Identification of More Risks Can Lead to Increased Over-Optimism of and Over-Confidence in Software Development Effort Estimates

Magne Jørgensen

Context*: Software professionals are, on average, over-optimistic about the required effort usage and over-confident about the accuracy of their effort estimates.* **Objective**: *A better understanding of the mechanisms leading to the over-optimism and over-confidence may enable better estimation processes and, as a consequence, better managed software development projects.* **Method**: *We hypothesize that there are situations where more work on risk identification leads to increased over-optimism and over-confidence in software development effort estimates, instead of the intended improvement of realism. Four experiments with software professionals are conducted to test the hypothesis.* **Results***: All four experiments provide results in support of the hypothesis. Possible explanations of the counter-intuitive finding relate to results from cognitive science on "illusion-of-control", "cognitive accessibility", "the peak-end rule" and "risk as feeling."* **Conclusions***: Thorough work on risk identification is essential for many purposes and our results should not lead to less emphasis on this activity. Our results do, however, suggest that it matters how risk identification and judgment-based effort estimation processes are combined. A simple approach for better combination of risk identification work and effort estimation is suggested.*

1. Introduction

Software development project plans are frequently based on over-optimistic effort estimates and over-confident assessments of estimation accuracy. The average effort estimation overrun seems to be about 30% (Moløkken and Jørgensen 2003), and 90% confidence minimum-maximum intervals of effort usage typically achieve hit rates of only 60-70% (Jørgensen, Teigen et al. 2004). This over-optimism and over-confidence may have many negative consequences, such as poorly managed projects and project delivery delays.

A disturbing finding, potentially contributing to this unfortunate situation, is that more work on risk identification seems to be capable of increasing, rather than decreasing, the level of over-optimism and over-confidence. The findings reported in (Sanna and Schwarz 2004) are perhaps the strongest evidence in support of this counter-intuitive effect. The authors found that students instructed to provide 12 ways to fail on an upcoming exam, produced more optimistic estimates of the exam outcome compared with students instructed to provide only three ways to fail. The same effect of identification of more ways to fail was present when the students predicted the time their exam preparation would be completed. The students who provided 12 ways to fail were significantly more over-optimistic regarding completion time than those providing only three ways. Theories regarding these findings are presented in Section 2.

We strongly believe that thorough work on risk identification should be part of most proper project estimation, planning and management processes. The goal of this work is therefore far from warning against spending much effort on risk identification. It is however important to design estimation processes where risk identification and effort estimation is combined so that biases towards over-optimism and over-confidence are reduced. This is the main goal of our work. Besides the goal of

improving estimation processes this paper also seeks a better understanding of the cognitive processes involved judgment-based effort estimation.

Based on these two goals, we hypothesize that there are situations where:

Identification of more risks leads to more over-optimistic effort estimates, and higher over-confidence in the estimation accuracy and success probability of software projects.

Spending effort to identify more risks sometimes leads to risk discovery or insight of great importance. In those cases, it is obvious that more effort on risk identification leads to more realistic effort estimates and more realistic success probability assessments. The focus of this paper is not on such situations. Instead, it targets situations with increased numbers of identified risks, without any substantial new insight in use of effort.

The four experiments we present will mainly analyze whether effort estimates become lower and the confidence becomes higher with more risk identification. To relate the findings to the above hypothesis we therefore add other results, such as the well-documented tendency towards over-optimism and over-confidence among software developers and, in one of the experiment (Experiment C), information about what other developers actually used to implement the specified software.

A risk in the context of this paper is an uncertain event or condition that, if it occurs, has a negative effect on a project's objective. This differs from the definition in for example the Project Management Body of Knowledge (PMBOK) (PMI 2008), where both positive and negative events and conditions are counted as risks. A risk may be characterized by the likelihood of the occurrence of the event or condition and the severity of the consequences. The effect of a set of risks on the most likely use of effort depends on the likelihoods and the severities of the uncertain events and conditions. There are many ways of combining risk identification and effort estimation processes. The one we examine in this paper is based on a process starting with risk identification and use of the identified risk as input to the effort estimation process.

Notice that this paper does not study complete risk analysis or risk management processes. The sole focus is on the relationship between simple, unaided risk identification and judgment-based effort estimation, when performed by software developers. We do not study the effect of thorough risk analysis and risk management processes with experienced project managers. Such processes may of course have a substantial positive effect on the project execution. Even the best risk management processes may however suffer, when trying to manage a project based on over-optimistic effort estimates provided by the software developers.

2. Related Work

The students participating in the study described in (Sanna and Schwarz 2004) had no formal risk models or historical risk data when assessing the effect of the identified risks (ways to fail) on exam results. Instead they relied on judgment-based prediction processes. This use of judgment-based prediction processes seems to be an essential enabler of the observed counter-intuitive effect of increased optimism and confidence from more risk identification, see Section 2.1. Cognitive theories potentially explaining the hypothesized effect of identification of more risks on effort estimates and success probabilities are described and discussed in Sections 2.2 (illusion-of-control), 2.3 (cognitive accessibility) and 2.4 (peak-end rule). We discuss

these explanations before we present the experiments and not as part of the discussion of the results from our experiments, because these theories affected how we designed our experiments.

2.1 Judgment-based Reasoning Processes

A core distinction between reasoning strategies is that between judgment-based (intuitive, unconscious, implicit, tacit) and analytic (conscious, explicit) processes. This distinction is reflected in the well-established dual-process theory of cognition (Evans 2003). The dual-process theory of cognition says that peoples' judgments reflect the operation of two distinct thinking systems, analysis and intuition, and that these two systems have a different evolutionary history and to some extent a different neurological basis. The analytic system uses probability calculus and formal logic to perform, for example, risk analysis and is relatively slow, effortful and requires conscious control. The intuitive system, which is the basis for what we term judgment-based processes in this paper, is faster, seemingly effort-less, automatic and not very accessible to human awareness. The intuitive system is believed to be the, evolutionary speaking, oldest and the one most people feel is most natural to use when responding to risks. The two thinking systems interact in many ways, but do also compete. A particularly interesting relationship is that, in several situations, we seem to have high confidence in the method and low confidence in the outcome when applying analytical processes, while we have low confidence in the method and high confidence in the outcome when applying judgment-based processes (Hammond, Hamm et al. 1987).

We have previously reported that the above relationship, combined with a need to reduce cognitive conflicts, can explain why formal software development effort estimation models (that rely heavily on judgment-based input) become "expert judgment in disguise" (Jørgensen and Gruschke 2005). To reduce the cognitive conflict, the software developers may, perhaps unconsciously, adapt the judgment-based input given to the effort estimation model so that the output of the model corresponds with what intuitively feels correct. This way the software developers are able to pretend a use the preferred analytical process and still believe in the output.

