The Impact of Irrelevant Information on Estimates of Software Development Effort

Stein Grimstad[1,2]

Magne Jørgensen[1]

[1]Simula Research Laboratory

[2]University of Oslo

Abstract

Software professionals typically estimate software development effort based on a requirement specification. Parts of this specification frequently contain information that is irrelevant to the estimation of the actual effort involved in the development of software. We hypothesize that effort-irrelevant information sometimes has a strong impact on effort estimates. To test this hypothesis, we conducted two controlled experiments with software professionals. In each of the experiments, the software professionals received

specifications describing the same requirements. However, we gave one group of the software professionals a version of the requirement specification where we had included additional, effort-irrelevant, information. In both experiments we observed that the estimates of most likely effort increased when the estimates were based on requirement specifications that contained the information irrelevant to development effort. The results suggest that when estimation-irrelevant information is included as input to expert judgment-based estimation processes, the estimators find it difficult to distinguish between the estimation-relevant and the estimation-irrelevant information. A possible consequence of our findings is that estimation-irrelevant information should be removed from the requirement specification prior to the use of it as input to estimation work.

Keywords: Cost Estimation, Irrelevant information, Expert judgment.

1 Introduction

Software projects frequently overrun their effort estimates [1]. This is a major concern for the software industry, because the quality of software effort estimates directly affects companies' ability to compete. Poor estimation performance often causes budget overruns, delays, lost contracts and low-quality software.

A recent review [2] summarizes findings suggesting that expert judgment-based estimation is the most popular estimation method in the software industry. Typically, studies report that 70-80% of industrial estimates are made by experts without using formal estimation models. The review summarizes studies of expert judgment- and

model-based effort estimates and concludes that the evidence does not support a replacement of expert judgment with estimation models. Although there are studies that have identified factors that affect the judgment-based effort estimates [2, 3], our understanding of the steps and biases involved in expert estimation is limited [4]. The popularity of the method, and the lack of knowledge about it, indicates that a better understanding of expert estimation may be required to meet the software industry's demand for more accurate effort estimates.

There are many factors that are relevant to the effort of software development [5, 6], e.g., amount of functionality, focus on cost control in the project and implementation technology. In an ideal world, we would like the estimate to be based on only relevant factors and not be affected by information that has no relation to the actual effort. Information about the choice of GUI colors in a web system should, for example, not affect the estimate of the effort required to develop a new order engine. Neither should the font size and margins of a requirement specification affect the estimate. However, an unpublished experiment conducted by the second author of this paper on computer science students found that this could be the case! In that experiment, half of the students estimated development effort based on a short requirement specification, and the other half estimated based on a long specification. The text in the two specifications was identical, but line-spacing, page set up and font size were adjusted so that the long version of the specification was seven pages and the short version only one page long. The students exposed to the long version provided on average 16% higher effort estimates. This effect caused by irrelevant information is consistent with research in other fields [7-11]. Hristova et al. [10], for example, report that the colour of the text influenced

price judgments, and, Gaeth and Shanteau [11] report that experienced soil judges are influenced by irrelevant factors in soil judgment.

Software effort estimates are often based on requirement specifications where the information varies in precision, structure and relevance. It is rare that all the information in a requirement specification is relevant for estimating software development effort. Estimation-irrelevant information is included in requirement specifications for a number of reasons, such as the following: insufficient time is spent on removing information of less relevance (e.g., the text is copied from a previous specification), the author lacks knowledge of what to include in a requirement specification, the information is useful for purposes other than software effort estimation. This study investigates empirically whether the presence of information that is irrelevant to estimation of software effort affects software professionals' effort estimates. The research question is as follows:

RQ: Are software professionals' estimates of most likely effort affected by estimation-irrelevant information in the requirement specifications?

By the term "estimation-irrelevant information" we mean information that does not have a direct or indirect *casual* relationship to software development cost. Notice that, as we interpret it in this paper, information can be estimation-irrelevant even if it has a *correlation* to actual effort. (Correlations may indicate a causal relationship at a deeper level, but do not themselves constitute a causal relationship.) For example, the length of the requirement specification may correlate with development effort. We would categorize the length of the requirement specification as effort estimation *relevant* if a

difference in length is caused by differences in the amount of development effort demanding requirements. Length or the requirement specification would, on the other hand, be categorized as *irrelevant* information if a difference in length is caused by differences in text formatting or by inclusion of information that has nothing to do with the development of the software.

Our hypothesis was tested in two controlled experiments. In both experiments, half of the software professionals estimated a software development task based on a requirement specification where we had introduced estimation-irrelevant information, and the other half estimated effort based on the same specification, with the estimation-irrelevant information removed.

The remainder of the paper is organized as follows. Section 2 describes related work on the effect of irrelevant information. The experiments are presented in Sections 3 and 4. Section 5 discusses the results. Section 6 summarizes the paper and provides recommendations.

2 Related work

In order to find software cost estimation studies related to our research question, we searched the BESTweb database[1] for studies that investigate empirically the impact of irrelevant information on software cost estimates. BESTweb is an online library of estimation papers that claims to include nearly all journal papers and many of the conference papers on software cost estimation. The selection of papers included in BESTweb are described in [12]. At the time of the review, the BESTweb library

[1] available at http:///www.simula.no/BESTweb

contained 964 estimation-relevant articles. We also included a recent study that we are aware of, not included in the BESTweb database. The following studies on the impact from irrelevant information in software engineering contexts were identified:

- Jørgensen and Sjøberg [13] report that preplanning effort estimates can have a major impact on detailed planning effort estimates, even when the estimators are told that the early estimates are not based on historical data or expert knowledge, i.e., should not be considered as relevant information to the estimation process. The estimators' awareness of the impact of the irrelevant information was low.
- Jørgensen and Sjøberg [14] report that information about the customer's expectations can significantly affect most-likely estimates of software development, even when the subjects are explicitly told that the customer's expectation is not an indicator of the actual effort. The estimators did not notice this effect or assessed it to be low.
- Aranda and Easterbrook [15] report that including customer expectations of cost, which are clearly marked as irrelevant in the requirement specification, can have a large impact on cost estimates. The impact could not be explained by the subjects' estimation experience.

