =1, Undecided =0, Difficult =−1, Very difficult =−2.

A.c Participants Expect Their Estimations to Be Realistic

In Experiment 1, immediately after the estimations, but before they knew the resulting priorities, we asked the participants whether they expected to have made reasonable, realistic and useful estimations (question Q4b, see Table 11). In Experiment 2, they were asked whether they believe that their results were realistic. At this point of time, they knew the priorities. Points were attributed to the answers: very = 2 points, rather =1, undecided = 0, rather not =−1, not at all = −2 points.

It is interesting to note that in Experiment 2, the average points for the probability and damage estimations were 0.13, i.e. lower than for the priorities (but not statistically significantly due to high variations). This means that some participants trusted the method to deliver priorities which are more realistic than the risk estimations on which they are based.

A.d Accuracy of the Results

Accuracy here means that after the experiment the participants considered the countermeasure priorities which resulted from their risk estimations to be plausible and reflecting their views. In Experiment 1, this question was asked in the post-test session 1 week after the experiment, in Experiment 2, the question was asked during the experiment session directly after the estimations.

In Experiment 1, nine participants answered this question (free-text answers). Four were in favour of Method 1, two wrote, that their first impression was that Method 1 delivered the most plausible results, because they corresponded to their intuitive priorities, but as Method 2 was a systematic method, probably this method should provide the best results. Another participant wrote that all methods delivered plausible as well as less plausible results for the different countermeasures. The last participant wrote that they were all plausible: Method 1 reflected his own perception, neglecting risk and cost. Method 2 was plausible as well, taking into account risk and cost.

In Experiment 2, Method 1 got an average of 1.29 points as a result to this question; Method 2 got 0.13 points on a scale between −2 and +2 (very well = 2 points, rather yes = 1, undecided = 0, rather not =−1, very badly = −2 points). This difference is statistically significant.

A.e Frequency of Naming a Misuse Case or Countermeasure as Especially Uncertain

The participants were asked to name the countermeasures or misuse cases for which they considered their estimated values especially uncertain. We counted how often a certain countermeasure (in Method 1) or a certain misuse case (in Method 2) was named (Variable e). To compare the methods and countermeasures, we calculated the average frequency with which a countermeasure or misuse case was named here, averaged over all countermeasures/misuse cases. We did so, because the number of countermeasures and misuse cases was not the same in the two experiments and the methods applied. The results are summarized in Table 12.

In Experiment 1, regarding Method 1, R8 was never named and R3 only once. R5 was mentioned twice, R1, R2, R4 and R9 three times, R7 four times and R6 five times by the ten participants. Concerning Method 2, eight out of ten participants named specific misuse cases, while six additionally or instead said that they were uncertain about practically all of them. If we take the latter group literally, each misuse case was named with the average frequency 0.7. If we do not count the participants stating they were uncertain concerning all misuse cases, the numbers are about 0.2 for the probability estimations as well as for the damage estimations.

In Experiment 2, regarding Method 1, R1 was named seven times (by the 18 participants who answered to this question), R6 and R7 four times, R2, R4 and R5 three times and R3 only once. Except for R1, which seems to be especially difficult to judge, and R3, which caused almost no irritation, most countermeasures seem to be equally difficult. Few correlations are seen among the answers of members of the same group. Only once, all three group members agreed that R1 was difficult, and three times two of three or four group members agreed about the same countermeasure.

In Experiment 2, concerning the probability estimation in Method 2, Misuse Cases 1–3 were named four times (by the 18 participants who answered this question), Misuse Case 4 only once, Misuse Case 5 nine times and Misuse Case 6 eight times. Only once, all three group members agreed about the same misuse case: for Misuse Case 5 they all found probability estimation difficult. Five times, two group members agreed about the same misuse case.

In Experiment 2, concerning the damage estimation in Method 2, the misuse cases were named with the following frequencies (by the 18 participants who answered this question): Misuse Case 6 nine times, 1 and 4 seven times each, 3 five times, 5 four times, and 2 only once. Only once all three group members agreed concerning 3. Seven times, two group members considered the same misuse case's damage estimation to be difficult. This means an average frequency for a misuse case of 0.3. These numbers are approximately the same as for the probability estimation.

The differences observed between methods (during the same experiment) and during different experiments for the same method are statistically significant.

A.3 Influence of Statistics

In Experiment 1, we tested the influence of statistics provided to the estimators. In Q5d, 8 out of 9 participants now clearly attributed different probabilities (reference risk) to the two security misuse cases, usually much higher ones, see Table 13 and Table 14. One participant wrote that he estimated the same probabilities as before (0.5% and 0.1%), but we doubt whether they were really derived from the statistics, as they differ too much from the estimations of the other participants.