The relatively low calculation capacity of the human brain limits the selection of strategies that are available for judgment-based effort estimation and project success assessments. It is, for example, unlikely that software developers' judgments are able to follow a formally correct process for assessing the total impact of identified risks on the required use of effort. Instead, the judgment-based estimation strategies have to be based on processes that are sufficiently accurate and simple to be processes with acceptable speed in real-life situations (Slovic, Finucance et al. 2007), i.e., they must follow the "satisficing" rather than the "optimizing" paradigm (Simon 1957). A consequence of the reliance of "satisficing" processes is that the outcome sometimes are biased and inaccurate, particularly when the context in which it was (evolutionary) developed is different from the one in which it is currently used. The presence of risk or uncertainty assessment biases are reported in many domains, including risk assessment in nuclear plants (Otway and von Winterfeldt 1992), finance (de Venter and Michayluk 2008) and in software development (Jørgensen, Teigen et al. 2004).

The existence of biases in judgment-based effort estimation contexts understandably leads to a wish to replace the judgment-based processes with formal models integrating risk and effort estimation. Such models may, such as models combining individual probabilities based on Monte Carlo simulation, may however be

very complex and require information hard to extract and validate based on objective data. Perhaps for these reasons, formal risk assessment models are seldom used by project managers (White and Fortune 2002). Instead, the assessment of the effect of the identified risks on the effort estimate and success probability is typically based on judgment-based processes.

2.2 Illusion-of-Control

One potential explanation of the counter-intuitive optimism and confidence-increasing effect of more risk identification reported in (Sanna and Schwarz 2004) is the well-documented illusion-of-control phenomenon (Langer 1975; Thompson, Armstrong et al. 1998), i.e., that we tend to believe that we can control risks to a greater extent than we actually can.

The illusion-of-control effect seems to become stronger with increased levels of familiarity, involvement and desire to experience control (Thompson, Armstrong et al. 1998). These conditions may be present when spending more effort on risk identification. Particularly interesting is in our opinion the strong emphasis in the project management tradition (where risk identification belongs) to *control* the risk. Risks are not just there. A good software professional should be able to control them, which is of course a very laudable and important goal. An additional possible consequence of this strong emphasis on and wish for control of risks may, however, be that more effort on risk identification (and analysis) leads to an even higher level of belief in that the risks are controllable, e.g., through higher degree of familiarity, involvement and desire. The empirical documentation of the unconscious transfer from desire in something to a belief that this will happen is strong, see for example the following studies on "wishful thinking": (Babad and Katz 1991; Henry 1994; Buehler, Griffin et al. 1997). Interesting is also the finding reported in (Pelham and Neter 1995), where higher motivation of accurate judgment, possibly corresponding to higher desire of controlling the risks, led to more accurate judgment for easy tasks, but lower accuracy for complex tasks. Most software development effort estimation tasks clearly belong to the set of complex tasks.

The illusion-of-control effect is, in our opinion, also interesting when it comes to other phenomena in software development contexts. In particular, we believe it could explain software developers' feeling that they have found the last coding error after thoroughly inspecting their own code for errors. Previous experience should have taught them that this is not likely to be the case, but the feeling of last error found (the feeling of control) is nevertheless there.

2.3 Accessibility Effect and the Over-weighting of the Last Identified Risk

The accessibility (mental availability, ease of mental recall) of risk scenarios and not only the risk scenarios themselves may count when people make their judgment about the total risk. One well-documented heuristic describing this is the availability heuristic (Tversky and Kahneman 1974; Milburn 1978). This heuristic implies that we use the mental accessibility of events and experience as indicator of its importance. One factor known to increase the accessibility of a risk scenario is that it has just been activated. The most recently activated risk scenarios, consequently, easily get over-emphasized when assessing the total risk.

The effect of this over-emphasis of the last activated risk scenario is reported in (Viscusi, Wesley et al. 1991): "*Individuals' perceptions of the risk levels to which they are exposed are likely to be greater: ..... (ii) for risks for which the unfavorable risk evidence is presented last even when there is no temporal order*".

In one of our own studies we found an example of the importance of cognitive accessibility in estimation strategy selection (Jørgensen 2009). In that study professional software developers estimated the same development projects. Before the estimation work started, the developers were randomly allocated either to tasks with questions related to "similarity" (e.g., "what is the city most similar to …?") or to tasks with questions related to "averages" (e.g., "what is the average height of …?"). All tasks were unrelated to the software development effort estimation task. Interestingly, those who had just activated similarity-based strategies were much more likely to use the same type of strategies (closest analogy-based instead of average-based strategy) in their subsequent estimation work. The total evidence is, as far as we can see, strongly in support of an over-emphasis of the most recently activated information or strategy, including the most recently activated risk scenario, see (Mussweiler 2003) for a review on the role of accessibility in judgments.

When identifying risks it is reasonable to assume that there will be a tendency towards starting with the risks that are most severe and likely to occur (we test this assumption in Experiment C) and to stop when no more relevant risks are possible to identify. This has the effect that the last identified risks typically may be less severe and harder to access than the first risks. This leads to two, possibly over-lapping, effects of more effort on risk identification. More effort on risk identification can lead to:

1) An increased accessibility of less severe risks since they are most recently activated and, consequently, to under-estimation of the total effect of the risks on the effort estimate. This seems to be the explanation used in (Viscusi, Wesley et al. 1991).
2) An increased hardness of accessing the last risk. This "hardness" may be used as an indicator of the total effect of the risks on the effort estimate and, consequently, leads to under-estimation of the effort. This is the explanation, as far as we interpret it, used in (Sanna and Schwarz 2004).

We find that both explanations are possible explanations of the empirical results identified.

2.4 The Peak-End Rule

The high importance of the last phase of a cognitive or emotional process as input to the assessment of the total experience is further illustrated by the findings reported in (Kahneman, Fredrickson et al. 1993). In that study people were subject to two trials. In the first trial they hold one hand in water at 14º C for 60 seconds. In the second they hold the hand in 14º C for 60 seconds and then kept the hand in the water for 30 additional seconds where the water temperature was gradually raised. A significant majority, when asked to repeat one of the trials, preferred the second trial, even though the total amount of pain was higher. This may show that the effect of the increased accessibility of the last part of the total experience in this experiment was stronger than the impact from the longer lasting pain.