In these studies, the irrelevant information is presented to the subjects as some sort of initial estimate that is irrelevant to the subjects' estimation process. This special case of irrelevant information is typically termed "anchoring". The impact of anchors is strong, and has been demonstrated in many domains [16]. However, the estimates may, as reported by research in other fields [7, 17-19], be affected severely by other types of

irrelevant information. These studies have shown that the introduction of irrelevant information can lead to increased estimation error, reduced estimation reliability, less learning, and increased over-confidence in one's own estimates. There is also evidence that personal characteristics, such as domain expertise [20], attention ability [21] and handedness [22], can make an estimator more likely to be subject to the effects of irrelevant information.

Results from other research fields should be considered with some care, because most of the studies are conducted in contexts that differ from software effort estimation. Hence, we need to carry out studies in contexts similar to those met by software professionals estimating development effort.

Consequently, the main contributions of this paper are these: 1) to study the effect of irrelevant information on judgment in a software development effort context, and 2) to investigate empirically the impact of textual estimation irrelevant information, i.e., the effect of non-numerical irrelevant information on effort estimates. Textual irrelevant information may be more common in software development effort estimation contexts than irrelevant numerical anchors, but we have been unable to find previous studies on this topic. We have found no other study on the impact of textual irrelevant information in software development effort estimation contexts.

3 Experiment 1

Experiment 1 was designed to test two issues: 1) the impact of irrelevant information on the effort estimates, and 2) the impact of level of specification precision on the estimates.

This paper focuses on only the first issue, i.e., the impact of estimation-irrelevant information. For this reason, the impact of the level of precision will only be discussed related to the possibility of interaction effects between precision and irrelevant information. Section 3.1 describes and discusses the design of the experiment, while Section 3.2 presents the results.

3.1 Design of Experiment

We wanted to investigate our research question in a realistic setting, i.e., software professionals completing estimation tasks similar to those they normally complete. In order to isolate the factors we wished to study, we applied a 2x2 factorial design with random allocation of treatment. The two binary factors were related to the presence of irrelevant information and the degree of precision of the requirement specification.

Participants

The experiment was conducted at a conference for software developers (JavaZone 2005[2]). There were 76 software professionals participating in the experiment. On average, the participants each had 10 years of experience as software developer. The sample is self-selected in the sense that the participants were those who chose to attend a particular lecture on software cost estimation. This suggests that the participants might be more than averagely interested in estimation. This, in turn, may imply that any biases in the sample of participants are likely to be in the direction of better than average estimation expertise.

[2] www.javazone.no

The randomized allocation of treatment is likely to have eliminated systematic differences in personal characteristics within the sample.

Estimation task

Cooksey [23] cautions that experts are especially sensitive to the realism and familiarity in judgmental tasks. The estimation task was therefore based on an actual industrial task to ensure realism. To increase the likelihood of familiarity, we chose a small development task. This is based on the belief that software developers more frequently estimate smaller tasks, i.e. parts of a project, than the entire project, i.e., the likelihood of previous experience with similar estimation tasks increases with small tasks. In addition, the estimation of a small task would increase the similarity of time used for estimation in the experiment and in a real-world context. The participants estimated the effort required to write a simple program that retrieves a file from a remote server, validates the data and stores the data in an existing database. We believe that this is a rather general task that requires little specialized technology and domain knowledge.

We created four variants of the same requirement specification: i) a high-level requirement specification without irrelevant information being introduced, ii) a high-level requirement specification that included irrelevant information, iii) a detailed requirement specification without irrelevant information being introduced, and iv) a detailed requirement specification that included irrelevant information. The difference between the high-level and the detailed requirement specification was that the detailed requirement specification included explicit validation rules and a complete example of file format and file policy. The irrelevant information consisted of information about end

users' work processes, a description of the selection criteria that were used for selection of data providers, and information about systems that their implementation would not have to integrate with i.e., information that should not lead to more or less development effort. The requirement specifications are shown in Appendix A.

Two experienced developers validated the requirement specification. These two developers were asked to evaluate whether the four variants of the specification: 1) lacked any information normally found in this type of specification, 2) contained any errors, 3) described a realistic development task, and 4) was representative of tasks normally implemented by the conference attendees likely to participate in our experiment. In addition, they were requested to evaluate whether the irrelevant information that was introduced really was irrelevant for the purpose of estimating the software development effort.

Treatment

The participants were divided into four groups (groups A, B, C and D) with different variants of the requirement specification allocated to each group; see Table 1.

Table 1 Treatment in Experiment 1

Group	Detailed requirements	Irrelevant information
A	No	No
B	Yes	No
C	No	Yes
D	Yes	Yes

The specifications were handed out so that every fourth participant, by physical location, was allocated to the same group. The estimation task was included in a set of three other experiments, and a survey.

The participants took about 10 minutes to complete the effort estimation task, which is not unrealistic for this type of small estimation task. The participants' responses were collected immediately after the allocated time had expired.

Measurement

The participants were asked to estimate the most likely effort (in hours) they would need to implement the specified program. To measure the participants' confidence in their most likely estimate, we asked them to provide a minimum-maximum interval (in hours) that they were 90% certain would contain the actual effort. In the results, we present the relative width of the minimum-maximum interval as a measure of confidence. The relative width of the minimum-maximum intervals is calculated by the following formula:

RWidth = (Maximum value – Minimum value) / Estimate of most likely effort

The lower the RWidth, the higher the confidence in the accuracy of the estimate.