When questioned whether the statistics facilitated the probability estimation, we received free-text answers. Some of them were: "They definitely were helpful. One feels more certain, thanks to this information." Others were more sceptical: "I would say that the statistics have strongly influenced my estimations. However, I wonder how similar the systems of these companies are to the reference system in the case study. Only when this is known, one can say whether the high estimated value is justified. I think that I still do not have enough information to deliver a good estimation." All together, 4 participants out of 9 wrote that the statistics were helpful, while one wrote they did not influence the estimation and four wrote they influenced the estimations, but they still were sceptical whether the estimated values were exact.

We expected that when providing the participants with several statistics, the standard deviation (relative to the average) of their estimations to decrease (Variable a). As one can see in Table 13 and Table 14, the estimated probabilities still differed among the participants. This was to be expected because four statistics were given, which did not apply to exactly the same environment as the case study. Therefore, interpretations and adaptations were necessary. The coefficient of variation became less than half by using the statistics.

References

Ambler SW (2002) Agile modeling-effective practices for extreme programming and the unified process. Wiley Computer Publishing, New York

Arora A, Hall D, Pinto CA, Ramsey D, Telang R (2004) An ounce of prevention vs. a pound of cure: how can we measure the value of IT security solutions? Carnegie Mellon CyLab

Beck K (2000) Extreme programming explained. Upper Saddle River, Addison-Wesley

Berander P (2004) Prioritization of Stakeholder Needs in Software Engineering. Understanding and Evaluation. Licenciate Thesis, Blekinge Institute of Technology, Sweden, Licentiate Series No 2004:12

Berander P, Jönsson P (2006) Hierarchical cumulative voting (HCV)-prioritization of requirements in hierarchies. Int J Softw Eng Knowl Eng 16(6):819–849. doi:10.1142/S0218194006003026

Carver J, Shull F, Basili V (2003) Observational Studies to Accelerate Process Experience in Classroom Studies: An Evaluation. Proc. of the 2003 Int. Symposium on Empirical Software Eng. ISESE, Rome, Italy, pp 72–79

Cook TD, Campbell DT (1979) Quasi-Experimentation–Design and Analysis Issues for Field Settings. Houghton Mifflin Company, Boston

Chulani S, Boehm B, Steece B (1999) Bayesian analysis of empirical software engineering cost models. IEEE Trans Softw Eng 25(4):573–583. doi:10.1109/32.799958

Daneva M, Herrmann A (2008) Requirements Prioritization Based on Benefit and Cost Prediction: A Method Classification Framework. Track on Software Process and Product Improvement (SPPI), 34th Euromicro Conf., Parma, Italy, 1–5 Sept. 2008

Davis AM (2003) The Art of requirements triage. IEEE Comput 36(3):42–49

Denne M, Cleland-Huang J (2003) Software by Numbers: Low-Risk, High-Return Development. Prentice-Hall

Devnani-Chulani S (1999) Bayesian Analysis of Software Cost and Quality Models. A Dissertation Presented to the Faculty of the Graduate School, University of Southern California, http://sunset.usc.edu/publications/TECHRPTS/PhD_Dissertations/files/SChulani_Dissertation.pdf

Dutoit AH, Paech B (2002) Rationale-based use case specification. Requirements Eng. J. 7:3–19. doi:10.1007/s007660200001

Dutoit AH, Paech B (2003) Eliciting and maintaining knowledge for requirements evolution. In: Aurum A, Jeffery R, Wohlin C, Handzic M (eds) Managing Software Engineering Knowledge. Springer, Berlin, pp 135–156

Feather MS, Cornford SL (2003) Quantitative risk-based requirements reasoning. Requirements Eng J 8 (4):248–265. doi:10.1007/s00766-002-0160-y

Feather MS, Cornford SL, Larson T (2000a) Combining the best attributes of qualitative and quantitative risk management tool support. Proc. 15th IEEE Int. Conf. on automated software eng., Grenoble, France, 11–15 September 2000. IEEE Computer Society, 309–312

Feather MS, Cornford SL, Gibbel M (2000b) Scalable mechanisms for requirements interaction management. IEEE Int. Conf. on Requirements Eng., Schaumburg, USA, pp 119–129