The dominance of the strongest experience (the peak) and the final experience (the end) for the perceived total experience has been repeated in many domains, e.g., marketing (Do, Rupert et al. 2008), and is termed the peak-end rule. According to the peak-end rule the most severe and the last risk scenario will to a large extent determine the feeling of total risk. As long as the peak, e.g., the most likely or most severe reason to fail in the study described in (Sanna and Schwarz 2004), is the about the same, the last risk scenario will according to the peak-end rule dominate the feeling of risk, i.e., the assessment of total risk. The peak-end rule, we believe,

provides additional empirical support to elements of the explanations described in Section 2.3.

3. The Experiments

Based on the discussions and possible explanations described in Section 2 and practical experimental issues, the following elements were input to the study design:

- The software professionals participating in our study should be randomly divided into two groups. One group should use very little time on risk identification (Group LESS) and one group should use substantial more time (more than those in Group LESS) on risk identification (Group MORE). Notice that the absolute level of risk identification work is not essential in our studies, i.e., it is not essential to be able to define what MORE and LESS means in absolute terms, as long as those in the group more spends more effort on risk identification. We are, as stated in the hypothesis, only interested in the direction of the effect, not the size of it given a particular level of risk identification effort.
- It should be unlikely that non-trivial risk elements, with large impact on development effort and success probability, were discovered through more effort on risk identification. To achieve this we used requirement specifications that described relatively simple software systems with no hidden complexities.
- The estimation work should be completed immediately after the risk identification work. This condition enabled us to better study the effect of doing more risk identification, e.g., it would be less likely that there would be confounding factors disturbing the effect of increases risk accessibility. This design element does, however, mean that we need more studies to evaluate the size of effect when the distance in time between the risk identification and effort estimation increases, i.e., when the accessibility of the last identified risk has decreased. Possibly, the effect will decrease with more distance between risk identification and effort estimation.
- The groups of participants instructed to perform less or more risk identification should identify the same risk peak, i.e., identify the same most important risk. This is, we believe, important to avoid that differences in risk peaks masked the effect of more risk identification due to the peak-end rule. To achieve this we asked those with less risk identification effort to identify the most (or the three most) important risk(s).
- It should be clear for the participants that the risk identification was intended as input to effort estimation work only. If the participants believed that this would be the only risk management activities in the project, it would be possible to argue that the increase in risk identification work could lead to better plans and consequently lower use of effort and higher probability of success. We emphasized therefore in the instructions that the risk identification work was there to serve as input to the estimation work.
- The population of participants, the risk identification instructions and the software requirement specifications should vary from experiment to experiment to test the robustness of the results. The variations should, however, not be so large that the experiments would not belong to the same "family of experiments". Belonging to the same family of experiments means that we could use the combined effect of the four experiments to support each other, i.e., as non-identical replications of each others. Replicated non-significant results in varying contexts can be more reliable than highly statistical significant results in one study, see for example (Rosenthal 1978; Hallahan 1996).

- Based on previous experience in similar effort estimation contexts, we expected that there would be a large variation in the effort estimates. For this reason, we based the analysis of differences in effort estimates on the more outlier robust Kruskal-Wallis rank-based, non-parametric test.

The four experiments (A, B, C and D) were completed in four different countries. Experiment A was conducted with software developers from Ukraine, Experiment B with software developers from Vietnam, Experiment C with software developers from Poland, and, Experiment D with project managers from Norway. All software developers had at least 6 months professional experience, most of them significantly more. All of them had at least three years of university education and were assessed by their manager to have acceptable English skills for participation in an experiment with instructions and requirement specifications in English. The software development effort estimates of Experiments A and B were based on the same requirement specification (see Appendix A), while Experiment C was based on a different, slightly larger requirement specification (see Appendix B). Experiment D did not ask for an estimate of effort, but instead an assessment of the probability of success of a project organizing a conference. The study designs are similar, but not identical, regarding the risk identification instructions and the format of success assessment responses.

4. Experiment A

4.1 Design

Experiment A was conducted with 52 software developers from three Ukrainian software companies. The developers were paid for their participation and were instructed to conduct their estimation work as realistically as possible. The estimation work typically took 10-20 minutes, which is less than typical estimation work, even for projects as small as in this estimation work. We discuss the consequences of this and other limitations of the experiments in Section 9.1.

The 52 developers were randomly divided into two groups; LESS (little effort risk identification) and MORE (more effort on risk identification). The groups received the following instructions:

- LESS: *Assume that you are asked by a client to estimate and develop the web-based system described on the next page (SeminarWeb). You are allowed to choose the development platform (programming language etc.) you want. As a start of the estimation work, please read the description of SeminarWeb and make a quick risk analysis, where you briefly describe the three most important risk factors (things that may go wrong) of this project. Risk factors are situations, events and conditions potentially leading to development problems, e.g., factors potentially leading to estimation errors, technical problems, low quality of the software and dissatisfied client. Once you have completed the risk analysis, please estimate the effort you most likely would need to develop SeminarWeb and fill in the answer on the bottom of this page.*
- MORE: *Assume that you are asked by a client to estimate and develop the web-based system described on the next page (SeminarWeb). You are allowed to choose the development platform (programming language etc.) you want. As a start of the estimation work, you are asked to conduct a risk analysis and describe the most important risk factors (things that may go wrong) of this project. Risk factors are situations, events and conditions potentially leading to development problems, e.g., factors potentially leading to estimation errors, technical problems, low quality of the software and dissatisfied client. Use the back side of*

this page if you need more space to describe the risk factors. Once you have completed the risk analysis, please estimate the effort you most likely would need to develop SeminarWeb and fill in the answer on the bottom of this page.

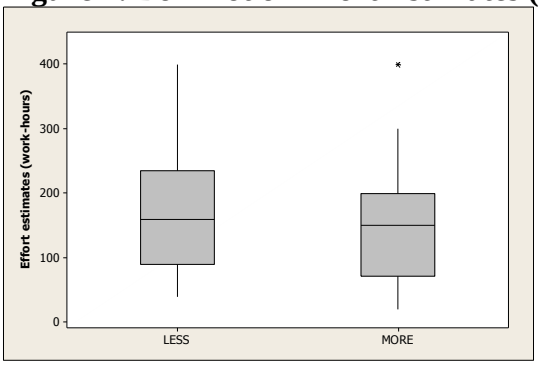
On the bottom of the instruction page all participants were asked to: i) Estimate the number of work-hours they would most likely need to develop and test the SeminarWeb system, ii) Assess the likelihood of project success (defined as less than 25% estimation error and a client satisfied with the quality and the user friendliness of the system), and iii) Assess his/her competence in developing SeminarWeb on the scale: *Very good - Good – Acceptable – Poor*. Our definition of project success is, of course, not meant to be a complete definition of project success. It is only meant as an operational definition serving the purpose of testing the hypothesis formulated in Section 1.