3.2 Results

An analysis of potential outliers revealed one obvious outlier that was removed from the data set. This participant submitted an effort estimate that was very much higher than that

of the other participants, i.e., he estimated the effort to be 1250 work-hours while the mean value of the remaining participants was 29.6 work-hours. We believe that this very high estimate indicates that the participant either did not take the task seriously, did not have the skill required to estimate meaningfully, or estimated something other than development and unit testing of the program specified in the requirement specification.

The results are displayed in Table 2 (estimates of most likely effort) and Table 3 (relative width of the minimum-maximum intervals).

Table 2 Estimates of Most Likely Effort (work-hours)

Group	N	Mean	Median	Min	Max	Stdv
A (high-level specification with no irrelevant info)	19	17.2	11.0	4	60	14.2
B (detailed specification with no irrelevant info)	18	22.2	17.5	5	70	17.5
C (high-level specification with irrelevant info)	20	32.8	30.0	8	80	20.1
D (detailed specification with irrelevant info)	18	46.7	24.5	4	250	65.5
A + B (no irrelevant information)	37	19.7	15.0	4	70	15.9
C + D (irrelevant information)	38	39.3	27.5	4	250	47.2

Table 3 Relative Width of Minimum-Maximum Interval (RWidth)

Group	N	Mean	Median	Min	Max	Stdv
A (high-level specification with no irrelevant info)	19	1.44	1.20	0.38	3.82	0.97
B (detailed specification with no irrelevant info)	18	1.27	0.83	0.22	6.00	1.32
C (high-level specification with irrelevant info)	20	0.86	0.78	0.33	1.50	0.40
D (detailed specification with irrelevant info)	18	0.99	0.85	0.47	2.00	0.44
A + B (no irrelevant information)	37	1.36	1.00	0.22	6.00	1.14
C + D (irrelevant information)	38	0.92	0.82	0.33	2.00	0.42

The results show that the participants that received requirement specifications with irrelevant information submitted, on average, higher effort estimates than the participants

that did not receive irrelevant information (mean of 19.7 vs. 39.3 work-hours). To analyze the effect of the independent variables "Irrelevant information introduced" (yes/no), "Detailed specification" (yes/no) and the interaction between the two on the dependent variable "Estimates" we fitted a General Linear Model (GLM). A log-transformation of the dependent variable was required to achieve a normal distribution of the residuals. The analysis shows that the impact of estimation-irrelevant information is highly significant ($p=0.01$), and that the interaction effect between the binary variables (related to presence of irrelevant information and/or level of specification) is not significant ($p=0.38$). The relative effect size of adding irrelevant information (based on Least Square Means estimates of irrelevant/relevant information) is +72%.

Surprisingly, the participants' confidence in the accuracy of their own estimates increased, i.e., the relative minimum-maximum interval width decreases, when irrelevant information was added! The mean relative width was 1.36 without irrelevant information, yet 0.92 when irrelevant information was included. The difference was statistically significant ($p=0.03$) applying log(relative width) to achieve a normal distribution of the residuals and GLM to compensate for any interaction effects between the independent variables related to irrelevant information and level of specification. This means that although the inclusion of irrelevant information affected the estimate, and consequently affected the level of realism negatively, confidence in the accuracy of the effort estimates actually increased.

4 Experiment 2

The second experiment was designed to: 1) test the robustness of the results in Experiment 1 on a different estimation task, 2) to further investigate the impact of irrelevant information, and 3) to investigate the effect of asking participants to explain the basis of their estimates, i.e., a weak variant of justification of effort estimates. As before, our focus is on the impact of irrelevant information. We will, therefore, only discuss the impact of justification in relation to possible interaction effect with level of irrelevant information. The design is discussed in Section 4.1. The results are presented in Section 4.2

4.1 Design of Experiment

The design of Experiment 2 was a 2x2 factorial design similar to the design of Experiment 1. The binary factors were presence/no presence of irrelevant information and justification/no justification of the estimates.

Participants

The experiment was conducted at an estimation seminar for professional software developers. We did not collect information about the participants' background in this experiment, but it is probable that the participants in Experiments 1 and 2 were similar with respect to experience and organizational role. This belief is based on the distribution of invited companies, and discussions with seminar attendees before and after the seminar.

Estimation Task

As in Experiment 1, we tried to create an estimation task that was small enough to be estimated realistically in a short experiment, representative for real-world estimation tasks, and based on assumptions about the use of technologies well known to the participants. The task was based on the estimation of a simple web application that registered seminar attendees in a database.

In this experiment, the treatments were as follows: requirements specification i) without irrelevant information and no request for justification of the estimate, ii) with irrelevant information and no justification, iii) without irrelevant information, but with justification, and, iv) with irrelevant information and justification. Justification of estimates was obtained by asking the participants to assess how relevant different parts of the requirement specification were for their estimates. The irrelevant information consisted of a description of a complex system that would, sometime in the future, replace the program they estimated. If this information had any relevance at all for the development effort, the impact should, we think, have been that information about future replacement would reduce the actual effort due to the lesser importance placed on long-term quality issues such as maintainability. The requirement specifications are shown in Appendix B. The different versions of the requirement specifications were, similarly to Experiment 1, validated with respect to quality, realism and whether the introduced additional information really was irrelevant for the purpose of effort estimation by software professionals.

Treatment

The participants were divided into four groups (group A, B, C and D) with different treatments allocated to each group, see Table 4.

Table 4 Treatment in Experiment 1

Group	Irrelevant information	Justification
A	No	No
B	Yes	No
C	No	Yes
D	Yes	Yes

As in Experiment 1, the participants were randomly allocated to treatment (by physical location in the seminar room), and the tasks were completed in a time frame and conditions similar to those in Experiment 1, i.e., the time spent on the estimate was about 10 minutes. The experimental task was included in a set of two experiments and one brief survey.