Feather MS, Cornford SL, Kiper JD, Menzies T (2006) Experiences using Visualization Techniques to Present Requirements, Risks to Them, and Options for Risk Mitigation. Proc. Int. Workshop on Requirements Eng. Visualization, Minneapolis/ St. Paul, Minnesota

Herrmann A, Paech B (2005) Quality Misuse. Proc. 11th Int. Workshop on Requirements Eng. for Software Quality, Foundations of Software Quality REFSQ, Essener Informatik Beiträge. Band 10:193–199

Herrmann A, Paech B (2006) Benefit Estimation of Requirements Based on a Utility Function. Proc. 12th Int. Workshop on Requirements Eng. for Software Quality, Foundations of Software Quality REFSQ, Essener Informatik Beiträge. Band 11:249–250

Herrmann A, Paech B (2008a) Practical Challenges of Requirements Prioritization Based on Risk Estimation: Result of Two Student Experiments. Technical Report SWEHD-TR-2008-03, University of Heidelberg, http://www-swe.informatik.uni-heidelberg.de/research/publications/reports.htm

Herrmann A, Paech B (2008b) MOQARE: Misuse-oriented quality requirements engineering. Requirements Eng J 13(1):73–86. doi:10.1007/s00766-007-0058-9

Herrmann A, Rückert J, Paech B (2006) Exploring the Interoperability of Web Services using MOQARE. Proc. IS-TSPQ First Int. Workshop on Interoperability Solutions to Trust, Security, Policies and QoS for Enhanced Enterprise Systems, Bordeaux, France

Höst M, Regnell B, Wohlin C (2000) Using students as subjects—a comparative study of students and professionals in lead-time impact assessment. Empir Softw Eng 5(3):201–214. doi:10.1023/A:1026586415054

ISO (International Standards Organization) (2002) ISO, Risk management–Vocabulary–Guidelines for use in standards, ISO Guide 73. International Standards Organization, Geneva

Jalali O, Menzies T, Feather M (2008) Optimizing requirements decisions with KEYS. Proc. 4th Int. Workshop on Predictor models in software eng., Int. Conf. on Software Eng., Leipzig, Germany. ACM New York, NY, USA, pp 79–86

Karlsson J (1996) Software requirements prioritising. Proc. 2nd Int. Conf. Requirements Eng., 110–116

Karlsson J, Wohlin C, Regnell B (1998) An evaluation of methods for prioritizing software requirements. Inf Softw Technol 39:939–947. doi:10.1016/S0950-5849(97)00053-0

Karlsson L, Berander P, Regnell B, Wohlin C (2004) Requirements Prioritisation: An Experiment on Exhaustive Pair-Wise Comparisons versus Planning Game Partitioning. Berander, P. Prioritization of Stakeholder Needs in Software Engineering, Understanding and Evaluation, Licenciate Thesis, Blekinge Institute of Technology, Licentiate Series No 2004:12

Karlsson L, Thelin T, Regnell B, Berander P, Wohlin C (2007) Pair-wise comparisons versus planning game partitioning—experiments on requirements prioritisation techniques. Empir Softw Eng 12(1):3–33. doi:10.1007/s10664-006-7240-4

Kontio J (1996) The Riskit Method for Software Risk Management, Version 1.00. University of Maryland. College park, MD, Computer Science Technical Reports CS-TR-3782

Leffingwell D, Widrig D (2000) Managing Software Requirements—A Unified Approach. Addison-Wesley, Reading, Massachusetts, USA

Mayer N, Rifaut A, Dubois E (2005) Towards a risk-based security requirements engineering framework. Proc. 11th Int. workshop on requirements eng. for software quality, foundations of software quality REFSQ, essener informatik beiträge. Band 10:89–104

Menzies M, Kiper J, Feather M (2003) Improved software engineering decision support through automatic argument reduction tools. 2nd Int. Workshop on Software Eng. Decision Support SEDECS2003, part of SEKE2003, June 2003

Ngo-The A, Ruhe G (2005) Decision Support in Requirements Engineering. In: Aurum A, Wohlin C (eds) Engineering and Managing Software Requirements. Springer, Berlin, Heidelberg

Papadacci E, Salinesi C, Rolland C (2004) Payoff Analysis in Goal-Oriented Requirements Engineering. Proc. 10th Int. Workshop on Requirements Eng. for Software Quality, Foundations of Software Quality REFSQ

Park J, Port D, Boehm B, In H (1999) Supporting distributed collaborative prioritization for winwin requirements capture and negotiations. Proc. Int. 3rd World Multiconference on Systemics, Cybernetics and Informatics SCI'99 2:578–584