4.2 Results

The simple software specification and the option to select familiar development technology meant, as far as we could see, that more risk identification was unlikely to lead to discovery of essential new risks (no new risk peaks). An informal inspection and categorization of the risks provided by the developers supported the assumption that more effort on risk identification hardly ever led to the discovery of essential new risks. The dominant risks (the risk peak) for both Group MORE and LESS were related to missing/unclear/misunderstood/changing requirements, general concern about potential technical problems, and personnel problems (such as sickness). In the Groups LESS and MORE, the dominant risks were, as expected, typically described first in the list of risks.

Those in Group MORE were expected to provide more risks ("the most important") than those in Group LESS ("the three most important"). Unfortunately, although they seemed to spend more effort on the risk identification, they did not provide many more risks. The mean number of risks in Group LESS was 2.5 and in Group MORE 3.0. The main reasons for this low difference in number of risks may be that the SeminarWeb project is quite simple, and the software developers had problems with or was not sufficiently motivated to find many risks. Even the rather small difference in number of identified risks, however, was connected with a difference in median effort estimates in the expected direction, see Figure 1.

Figure 1: Box Plot of Effort Estimates (Experiment A)

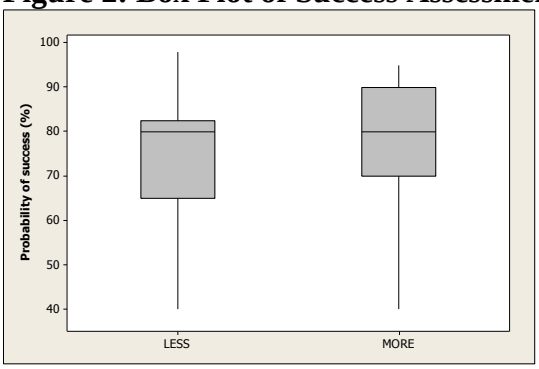


The median effort estimate of those in Group LESS was 160 work-hours and in Group MORE 150 work-hours, i.e., increased effort on risk identification was connected with lower effort estimates. The variation in effort estimates within the

groups is large, which leads to a low statistical power of the study is low and a non-significant difference. A Kruskal-Wallis, one-sided test of differences in mean ranks gives p=0.3.

The assessed project success probability was also in the hypothesized direction although the median values were the same (80%), see Figure 2. A Kruskal-Wallis, one-sided test of differences in mean ranks gives p=0.10. More risk identification, consequently, was connected with higher confidence in project success.

Figure 2: Box Plot of Success Assessment (Experiment A)



The self-assessed competence in developing SeminarWeb was completed after the risk identification and estimation work and could potentially be impacted by the difference in amount of risk identification. An analysis gave that more risk identification was connected with the perception of higher competence. As an illustration, while 64% of those in Group LESS categorized their competence as "very good" or "good", the corresponding proportion of those in Group MORE was 74%. Before the experiment started, the participants had completed a questionnaire where they should assess their general competence as software developers along the same scale (Very good – good – acceptable – poor). The proportion of developers in the categories "very good" and "good" were at that stage almost the same, i.e., 72% in Group LESS and 69% in Group MORE. It is therefore likely that the increase in assessed competence from Group LESS to Group MORE was caused by the added effort on risk identification. Self-assessed competence can be seen as an alternative measure of project success confidence and, consequently, as a further support of the observed increase in confidence with more risk identification. This result is also possible to interpret as an observation of increased feeling of control with more risk identification work and, perhaps, an observation connected to the illusion-of-control phenomenon.

5. Experiment B

5.1 Design

The population of participants in Experiment B consists of 49 software developers from two Vietnamese companies. The experiment replicates the study design from Experiment A, but changed the risk identification instructions slightly to increase the difference in amount of effort spent on risk identification between Groups LESS and MORE.

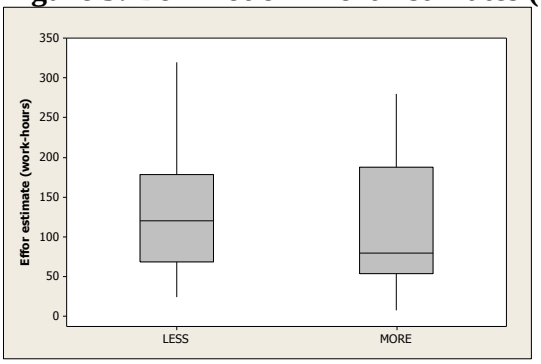
In Experiment B we instructed those in Group LESS to provide "the most important risk factor" (in Experiment A we asked for the three most important risk factors). Those in Group MORE were instructed to spend at least 5 minutes to think back on previous similar development tasks, analyze what went wrong when

completing those tasks, and then spend at least 5 minutes to identify what potentially could go wrong in this project (SeminarWeb), i.e., to use a more extensive risk identification process than the Group MORE participants in Experiment A. Group MORE also got a short checklist of potential risk factors to support their risk identification. This change in study design led to larger differences in number or risk factors identified. The mean number of risk factors of those in Groups LESS and MORE was 1.0 and 3.4, respectively.

5.2 Results

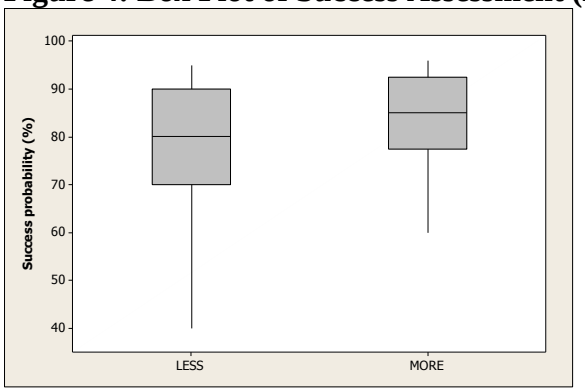
The results of Experiment B were in the same direction as those in Experiment A, see Figure 3. Those in Groups LESS and MORE had median effort estimates of 120 work-hours, and 80 work-hours, respectively. A Kruskal-Wallis, one-sided test of differences in mean ranks gives p=0.2.

Figure 3: Box Plot of Effort Estimates (Experiment B)



The success assessment differences between the groups were also in the hypothesized direction, see Figure 4. The median values of the Groups LESS and MORE were 80% and 85%, respectively. A Kruskal-Wallis, one-sided test of differences in mean ranks gives p=0.2.

