Evidence-Based Guidelines for Assessment of Software Development Cost Uncertainty

Magne Jørgensen

Abstract—Several studies suggest that uncertainty assessments of software development costs are strongly biased toward overconfidence, i.e., that software cost estimates typically are believed to be more accurate than they really are. This overconfidence may lead to poor project planning. As a means of improving cost uncertainty assessments, we provide evidence-based guidelines for how to assess software development cost uncertainty, based on results from relevant empirical studies. The general guidelines provided are: 1) Do not rely solely on unaided, intuition-based uncertainty assessment processes, 2) do not replace expert judgment with formal uncertainty assessment models, 3) apply structured and explicit judgment-based processes, 4) apply strategies based on an outside view of the project, 5) combine uncertainty assessments from different sources through group work, not through mechanical combination, 6) use motivational mechanisms with care and only if greater effort is likely to lead to improved assessments, and 7) frame the assessment problem to fit the structure of the relevant uncertainty information and the assessment process. These guidelines are preliminary and should be updated in response to new evidence.

Index Terms—Cost estimation, management, software psychology, uncertainty of software development cost.

1 Introduction

"IT [the 1976 Olympics in Montreal] can no more lose money than a man can have a baby."—Montreal Mayor Jean Drapeau. The Olympics lost more than one billion dollars [1].

Accurate assessment of the uncertainty of software development cost estimates[1] is important 1) when deciding whether or not to embark upon a project, 2) to support the bidding process, and 3) to support decisions about how large the project's contingency budget should be [2]. To illustrate the use of uncertainty assessments, consider a project that the project leader believes has a median cost (an estimate) of about \$100,000, i.e., he believes that there is a 50 percent probability of spending \$100,000 or less. He wants to be reasonably sure that the actual cost does not exceed the budgeted cost. That being so, he needs to decide how large a contingency buffer he should add to the estimated cost to have the desired confidence in that the actual cost does not overrun the project's budget. As a prerequisite for making such a decision, he must assess the probabilities of different levels of cost usage. If, for example, he assesses the probabilities of exceeding different levels of cost to be as described in Fig. 1, then, to be about 70 percent sure of not exceeding the budget, the project should have a contingency buffer of \$25,000, i.e., \$125,000-\$100,000.

Clearly, the use of cost uncertainty assessments that reflect the underlying uncertainty will improve the budgeting and planning process. Unfortunately, as documented in this paper, it is easy to be overconfident about the accuracy of cost estimates, e.g., to base the contingency buffer on a much too optimistic cost uncertainty distribution. For this reason, we need to improve the way in which we conduct software cost estimation uncertainty assessments. Notice that this paper is *not* about the estimation of most likely cost or about the reasons for estimation error. The paper focuses on how to improve the assessment of the cost uncertainty of software projects.

The paper is organized as follows: Section 2 introduces cost uncertainty terminology, categories, and evaluation measures useful for the understanding of the other parts of the paper. Section 3 describes how we identified the studies used as input to the guidelines and discussions. Section 4 reviews state-of-practice of software cost uncertainty assessments. This section concludes that unaided human judgment-based uncertainty assessment currently is the dominating type of strategy and that this type of strategy easily leads to overconfidence. Section 5 discusses extensions to, and replacements of, unaided human judgment-based assessment strategies. Section 6 suggests guidelines for uncertainty assessments based on the discussions in Section 4 and 5. Section 7 concludes the paper and points out important topics for future research.

2 Uncertainty Terminology, Categories, and Measures

2.1 Uncertainty and Probability

In this paper, we define uncertainty in terms of probability, i.e., the degree of uncertainty of an event is described by the probability that the event will happen. For example, we may describe the cost uncertainty of a software project by its 90 percent confidence effort prediction interval [\$20,000; \$40,000]. This prediction interval means that it believed that

1. We use the term "cost estimate" to denote both cost and effort estimate in this paper.

• *The author is with Simula Research Laboratory, PO Box 134, 1325 Lysaker, Norway. E-mail: magne.jorgensen@simula.no.*

Manuscript received 6 Sept. 2004; revised 27 July 2005; accepted 1 Aug. 2005; published online 1 Dec. 2005.
Recommended for acceptance by A. Finkelstein.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number TSE-0185-0904.