Measurement

As in Experiment 1, the participants were asked to estimate the most likely effort in work-hours and to provide a minimum-maximum interval (also in work-hours) that they were 90% certain would contain the actual effort.

4.2 Results

One participant in Group a (200 work-hours) and three participants in Group C (estimates of 150, 300 and 400 work-hours) were considered to be outliers and removed. These

participants were removed on the basis that the very high effort estimates made it likely that they either did not take the task seriously, did not have sufficient expertise, or misunderstood the task.

The results are displayed in Table 5 (estimates of most likely effort), and in Table 6 (relative width of the minimum-maximum intervals).

Table 5 Estimates of Most Likely Effort

Group	N	Mean	Median	Min	Max	Stdv
A (basic)	21	11.8	8	0.5	40	11.8
B (irrelevant information)	23	14.8	8	1.0	40	12.8
C (justification)	20	20.5	8	0.5	120	30.0
D (irrelevant information and justification)	24	22.5	11	3.0	100	24.4
A + C (no irrelevant information)	41	16.0	8	0.5	120	22.7
B + D (irrelevant information)	47	18.7	10	1.0	100	19.8

Table 6 Relative Width of Minimum-Maximum Interval (RWidth)

Group	N	Mean	Median	Min	Max	Stdv
A (basic)	21	1.22	1.00	0.40	3.80	0.73
B (irrelevant information)	23	1.21	1.20	0.38	2.10	0.55
C (justification)	20	1.18	1.00	0.25	2.67	0.68
D (irrelevant information and justification)	24	1.24	1.26	0.40	3.00	0.63
A + C (no irrelevant information)	41	1.20	1.00	0.25	3.80	0.70
B + D (irrelevant information)	47	1.22	1.20	0.38	3.00	0.58

The results show that (i) the participants who received requirement specifications with irrelevant information submitted higher effort estimates than those that did not receive irrelevant information (mean of 16.0 vs. 18.7 work-hours), and that (ii) the impact of irrelevant information seemed to be somehow moderated when the participants had to justify their estimates (mean of 11.8 vs. 14.8 work-hours when they did not have to

justify their estimate, mean of 20.5 vs. 22.5 when they did have to provide justification). Statistical analysis of the data, similar to that in Experiment 1, i.e., GLM analysis of log(ML estimates) and the binary variables “Irrelevant information” (yes/no) and “Justification” (yes/no), shows that the impact of estimation-irrelevant information is significant (p=0.08). The interaction effect (irrelevant information/justification) on the estimates is not significant (p=0.83). The relative effect size of adding irrelevant information (based on Least Square Means estimates of irrelevant/relevant information) is +52%. The results strengthen the results from Experiment 1 as they clearly point in the same direction.

The participants’ confidence in their own estimates, measured as mean relative width of minimum-maximum intervals, was not much affected by irrelevant information in this experiment with mean values of 1.22 (irrelevant information included) vs. 1.20 (without irrelevant information). Statistical analysis, by GLM analysis of log (relative width), of statistical significance of difference confirms this (p=0,74).

5 Discussion

In both experiments, the average estimate of most likely effort increased when estimation irrelevant information was included. The main reason for this effect of irrelevant information in our experiments may be the use of simple, unconscious estimation processes, so-called judgmental heuristics by the participants. Such heuristics are frequently used to solve complex problems, where the human mind is not capable of implementing the “normatively correct” processes [24].

To keep the estimation processes simple, the participants in our study may have based their effort estimates on easily available variables with no causal relationship to effort, on the assumption that these variables usually *correlate* well with amount of effort. The length of the text, the number of systems mentioned, or simply the assumption that everything in the requirement specification is relevant [9] may be examples of irrelevant information used as indicator of effort. Judgmental heuristics, such as those used in software effort estimation, are often unconscious processes (often they have originally been analytic and evolved into tacit processes as they have been used repeatedly with success [24]). The unconscious use of variables means that the effort estimates may have been affected by information elements that the estimators, when asked about it, would admit are irrelevant in the current estimation situation.

Two elements of such, more or less, unconscious heuristics are "estimation-by-analogy" and "first impression":

- *Estimation-by-analogy*: Estimation by analogy is quite common in software effort estimation [25]. Estimation by analogy is based, to some extent unconsciously, on retrieving one or more tasks from memory (the analogies) that resemble the task that is going to be estimated, and then creating the estimate based on properties and actual effort of the retrieved tasks. In the experiments, the selection of analogies might have been based on surface cues (e.g., the number of systems mentioned, the technical platform information) or in-depth cues (e.g., the steps involved in solving the task). Irrelevant information might have surface similarity to previous tasks that differ in the underlying structure. This means that the irrelevant information might have led to misleading, or at least other, analogies compared to the situation without irrelevant information.

- “First impression”: People can be strongly affected by their first impression when making predictions and other decisions under uncertainty [26], e.g. studies have found that court decisions are affected severely by the jury's first impression. The estimation process might be based on an early, unconscious, categorization (“first impression”) of the estimation task into a predefined category. When the estimate is created, it is strongly influenced by the initially chosen category, e.g., that the task looks like a “medium-large task”. In both experiments, the irrelevant information was placed early in the requirement specification, and might therefore have caused an incorrect ”first impression” that was difficult to change with more information, e.g., by reading further in the requirement specification, as confirming evidence has a stronger effect than evidence that does not fit with the initially chosen task category, i.e., the effect of “theory-loaded observations” [27, 28].

More studies on the effect of the amount, the type, the extremity, the framing and the placement of irrelevant information are needed to better understand when and how it affects effort estimates. Until we know how to neutralize the effect of irrelevant information, we believe the best strategy is to try to avoid it altogether, particularly in the early stages of the estimation process.