Figure 4: Box Plot of Success Assessment (Experiment B)



In Experiment A, 64% in Group LESS and 74% in Group MORE assessed their own competence in development of SeminarWeb as “very good” or “good”, i.e., an increase in level of risk identification was connected with an increase in self-assessed competence. In Experiment B, the corresponding proportions were: 60% in Group LESS, and 79% in Group MORE. Both experiments therefore observe an increase in self-assessed competence from Group LESS to MORE, i.e., further support of our hypothesis of increased confidence with more risk identification.

6. Experiment C

6.1 Design

The population of participants in Experiment C consists of 50 software developers from a Polish company. This experiment replicates the study design from Experiment B, with some changes. It uses a different requirement specification ("Database of Empirical Studies", see Appendix B), includes some elements of risk analysis (not only risk identification, but also risk probability and severity assessment) for those in Group MORE, and uses a different project success definition.

The motivation behind the change in risk identification process was to enable a test of whether inclusion of a simple analysis of risk probability and consequences would affect the results. The extended risk identification was completed in a table on the format: 1) Description of risk, 2) Probability of occurrence (low, medium or high), 3) Severity (low, medium, high).

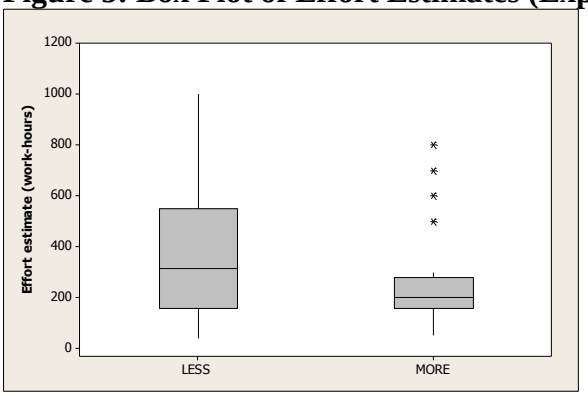
The success assessment question was changed from an assessment of the probability to achieve "less than 25% error and a satisfied client" in Experiments A and B, to an assessment of the probability to experience "more than 25% effort overrun".

The mean number of risk factors of those in Groups LESS and MORE in Experiment C was 1.0 and 4.1, respectively.

6.2 Results

The results were similar to those in Experiments A and B, see Figure 5. Those in Group LESS had a median effort estimate of 316 and those in Group MORE 200 work-hours. As before, more work on risk identification resulted in lower effort estimates. A Kruskal-Wallis, one-sided test of differences in mean ranks gives p=0.09.

Figure 5: Box Plot of Effort Estimates (Experiment C)

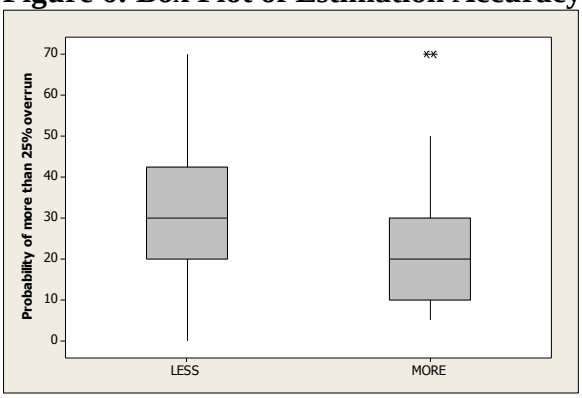


In a previous study, see (Jørgensen and Carelius 2004), four companies, independent from each other, implemented the application based on essentially the same requirement specification as the one we presented to the participants in Experiment C. The main difference was the removal of background information, the motivation of the project and some illustrating examples on how to understand the requirements. This information was removed to speed up the reading phase in our experiment. The median actual completion time of the four companies completing the development work of the Database of Empirical Studies-system was about 700 work-hours. The actual use of effort to develop the software was consequently much higher

than the estimated median of 200 or 316 work-hours of the participants in our experiment. This suggests that the developers in our experiment on average had a strong bias towards over-optimism, i.e., that lower effort means more optimism.

The assessed probability of exceeding the estimate with more than 25% was lower (indicates a higher confidence in project success) among those with more risk identification, see Figure 6. The median values of the Groups LESS and MORE were 30% and 20%, respectively. A Kruskal-Wallis, one-sided test of differences in mean ranks gives p=0.05. Three out of the four companies that had implemented the Database of Empirical Studies-system had estimation errors (overruns) much larger than 25%. Together with the observation that the estimates were typically much lower than the actual use of effort, this suggests an over-confidence in project success among the both groups of software developers in Experiment C.

Figure 6: Box Plot of Estimation Accuracy Confidence (Experiment C)



In Experiments A and B, those in Group MORE perceived themselves as more competent than those in Group LESS. The same was the case in Experiment C. While 68% of those in Group LESS perceived themselves as "Very good" or "Good", the corresponding proportion among those in Group MORE was 76%. All three experiments, consequently, gave that the perceived skill of those in Group MORE was higher than that in Group LESS. This suggests that more risk identification had the side effect that the developers felt more competent. To what extent this is caused by a perceived increase in skill or by a perceived decrease in complexity of development work is not possible to decide from Experiments A, B and C.

The design of Experiment C enabled us to test one essential assumption of the explanations described in Section 2.3 and 2.4, i.e., the assumption that the software professionals start with the most important risk and that the risk identified last is less severe and/or probable to occur. The distribution of risk probabilities and severities of the first and last risks of those in Group MORE is described in Table 1. It shows for example that there were 15 responses where the first risk and only 3 responses where the last risk had a high severity. Two developers provided only one risk, so the sum of last risk values is 23. As can be seen, Table 1 provides strong support for the assumption that the developers start with the identification of the most important risks.

Table 1: Risk probability and Severity in Group MORE

Value	First risk		Last risk	
	Probability	**Severity**	**Probability**	**Severity**
High	10	15	3	3
Medium	10	10	6	13
Low	5	0	15	8

Another assumption we made was that the developers in Group LESS and MORE would have the same "peak", i.e., that they would identify about the same risks as the most important. The three most important risks in both groups were i) Integration between the system to be developed and the organization's existing database (this risk turned out to be a major reason for effort overrun in the actual implementation of the system), ii) Changing requirements, iii) Requirements related to database extensions. These three risks covered 19 out the 25 first risks in Group MORE and 25 out of 25 of the most important risks in Group LESS. This provides strong support for the assumption of similar risk peaks in the two groups of developers.