Raiffa H, Richardson J, Metcalfe D (2002) Negotiation analysis—the science and art of collaborative decision making. Belknap, Cambridge

Regnell B, Höst M, Natt och Dag J, Beremark P, Hjelm T (2001) An industrial case study on distributed prioritisation in market-driven requirements engineering for packaged software. Requirements Eng 6:51–62. doi:10.1007/s007660170015

Remus W (1989) Using Students as Subjects in Experiments on Decision Support Systems. Proc. 22nd Annual Hawaii Int. Conf. on System Sciences, Vol. III: Decision Support and Knowledge Based Systems Track, pp 176–180

Richardson R (2003) 2003 CSI/FBI Computer Crime and Security Survey. Computer Security Institute. http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2003.pdf (last visit: nov 07)
Robson C (2002) Real World Research. Blackwell Publishing, Cornwall, UK
Ruhe G, Eberlein A, Pfahl D (2003) Trade-off analysis for requirements selection. Int J Softw Eng Knowl Eng 13(4):345–366. doi:10.1142/S0218194003001378
Runeson P (2003) Using students as experiment subjects—an analysis on graduate and freshmen student data. Proc. Int. Conf. Empirical Assessment and Evaluation in Software Eng. EASE Keele, UK, pp 95–102
Ryan K, Karlsson J (1997) Prioritizing Software Requirements in an Industrial Setting. Proc. Int. Conf. on Software Eng., pp 564–565
Saaty TL (1980) The Analytic Hierarchy Process. McGraw-Hill, New York
Sindre G, Opdahl AL (2000) Eliciting security requirements by misuse cases. Proc. TOOLS Pacific 2000:120–131
Sindre G, Opdahl AL (2001) Templates for Misuse Case Description. Proc. 7th Int. Workshop on Requirements Eng.: Foundation of Software Quality–REFSQ, Essener Informatik Beiträge Band 6. Essen, Germany, pp 125–136
Stamatis DH (2003) Failure Mode and Effect Analysis–FMEA from Theory to Execution. American Society for Quality Press, Milwauki, USA
Stylianou AC, Kumar RL, Khouja MJ (1997) A total quality management-based systems development process. ACM SIGMIS Database 28(3):59–71. doi:10.1145/272657.272691
Sysiphus (2007) http://sysiphus.in.tum.de/ (last visit: nov 2007)
Tichy WF (2000) Hints for reviewing empirical work in software engineering. Empir Softw Eng 5(4):309–312. doi:10.1023/A:1009844119158
Tversky A, Kahneman D (1974) Judgment under uncertainty: heuristics and biases. Science 185:1124–1131. doi:10.1126/science.185.4157.1124
van den Akker M, Brinkkemper S, Diepen G, Versendaal J (2004) Flexible Release Composition using Integer Linear Programming. Technical Report UU-CS-2004-063, Institute of Information and Computing Sciences, Utrecht University, Netherlands
van den Akker M, Brinkkemper S, Diepen G, Versendaal J (2006) Software product release planning through optimization and what-if analysis. Technical Report UU-CS-2006-63, Department of Information and Computing Sciences, Utrecht University, The Netherlands
Wiegers K (1999) First things first: prioritizing requirements. Software Development 7(9) http://www.processimpact.com/pubs.shtml#requirements (last visit: nov 07)
Wolf T, Dutoit AH (2004) A rationale-based analysis tool. Proc. 13th Int. Conf. on Intelligent on Adaptive Systems and Software Eng, Nice, France
Xie N, Mead NR, Chen P, Dean M, Lopez L, Ojoko-Adams D, Osman H (2004) SQUARE Project: Cost/Benefit Analysis Framework for Information Security Improvement Projects in Small Companies. Software Engineering Institute, Carnegie Mellon University, Technical Note CMU/SEI-2004-TN-045

Andrea Herrmann is a Post Doc scientist with her research and teaching focus on requirements engineering and project management. She has 7 years of scientific experience and for 6 years worked in software projects. She is spokeswoman of the special interest group “Requirements Engineering” in the German Computer Science Society.

Barbara Paech holds the chair "Software Engineering" at the University of Heidelberg. Till October 2003 she was department head at the Fraunhofer Institute Experimental Software Engineering. Her teaching and research focuses on methods and processes to ensure quality of software with adequate effort. Since many years she is particularly active in the area of requirements and rational engineering. She has headed several industrial, national and international research and transfer projects. She is spokeswoman of the section "Software Engineering" in the German computer science society.