Some might find this advice counterintuitive, because average effort estimates increased in both experiments when irrelevant information was included, and it is well known that software cost estimates are usually too optimistic. However, other irrelevant information can cause effort estimates to decrease. Unless the estimation process is based

on information that is relevant for the actual use of effort, systematic improvement of judgment-based effort estimation may be very difficult.

6 Summary and Recommendations

It may be the exception, rather than the rule, that all information in a requirement specification is relevant for estimation of software development effort. Does this not matter or is it essential to avoid irrelevant information in requirement specifications? We designed two experiments to answer the research question of whether effort estimates were affected by the presence of irrelevant information.

The two experiments (with 76 and 92 participants) answered this research question with the observation that estimation-irrelevant information in requirement specifications *strongly* affected the software effort estimates. The average effort estimates increased significantly in both experiments when estimation-irrelevant information was included. In addition, the results of Experiment 1 suggest that the estimators may also become more confident in the accuracy of their own estimates when they are exposed to irrelevant information.

The magnitude of the effect differed and we currently have a quite incomplete understanding of how, when and how much different irrelevant information affects cost estimation. Consequently, further research is needed.

Until we have a better understanding of the impact of irrelevant information on expert judgment-based effort estimates, we believe it to be essential that irrelevant information is *removed* from requirement specifications before presented to the

estimators [29]. If this is impossible, it may be a good idea to highlight and present early the most relevant information to avoid incorrect first impressions [30]. The removal of irrelevant information is important even when using formal estimation models, i.e., formal estimation models are typically based on expert judgment-based input. This input may also be affected by irrelevant information.

Acknowledgment

Thanks to Dr. Erik Arisholm at Simula Research Laboratory for criticism and suggestions that improved the study design and the statistical analysis. Also, thanks to the Norwegian Research Council's SPIKE project for funding.

References

1. Moløkken, K. and M. Jørgensen. *A review of software surveys on software effort estimation*. in *International Symposium on Empirical Software Engineering*. 2003. Rome, Italy: Simula Res. Lab. Lysaker Norway.
2. Jørgensen, M., *A review of studies on expert estimation of software development effort.* Journal of Systems and Software, 2004. **70**(1-2): p. 37-60.
3. Armstrong, J.S., ed. *Principles of forecasting: A handbook for researchers and practitioners*. International Series in Operations Research & Management Science. 2001, Kluwer Academic Publishers: Boston. XII, 849 s. : ill.
4. Jørgensen, M. *The "Magic Step" of Judgment-Based Software Effort Estimation*. in *International Conference on Cognitive Economics*. 2005. Sofia, Bulgaria: NBU Press.
5. Boehm, B., et al., *Software cost estimation with Cocomo II*. 2000, New Jersey: Prentice-Hall.
6. Albrecht, A.J. and J.E. Gaffney Jr, *Software function, source lines of code, and development effort prediction: A software science validation.* IEEE Transactions on Software Engineering, 1983. **9**(6): p. 639-648.
7. Whitecotton, S.M., D.E. Sanders, and K.B. Norris, *Improving predictive accuracy with a combination of human intuition and mechanical decision aids.* Organizational Behaviour and Human Decision Processes, 1998. **76**(3): p. 325-348.
8. Lim, J.S. and M. O'Connor, *Judgmental forecasting with time series and causal information.* International Journal of Forecasting, 1996. **12**(1): p. 139-153.

9. Tversky, A. and D. Kahneman, *Judgment under uncertainty: Heuristics and biases.* Science, 1974. **185**: p. 1124-1131.
10. Hristova, P., G. Petkov, and B. Kokinov. *The Influence of Irrelevant Information on Price Judgment*. in *Advances in Cognitive Economics*. 2005. Sofia, Bulgaria: NBU Press.
11. Gaeth, G.J. and J. Shanteau, *Reducing the Influence of Irrelevant Information on Experienced Decision Makers,* in *Judgment and Decision Making: An interdisciplinary reader*, T. Connoly, H. Arkes, and K.R. Hammond, Editors. 2003, Cambridge University Press: Cambridge. p. 305-321.
12. Jørgensen, M. and M. Shepperd, *A systematic Review of Software Development Cost Estimation Studies.* Submitted to IEEE Transactions on Software Engineering, 2005.
13. Jørgensen, M. and D.I.K. Sjøberg, *Impact of effort estimates on software project work.* Information and Software Technology, 2001. **43**(15): p. 939-948.
14. Jørgensen, M. and D.I.K. Sjøberg, *The impact of customer expectation on software development effort estimates.* International Journal of Project Management, 2004. **22**: p. 317-325.
15. Aranda, J. and S. Easterbrook. *Anchoring and Adjustment in Software Estimation*. in *European software engineering conference*. 2005. Lisbon, Portugal: ACM Press.
16. Epley, N., *A Tale of Tuned Decks? Anchoring as Accessibility and Anchoring as Adjustment,* in *Blackwell Handbook of Judgment and Decision Making*, D.J. Koehler and N. Harvey, Editors. 2004, Blackwell Publishing Ltd. p. 240-257.
17. Zukier, H., *The dilution effect: The role of correlation and the dispersion of predictor variables in the use of diagnostic information.* Journal of Personality and Social Psychology, 1982. **43**: p. 1164-1174.
18. Stewart, T.R., *Improving Reliability of Judgmental Forecasts*, in *Principles of Forecasting*, J.S. Armstrong, Editor. 2001, Kluwer Academic Publishers: Boston. p. 81-106.
19. Oskamp, S., *Overconfidence in case-study judgments.* Journal of Consulting Psychology, 1965. **29**(3): p. 261 - 265.
20. Ettenson, R., J. Shanteau, and J. Krogstad, *Expert judgment: Is more information better.* Psychological Reports, 1987. **60**(1): p. 227-238.
21. Marzocchi, G., et al., *The disturbing effect of irrelevant information on arithmetic problem solving in inattentive children.* Developmental Neuropsychology, 2002. **21**(1): p. 73-92.
22. Jasper, J. and S. Christman, *A neuropsychological dimension for anchoring effects.* Journal of Behavioral Decision Making, 2005. **18**(5): p. 343.
23. Cooksey, R.W., *Judgment Analysis: Theory, Methods and Applications*. 1996, London: Academic Press, Inc.
24. Gigerenzer, G., *Fast and Frugal Heuristics: The tools of Bounded Rationality,* in *Blackwell Handbook of Judgment and Decision Making*, D.J. Koehler and N. Harvey, Editors. 2004, Blackwell Publishing: Oxford, UK. p. 63-88.
25. Heemstra, F.J., *Software cost estimation.* Information and Software Technology, 1992. **34**(10): p. 627-639.