7. Experiment D

7.1 Design

We wanted to test our hypothesis in a situation where the success assessment was not about the developers own work, the estimator had no opportunity to impact the outcome and where the participants were project managers. For this purpose we designed Experiment D.

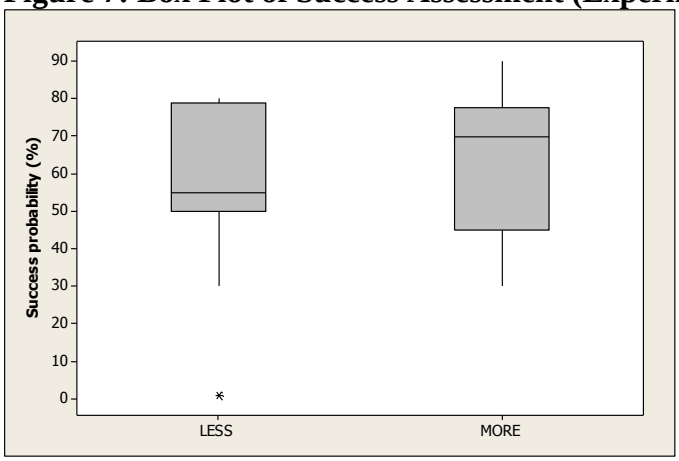
The population of participants in Experiment D consists of 29 project managers participating on a conference on project management. Most of them were senior project managers with extensive experience. They were randomly divided into two groups: LESS and MORE. Both groups received the following information: "*Assume that* [Name of the project management conference] *will be organized next year and that success is defined as: i) minimum 200 participants, ii) at least 90% of the participants evaluate their benefit from the conference to be "acceptable", "good" or "very good".*" Then, those in Group LESS were asked to provide three risks that could prevent a successful conference and those in Group MORE to provide as many risks as they could think of in five minutes. Finally, participants in both groups were asked to assess the probability that the next year's conference would be a success.

7.2 Results

Those in Group LESS had a mean of 2.9 risk factors and those in Group MORE had a mean of 7.9 risk factors. The median estimated probability of a success was 55% among those in Group LESS and in as high as 70% among those in Group MORE, see Figure 7. A Kruskal-Wallis, one-sided test of differences in mean ranks gives p=0.3. The number of participants of the conference the year they participated (the current conference) was lower than 200, i.e., lower than the minimum to call it a success next year. This indicates that both groups were over-confident in their success assessment, and that the increase in success probability with more risk identification indicates an increase in over-confidence.

Although the difference in success probability is not statistically significant, it is in the hypothesized direction and, because it can be considered as a non-identical replication of the findings in Experiments A-C, provides further support of our hypothesis, see for example (Rosenthal 1978; Hallahan 1996).

Figure 7: Box Plot of Success Assessment (Experiment D)



8. Summary of Results

Tables 2 and 3 summarize the effort estimates and success probabilities from the family of the four experiments (n.r. means that a value is not relevant for the experiment).

Table 2: Summary of the Four Experiments: Effort Estimates

Exp	Groups	Specification	LESS	MORE
A	LESS (up to 3 risk factors) MORE (as many risks as possible)	SeminarWeb	160	150
B	LESS (one risk factor) MORE (looking back + as many risks as possible)	SeminarWeb	120	80
C	MORE (looking back + as many risks as possible + simple risk analysis)	Database of Empirical Studies	316	200
D	LESS (up to 3 risk factors) MORE (as many risks as possible)	Organize a conference	n.r.	n.r.

Table 3: Summary of the Four Experiments: Success Probability

Exp	Groups	Specification	LESS	MORE
A	LESS (up to 3 risk factors) MORE (as many risks as possible)	SeminarWeb	80%	80%
B	LESS (one risk factor) MORE (looking back + as many risks as possible)	SeminarWeb	80%	85%
C	LESS (one risk factor) MORE (looking back + as many risks as possible + simple risk analysis)	Database of Empirical Studies	70%[1]	80%[1]
D	LESS (up to 3 risk factors)	Organize a	55%	70%

	MORE (as many risks as possible)	conference		

1: This value is calculated as 100% minus the assessed probability of exceeding the estimate with 25% or more.

As can be seen from Tables 2 and 3, all four experiments support the hypothesis of more optimism with more effort on risk identification (Group MORE compared with Group LESS) in the studied contexts.

Notice that the effort estimate and success probability assessment are interconnected values. This interconnection, caused by the way we define project success, would have been problematic when interpreting the results if more risk identification had led to higher effort estimates. Higher effort estimates are for example likely to lead to higher likelihood of not overrunning the effort estimate with more than 25%, which was part of our project success definition in Experiment C. Since the opposite was observed, i.e., more risk identification led to lower effort estimates, this interconnection instead strengthen the support of our hypothesis. Lower estimates should, rationally speaking, lead to lower confidence in project success, i.e., the opposite to what actually was observed. The actual differences in over-confidence between the groups are therefore, we argue, larger rather than the reported values.

We believe that our results, together with previous results, particularly that of (Sanna and Schwarz 2004), demonstrates that there are contexts were more effort on risk identification have counter-intuitive and unwanted effects, i.e., is likely to lead to more over-optimism and more over-confidence. Consequently, we should look for processes that combine risk identification and effort estimation that avoid these effects.

9. Discussion

9.1 Limitations

The risk identification and estimation contexts of our experiments are different from those usually experienced by software developers. The software developers knew for example that the project would not be started and they had a high time pressure for the completion of the risk identification and estimation work. There is, for this reason, a need for more studies to investigate the effect of more risk identification in other contexts, e.g., situations with more time available, where the effort estimating do not follows immediately after the risk identification and where the estimates are based on team work. The above limitation does, however, not imply that the experimental results are without relevance. Many real-life situations are similar to the one studied, e.g., based on project meetings with developers with high pressure to produce a high number of activity estimates in a short period of time. Besides, the results are in our opinion useful as input to a better understanding of the processes involved in judgment-based effort estimation.

We have only studied a few, quite informal, variants of risk identification processes and only estimation work related to relatively small software applications. Although the observed effect of more risk identification is supported by results from other domains, we should be careful with generalization to other types of risk related processes and other types of software projects. More rigorous risk analysis and management processes including the development of plans for risk management may for example remove or strongly reduce the effect.

The participants in Experiments A, B and C were paid and instructed by us and by their manager to work as similar to ordinary work as possible. We do, however, not know much about how motivated they were to do their best. If the level

of motivation had been higher, those in the MORE group may for example have been able to identify more risks. This, we believe, would have led to even larger difference in effort estimates. In other words, it is possible that a higher motivation in identification of risks would lead to stronger rather than weaker effects than the one we observed.