26. Keren, G. and K.H. Teigen, *Yet Another Look at the Heuristics and Biases Approach*, in *Blackwell Handbook of Judgment and Decision Making*, D.J. Koehler and N. Harvey, Editors. 2004, Blackwell Publishing: Oxford, UK. p. 89-109.
27. Brehmer, B., *In one word: Not from experience.* Acta Psychologica, 1980. **45**: p. 223-241.
28. Einhorn, H.J. and R.M. Hogarth, *Confidence in judgment: Persistence of the illusion of validity.* Psychological Review, 1978. **85**(5): p. 395-416.
29. Williams, C., *The effect of an irrelevant information dimension on "same-different" judgments of multi-dimensional stimuli.* Quarterly Journal of Experimental Psychology, 1974. **26**: p. 26-31.
30. Conway, J.M., R.A. Jako, and D.F. Goodman, *A meta-analysis of interrater and internal consistency reliability of selection interviews.* Journal of Applied Psychology, 1995. **80**: p. 565-579.

Appendix A: The Requirement Specifications of Experiment 1

Group A: High-level requirement specification without introduced irrelevant information

The following scenario is constructed, but effort has been made to ensure realism:

A publishing company, Company X, is going to start printing telephone catalogues. They have agreed to buy subscriber information from a telecom company. The subscriber information is available in a file that can be retrieved by ftp transfer. In order to produce the catalogues, the subscriber information must be imported to a database that is used by Company X's publishing system.

Therefore, Company X needs a small program that automatically retrieves the file, containing records of names and telephone numbers, from an external server, performs simple data validation and stores the records in a local database (accessible through a jdbc interface). There are no performance requirements (the file is relatively small) and no authentication, except standard logon, is required.

The program should be written in plain Java, and will be run manually from command line by a Company X operator. Any errors should be logged to a log-file on the operator's

pc, and then handled manually by the operator. The program will be run only once, so no check for duplicate records are needed.

Assume that you are a developer in Company X's IT department and that you are familiar with the relevant technologies. You are free to use development tools of choice.

A) Estimate the most-likely effort (i.e. do not include any risk buffer) it would require for you to implement and unit test the program?

Most-likely Estimate _____ (hours)

B) Indicate an interval (by specifying max effort and min effort) that you are 90% confident that the actual effort of implementing and unit testing the program will fall within?

Min _____ (hours) Max _____ (hours)

Group B: Detailed requirement specification without introduced irrelevant information

The following scenario is constructed, but effort has been made to ensure realism:

A publishing company, Company X, is going to start printing telephone catalogues. They have agreed to buy subscriber information from a telecom company. The subscriber information is available in a file that can be retrieved by ftp transfer. In order to produce the catalogues, the subscriber information must be imported to a database that is used by Company X's publishing system.

Therefore, Company X needs a small program that automatically retrieves the file, containing records of names and telephone numbers, from an external server, performs simple data validation and stores the records in a local database (accessible through a jdbc interface). There are no performance requirements (the file is relatively small) and no authentication, except standard logon, is required.

The program should be written in plain Java, and will be run manually from command line by a Company X operator. Any errors should be logged to a log-file on the operator's pc, and then handled manually by the operator. The program will be run only once, so no check for duplicate records are needed.

There will be only one file in the source FTP folder at any time, and the file name does not change. Each record in the file will always have the following structure:

PersonalId; Name; Address; Country; Mobile phone number; Private phone number;

Example:

27027342911; Ola Normann; Ullevållsæteren 0999 Oslo; Norge; 99990000; 22112211;

04128042844; Kari Nordmann; Frognersæteren 9999 Oslo; Norge; 90000000; 55554444;

The data transformation and parsing is straight forward due to the simple and consistent format of the file and the correspondingly simple database.

The complete set of validation rules are:

- mobile phone number must be 8 digits long and the first digit must be 4 or 9
- phone number must be 8 digits long and the first digit must be different from 4 and 9
- month of birth (third and fourth digits in PersonalId) must be between 01 and 12
- day of birth (two first digits in PersonalId) must be between 01 and 31
- name and address fields must be non-empty

Assume that you are a developer in Company X's IT department and that you are familiar with the relevant technologies. You are free to use development tools of choice.

A) Estimate the most-likely effort (i.e. do not include any risk buffer) it would require for you to implement and unit test the program?

Most-likely Estimate _____ (hours)

B) Indicate an interval (by specifying max effort and min effort) that you are 90% confident that the actual effort of implementing and unit testing the program will fall within?

Min _____ (hours) Max _____ (hours)

Group C: High-level requirement specification including irrelevant information

The following scenario is constructed, but effort has been made to ensure realism:

A publishing company, Company X, is going to start printing telephone catalogues. They have agreed to buy subscriber information from a telecom company. The subscriber information is available in a file that can be retrieved by ftp transfer. In order to produce

the catalogues, the subscriber information must be imported to a database that is used by Company X's publishing system.