Many of the software developers in Experiments A, B and C had never been responsible for planning a project, only to provide estimates for their own work. This means that we should be careful about generalizing the results to experienced project managers. The results in Experiment D suggest, however, that that the effect is present there, as well.

As can be seen, there are several limitations of the experiments which should lead to careful interpretation of the results. The fact that we found the same effect in four experiments with populations from different countries and variation in instructions and requirements specifications means, however, that it is unlikely that the effect happened by accident. The main limitation is, in our opinion, the generality of the results. Our results may mainly be valid in contexts where: i) More effort on risk identification does not lead to significant new insight, ii) There is a high time pressure, i.e., only a short time available to identify the risks and estimate the effort, and, iii) The estimation work is completed immediately after the risk identification work.

9.2 Evaluation of Possible Explanations

Earlier, in Section 2, we proposed one enabler (judgment-based processes) and several, partly over-lapping, explanations (illusion-of-control, increased "hardness" of identifying the last risk, increased accessibility of the last risk, and the peak-end rule) of more optimism and confidence as a result of more risk identification. We find it hard, based on our experiments and previous work, to exclude any of the above explanations. The finding in Experiment C showing that the risk identified first was much more severe than the risk identified last, provides support for the explanation based of the increased activation of the last risk identified. It does, however, not exclude the accessibility "hardness" or the illusion of control explanations. It may be the case that all explanations are valid and that the context determines which of them that will have the strongest effect. Studies in different contexts, better designed to separate these explanations are therefore needed.

The perhaps most robust insight from our four experiments may be that the feeling-of-risk, regardless of determined by increased feeling of control, an over-weighting of characteristics of the last identified risk factor or some other explanation, is able to affect the estimation of software development effort and the assessment of probability of project success. This feeling-of-risk is, as discussed in Section 2, hardly based on a formally correct "sum" of risks and can be impacted by many effort and success-irrelevant factors.

9.3 Practical Consequences

Assuming that previous results on this topic, e.g., (Viscusi, Wesley et al. 1991; Sanna and Schwarz 2004), and the results from our own four experiments are trustworthy and relevant in software project contexts, what should be the consequences for software project estimation practice? Clearly, thorough work on risk identification is essential for good project management and should be included as mandatory practice in most software projects.

We believe that the following sequence of risk identification and effort estimation would reduce the unwanted effects of more work on risk identification:

1) Estimate the "nominal" effort (e.g., the most likely effort assuming normal productivity and no substantial problems).
2) Identify risks and their expected impact on development effort.
3) Add the expected impact of each of the identified risks on the use of effort.
4) Add effort (contingency buffer) for not identified (currently unknown) risks.

All the above steps should be supported with historical data and checklists, whenever possible. The more negative, uncertain events (risks) identified in Step 2), the higher the estimated effort will be when following this process, i.e., the process avoids that the feeling of risk is used as input to the effort estimate.

10. Summary

All four experiments provide empirical support of the hypothesis that there are contexts where more work on risk identification leads to lower effort estimates and higher confidence in project success. This may contribute to the typical over-optimism and over-confidence in software development effort estimates. The contexts studied in our experiments are characterized by high time pressure, risk identification completed immediately before the estimation work not leading to substantial new insight in the project complexities. We do therefore not know to what extent our results will hold in other contexts.

There are several possible explanations for our observations. All explanations share the assumption that the effort estimation is judgment-based and impacted by the feeling of risk following work on risk identification. This feeling of risk may, amongst others, be impacted by an increased feeling of control with more risks identified, how difficult it was to identify the last risk, or, an over-weighting of the last activated, most accessible risk. In all cases, more effort on risk identification can lead to more over-optimism and over-confidence. A supporting explanation is the so-called peak-end rule, i.e., the observation that the peak and the end experience are the most important when assessing the total experience. Given the same peak, the end experience, e.g., the last identified risk scenario, dominates the difference in assessment of the total experience, e.g., the feeling of risk. More work on risk identification will typically lead to lower severity of the last identified risk. As a consequence, the total risk will be assessed to be less severe as long as the peak remains constant.

We recommend the following estimation approach to reduce the over-optimism and over-confidence: i) Estimate the "nominal" effort (the most likely effort assuming normal productivity and no particular problems), ii) Identify risks with impact on development effort, iii) Add the assessed impact of each of these risks to the effort estimate, iv) Add the effort (contingency buffer) for not identified (unknown) risks. We believe that this process, which we have observed in use by some experienced project managers, is likely to lead to less over-optimism and over-confidence, compared with processes where the risk identification is completed immediately before and used as input to the estimation work. We are in the process of conducting studies that empirically study the effect of this recommended process change.

References:

Babad, E. and Y. Katz (1991). "Wishful Thinking—Against All Odds." Journal of Applied Social Psychology **21**(23): 1921-1938.

Buehler, R., D. Griffin, et al. (1997). "The role of motivated reasoning in optimistic time predictions." Personality and Social Psychology Bulletin **23**(3): 238-247.

de Venter, G. and D. Michayluk (2008). "An insight into overconfidence in the forecasting abilities of financial advisors." Australian Journal of Management **32**(3): 545-557.

Do, A. M., V. Rupert, et al. (2008). "Evaluations of pleasurable experiences: The peak–end rule." Psychonomic Bulletin & Review **15**(1): 96-98.

Evans, J. S. t. (2003). "In two minds: dual-process accounts of reasoning." Trends in cognitive sciences **7**(10): 454-459.

Hallahan, M. R., R (1996). "Statistical power: Concepts, procedures, and applicationsN* and ROBERT ROSENTHA." Behaviour Research and Therapy **34**(5-6): 489-499.

Hammond, K. R., R. M. Hamm, et al. (1987). "Direct comparison of the efficacy of intuitive and analytical cognition in expert judgment." IEEE Transactions on Systems, Man, and Cybernetics **17**(5): 753-770.

Henry, R. A. (1994). "The effects of choice and incentives on the overestimation of future performance." Organizational Behaviour and Human Decision Processes **57**(2): 210-225.

Jørgensen, M. (2009). "Selection of Work-effort Estimation Strategies." A preliminary version can be downloaded from simula.no/research/engineering/publications/Simula.SE.235.

Jørgensen, M. and G. Carelius (2004). "An empirical study of software project bidding." IEEE Transactions on Software Engineering **30**(12): 953-969.

Jørgensen, M. and T. M. Gruschke (2005). Industrial Use of Formal Software Cost Estimation Models: Expert Estimation in Disguise? Empirical Assessment of Software Engineering (EASE), Keele, England, Keele University.