The publishing system used to produce the catalogues is used to produce all publications made by CompanyX. It is primarily used by graphical designers, content editors and production managers. The users access it through a web interface. The system is made, and maintained, by an external vendor, and has been used by Company X for several years. In addition to the publishing system, all users groups use their specific portfolio of domain specific tools (such as Photoshop for the graphical designers).

Several other telecom operators offer access to similar subscriber information, but CompanyX has chosen to buy subscriber information from this telecom operator because they offered the best price. The file with the subscriber information is generated by the telecom operator's datawarehouse system which runs on top of an Oracle database. The file is stored locally, and then moved to the transfer area which is located on a Unix server that is opened for external FTP access. The url to the ftp server is: 89.93.163.115. The username is "companyX" and the password is "topsecret".

Therefore, Company X needs a small program that automatically retrieves the file, containing records of names and telephone numbers, from an external server, performs simple data validation and stores the records in a local database (accessible through a jdbc interface). There are no performance requirements (the file is relatively small) and no authentication, except standard logon, is required.

The program should be written in plain Java, and will be run manually from command line by a Company X operator. Any errors should be logged to a log-file on the operator's pc, and then handled manually by the operator. The program will be run only once, so no check for duplicate records are needed.

Assume that you are a developer in Company X's IT department and that you are familiar with the relevant technologies. You are free to use development tools of choice.

A) Estimate the most-likely effort (i.e. do not include any risk buffer) it would require for you to implement and unit test the program?

Most-likely Estimate _____ (hours)

B) Indicate an interval (by specifying max effort and min effort) that you are 90% confident that the actual effort of implementing and unit testing the program will fall within?

Min _____ (hours) Max _____ (hours)

Group D: Detailed requirement specification including irrelevant information

The following scenario is constructed, but effort has been made to ensure realism:

A publishing company, Company X, is going to start printing telephone catalogues. They have agreed to buy subscriber information from a telecom company. The subscriber information is available in a file that can be retrieved by ftp transfer. In order to produce the catalogues, the subscriber information must be imported to a database that is used by Company X's publishing system.

The publishing system used to produce the catalogues is used to produce all publications made by CompanyX. It is primarily used by graphical designers, content editors and production managers. The users access it through a web interface. The system is made, and maintained, by an external vendor, and has been used by Company X for several years. In addition to the publishing system, all users groups use their specific portfolio of domain specific tools (such as Photoshop for the graphical designers).

Several other telecom operators offer access to similar subscriber information, but CompanyX has chosen to buy subscriber information from this telecom operator because they offered the best price. The file with the subscriber information is generated by the telecom operator's datawarehouse system which runs on top of an Oracle database. The file is stored locally, and then moved to the transfer area which is located on a Unix

server that is opened for external FTP access. The url to the ftp server is 81.93.163.115. The username is "companyX" and the password is "topsecret"

Therefore, Company X needs a small program that automatically retrieves the file, containing records of names and telephone numbers, from an external server, performs simple data validation and stores the records in a local database (accessible through a jdbc interface). There are no performance requirements (the file is relatively small) and no authentication, except standard logon, is required.

The program should be written in plain Java, and will be run manually from command line by a Company X operator. Any errors should be logged to a log-file on the operator's pc, and then handled manually by the operator. The program will be run only once, so no check for duplicate records are needed.

There will be only one file in the source FTP folder at any time, and the file name does not change. Each record in the file will always have the following structure:

PersonalId; Name; Address; Country; Mobile phone number; Private phone number;

Example:
27027342936; Ola Normann; Ullevållsæteren 0999 Oslo; Norge; 99990000; 22112211;
04128042844; Kari Nordmann; Frognersæteren 9999 Oslo; Norge; 90000000; 55554444;

The data transformation and parsing is straight forward due to the simple and consistent format of the file and the correspondingly simple database.

The complete set of validation rules are:

- mobile phone number must be 8 digits long and the first digit must be 4 or 9
- phone number must be 8 digits long and the first digit must be different from 4 and 9
- month of birth (third and fourth digits in PersonalId) must be between 01 and 12
- day of birth (two first digits in Personalid) must be between 01 and 31
- name and address fields must be non-empty

Assume that you are a developer in Company X's IT department and that you are familiar with the relevant technologies. You are free to use development tools of choice.

A) Estimate the most-likely effort (i.e. do not include any risk buffer) it would require for you to implement and unit test the program?

Most-likely Estimate _____ (hours)

B) Indicate an interval (by specifying max effort and min effort) that you are 90% confident that the actual effort of implementing and unit testing the program will fall within?

Min _____ (hours) Max _____ (hours)

Appendix B: The Requirement Specifications of Experiment 2

Group A: Requirement specification without irrelevant information and no request for justification of the estimate

In this exercise, you will be asked to estimate the effort required to implement a solution.

Simula organizes several seminars each year. At present, participants register for the seminars by sending an email to a given contact person.

Simula wants you to develop a simple web system (one web page) for registration of participants. The system will handle registration of all Simula's seminars. The participants will register on the web by submitting their email address and a registration code that uniquely identifies the seminar (this code is sent to them by email). The only functionality in the system is to store the email address and the registration code in a database. All queries will be done manually (in sql), and there is no need for any validation of submitted data. There are no security requirements.

The system will run on a webserver that has Tomcat, Java and MySql installed and running. Assume that you are familiar with the relevant technologies. You are free to use development tools of choice.

A) Estimate the most-likely effort (i.e. do not include any risk buffer) it would require for you to implement and unit test the program?

Most-likely Estimate _____ (hours)

B) Indicate an interval (by specifying max effort and min effort) that you are 90% confident that the actual effort of implementing and unit testing the program will fall within?

Min _____ (hours) Max _____ (hours)

Group B: Requirement specification with irrelevant information and no justification of the estimate

In this exercise, you will be asked to estimate the effort required to implement a solution.

Simula organizes several seminars each year. At present, participants register for the seminars by sending an email to a given contact person.

Simula has ordered a web-based registration system. This system will let participants choose seminar from a list, and then register name, company and contact information. An email that confirms the registration will be sent to the participants. The system will keep

track of the number of participants for each seminar and automatically close registration when the seminar is fully booked. There will also be a management module where employees at Simula can log in and query and manipulate conferences. This system will run on a Weblogic server application server, use a Sybase database and be fully integrated with Simula's other web pages. However, due to economical priorities, this system will not be developed before the end of 2006. Simula therefore needs an intermediate system for registration. This system will be thrown away when the new system is ready.

Simula wants you to develop a simple web system (one web page) for registration of participants. The system will handle registration of all Simula's seminars. The participants will register on the web by submitting their email address and a registration code that uniquely identifies the seminar (this code is sent to them by email). The only functionality in the system is to store the email address and the registration code in a database. All queries will be done manually (in sql), and there is no need for any validation of submitted data. There are no security requirements.

The system will run on a webserver that has Tomcat, Java and MySql installed and running. Assume that you are familiar with the relevant technologies. You are free to use development tools of choice.

A) Estimate the most-likely effort (i.e. do not include any risk buffer) it would require for you to implement and unit test the program?

Most-likely Estimate _____ (hours)

B) Indicate an interval (by specifying max effort and min effort) that you are 90% confident that the actual effort of implementing and unit testing the program will fall within?

Min _____ (hours) Max _____ (hours)

Group C: Requirement specification without irrelevant information, but with justification of the estimate

In this exercise, you will be asked to estimate the effort required to implement a solution, and also to point out the information in the specification that influenced your estimate.

Simula organizes several seminars each year. At present, participants register for the seminars by sending an email to a given contact person.

Simula wants you to develop a simple web system (one web page) for registration of participants. The system will handle registration of all Simula's seminars. The participants will register on the web by submitting their email address and a registration code that uniquely identifies the seminar (this code is sent to them by email). The only functionality in the system is to store the email address and the registration code in a database. All queries will be done manually (in sql), and there is no need for any validation of submitted data. There are no security requirements.

The system will run on a webserver that has Tomcat, Java and MySql installed and running. Assume that you are familiar with the relevant technologies. You are free to use development tools of choice.

The table below lists some of the information elements in the requirement specification. Please indicate how relevant each information element is for your estimate:

	Not relevant	Slightly relevant	Some relevance	Relevant	Very relevant	Critical
Functional requirements (e.g., register data entered in a web page in database)						
Technical requirements (e.g., Java, Tomcat, MySql)						
Context information (e.g. manual queries in sql, invitation letter to participants)						
Development tools (e.g., that they are free of choice)						
Impression of the customer (Simula)						
Expected incompleteness of requirements (requirement changes)						
Other factors (please list below)						

Other factors: _____

A) Estimate the most-likely effort (i.e. do not include any risk buffer) it would require for you to implement and unit test the program?

Most-likely Estimate _____ (hours)

B) Indicate an interval (by specifying max effort and min effort) that you are 90% confident that the actual effort of implementing and unit testing the program will fall within?

Min _____ (hours) Max _____ (hours)

Group D: Requirement specification with irrelevant information and justification of the estimate

In this exercise, you will be asked to estimate the effort required to implement a solution, and also to point out the information in the specification that influenced your estimate.

Simula organizes several seminars each year. At present, participants register for the seminars by sending an email to a given contact person.

Simula has ordered a web-based registration system. This system will let participants choose seminar from a list, and then register name, company and contact information. An email that confirms the registration will be sent to the participants. The system will keep track of the number of participants for each seminar and automatically close registration when the seminar is fully booked. There will also be a management module where employees at Simula can log in and query and manipulate conferences. This system will run on a Weblogic server application server, use a Sybase database and be fully integrated with Simula's other web pages. However, due to economical priorities, this system will not be developed before the end of 2006. Simula therefore needs an intermediate system for registration. This system will be thrown away when the new system is ready.

Simula wants you to develop a simple web system (one web page) for registration of participants. The system will handle registration of all Simula's seminars. The

participants will register on the web by submitting their email address and a registration code that uniquely identifies the seminar (this code is sent to them by email). The only functionality in the system is to store the email address and the registration code in a database. All queries will be done manually (in sql), and there is no need for any validation of submitted data. There are no security requirements.

The system will run on a webserver that has Tomcat, Java and MySql installed and running. Assume that you are familiar with the relevant technologies. You are free to use development tools of choice.

The table below lists some of the information elements in the requirement specification. Please indicate how relevant each information element is for your estimate:

	Not relevant	Slightly relevant	Some relevance	Relevant	Very relevant	Critical
Functional requirements (e.g., register data entered in a web page in database)						
Technical requirements (e.g., Java, Tomcat, MySql)						
Information about the system that will replace the intermediate system						
Context information (e.g. manual queries in sql, invitation letter to participants)						
Development tools (e.g., that they are free of choice)						
Impression of the customer (Simula)						
Expected incompleteness of requirements (requirement changes)						
Other factors (please list below)						

Other factors: _____

A) Estimate the most-likely effort (i.e. do not include any risk buffer) it would require for you to implement and unit test the program?

Most-likely Estimate _____ (hours)

B) Indicate an interval (by specifying max effort and min effort) that you are 90% confident that the actual effort of implementing and unit testing the program will fall within?

Min _____ (hours) Max _____ (hours)