Jørgensen, M., K. H. Teigen, et al. (2004). "Better sure than safe? Over-confidence in judgement based software development effort prediction intervals." Journal of Systems and Software **70**(1-2): 79-93.

Kahneman, D., B. L. Fredrickson, et al. (1993). "When more pain is preferred to less: Adding a better end." Psychological Science **4**(6): 401-405.

Langer, E. J. (1975). "The Illusion of Control." Journal of Personality and Social Psychology **32**(2): 311-328.

Milburn, M. A. (1978). "Sources of bias in the prediction of future events." Organizational Behaviour and Human Performance **21**(1): 17-26.

Moløkken, K. and M. Jørgensen (2003). A review of software surveys on software effort estimation. International Symposium on Empirical Software Engineering, Rome, Italy, Simula Res. Lab. Lysaker Norway.

Mussweiler, T. (2003). "Comparison Processes in Social Judgment: Mechanisms and Consequences." Psychological Review **110**(3): 472-489.

Otway, H. and D. von Winterfeldt (1992). "Expert judgment in risk analysis and management: Process, context, and pitfalls." Risk Analysis **12**(1): 83 - 93.

Pelham, B. W. and E. Neter (1995). "The effect of motivation of judgment depends on the difficulty of the judgment." Journal of Personality and Social Psychology **68**(4): 581-594.

PMI (2008). A guide to the project management body of knowledge (PMBOK guide). Maryland, USA, Project Management Institute.

Rosenthal, R. (1978). "Combining results of independent studies." Psychological bulleting **85**: 85-193.

Sanna, L. J. and N. Schwarz (2004). "Integrating temporal biases: The interplay of focal thoughts and accessibility experiences." Psychological science **15**(7): 474-481.

Simon, H. A. (1957). Models of Man, Social and Rational: Mathematical Essays on Rational Human Behavior in a Social Setting. New York, Wiley.

Slovic, P., M. L. Finucance, et al. (2007). "The affect heuristic." European journal of operational research **177**: 1333-1352.

Thompson, S. C., W. Armstrong, et al. (1998). "Illusions of control, underestimations, and accuracy: A control heuristic explanation." Psychological Bulletin **123**(2): 143-161.

Tversky, A. and D. Kahneman (1974). "Judgment under uncertainty: Heuristics and biases." Science **185**: 1124-1131.

Viscusi, W. K., A. M. Wesley, et al. (1991). "Communication of ambiguous risk information " Theory and decisions **31**(2-3): 159-173.

White, D. and J. Fortune (2002). "Current practice in project management — an empirical study." International Journal of Project Management **20**(1): 1-11.

Appendix 1: Description of SeminarWeb

The company XXX organizes several seminars for software developers each year. Presently, the seminar administrator manually sends email invitation to potential participants and the participants register for the seminars by sending a responding email to the administrator. The company now wants to develop a web-based system (SeminarWeb) to administrate invitations and register participants. The process to be supported by SeminarWeb is as follows:

1. The administrator of SeminarWeb inputs person information (name, employer, address, email) of potential participants through a web-interface. This information should be stored in a database. There will typically be information about 100-500 persons in this database.
2. The administrator use a web-interface to register and store the following information about a seminar: The name of the seminar, the date of the seminar, a text that informs briefly about the seminar and how to register (this text – max 50 words - will be written by the administrator) and a link to a pdf-document with the full description of the seminar.
3. The administrator sends out the invitation through the following web-interface supported steps: a) Select a prepared invitation (by selecting the name of the seminar), b) Display all persons stored in the database (no search facilities are needed), c) Select the persons that should be invited to the seminar, d) Select "Execute invitation". When selecting "Execute invitation" the system should send the invitation (the brief text in the email and the pdf-document as attachment) by email to all the selected persons. In this first version of SeminarWeb there is no need for procedures to manage failed emails.
4. An invited person should be able to register for a seminar by submitting his or her name, email address and the name of the seminar through a web-interface. The system should store the registered information. Persons not invited should not be able to register. When a person has registered, he or she should get a confirmation by email that the registration has been received.
5. The administrator should be able to monitor and print out the registered participants of a seminar through a web-interface.

Appendix 2: Description of Database of Empirical Studies (DES)

The researchers at Simula conduct many empirical studies (controlled experiments, case studies, surveys, etc). Simula now wants to build a dynamic website with information about their empirical studies. The website should enable an efficient support for management, retrieval and reporting of studies conducted at Simula. The system should handle at least 50 concurrent users. Integration with Simula's publication database and Simula's people database (these databases are implemented using MySQL, Java and Apache Tomcat and have a JDBC interface). The system should be developed from scratch.

The website should have three types of users (guests, registered users and administrators). Administrators have access to the functionality that is available to registered users, and registered users have access to the functionality that is available to guests.

The **user stories** that should be implemented are the following:

U1: As an administrator, I want to add and remove registered users. I want to grant administrator privileges to registered users.

U2: As a registered user, I want to change my password.

U3: As a registered user, I want to register new studies, so that they will be available for other researchers. For each study, I want to register:

a) The name of the study,
b) The type of study (experiment, case study, survey, etc). I want to select type of study from a predefined list,
c) The people responsible for the study. The list of available persons should be retrieved from Simula's people database,
d) A short description of the study. I should be able to use rich text formatting,
e) Publications. The list of available publications should be retrieved from Simula's publication database,
f) Study start date and study end date. I want to select dates from "Select a calendar",
g) Documents (raw data in excel, design documents, etc). I want to upload the documents (if any) from my PC.

U4: As a registered user, I want to be able to edit and delete the studies I have registered.

U5: As an administrator, I want to add new and update existing fields for studies (free text and predefined lists), so that I can support future needs.

U6: As a guest, I want to search for studies. I want to search by the name of the study, type of study, the year the study ended or through a free text search (searches in all fields and in filenames of documents, but not inside documents). I want to see the study name, type of study, the year the study ended, study responsible(s) for each study that matches my search criteria.

U7: As a guest, I want the name of the study to be displayed as a link to a page that displays all the study data (including downloadable documents, if any). I want the study responsible to be displayed as a link to a new search that displays all the person's studies.

U8: As a guest, I want to be able to sort the search result by any field.

U9: As a guest, I want to see aggregated study information (number studies of different types per year) graphically. I want to be able to specify the period for the aggregation (e.g., number of empirical studies completed per year in 2003-2008) and the types of studies to be included in the aggregation (e.g. case studies and surveys).

U10: As a guest, I want to export search results and/or study data to a comma-separated file that I can be store locally on my PC, so that I can import the data in other applications.