0098-5589/05/\$20.00 © 2005 IEEE Published by the IEEE Computer Society

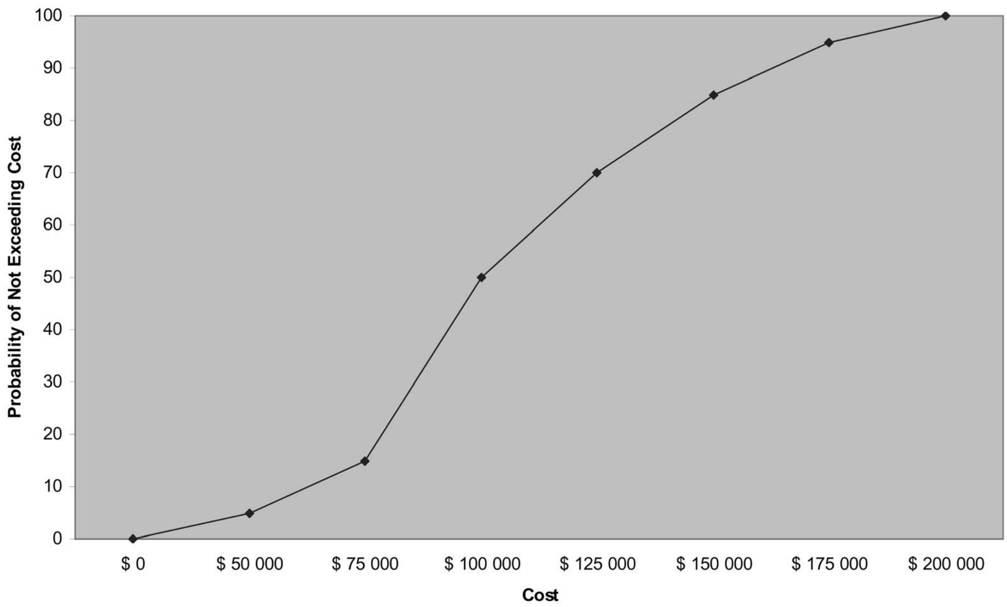

Fig. 1. Cost uncertainty distribution.

it is a 90 percent probability that the actual cost is higher than or equal to $20,000 and lower than or equal to $40,000.

When assessing cost uncertainty the *objective*, i.e., the actual, probabilities are typically not known. The probability-based uncertainty assessments are therefore assessments based on *subjective* probabilities, e.g., what the subjects *believe* are the probabilities of including the actual costs in cost intervals. In principle, we will never know the objective probability of including the actual cost in a single cost prediction interval. This paper and, as far as we know, most other papers on judgment-based uncertainty introduce *frequency* of a class of similar events as a substitute for the *average* objective probability of the individual events in that class. For example, assume that a project manager provides minimum-maximum cost intervals of 100 software projects. The observed frequency of including the actual cost in the minimum-maximum intervals is 60 percent, which is interpreted as the average objective probability of including the actual cost in the cost minimum-maximum intervals. Now, if the subjective probabilities, i.e., the uncertainty assessments, of the project manager were much higher, e.g., that the project manager believed that he on average had a 90 percent probability of including the actual costs, we say that the project manager is overconfident in his/her cost estimation accuracy assessment.

Both subjective and objective probabilities may be described as based on two different sources: 1) the *inherent* uncertainty regarding cost usage in a software project and 2) the uncertainty caused by *lack of knowledge* about the project. We do not separate between these two sources in most of the discussions in this paper, but it may nevertheless be important to understand that an uncertainty assessment is a result of uncertainty from both sources.

2.2 Categories of Uncertainty

The suggested categories of uncertainty are based on our own observations of how software projects treat uncertainty and inspired by the uncertainty categories presented in [3], [4]. The categories are: *normal variance*, *known risks*, *unforeseen events*, and *flexibility of outcome and process*. In comparison with previous frameworks, our contribution is the introduction of the uncertainty category *flexibility of outcome and process*. We believe it is useful to be aware of the different types of uncertainties to understand the results of the reported studies and the motivation for the guidelines. Our interpretation of the uncertainty categories is as follows:

- *Normal variance*. Uncertainty of known activities resulting from what is considered to be "normal variation" in a project's software development performance. Normal variation is typically a result of many small uncertainties.
- *Known risks*. Uncertainty that results from the potential occurrence of foreseen events (positive and negative) that are analyzed at the time of estimation. These events (if occurring) impact on the project's performance significantly, i.e., the impact is considered to be outside the "normal variance."
- *Unforeseen events*. Uncertainty that results from the occurrence of events not included in the normal variation type of events and not known at time of estimation, i.e., "unexpected events."
- *Flexibility of outcome and process*. Uncertainty reduction that results from flexibility in what the customer perceives as acceptable outcome and process. For example, a project may be able to compensate for the occurrence of unforeseen events, without adding a contingency buffer, by a simplification of functionality or documentation. This flexibility in process and product, included in many software projects, leads to a greater reduction in uncertainty than a situation in which there is only one possible outcome and only one possible process leading to the project's completion [5]. The impact of this category is similar

to the impact on project management of the "option of corrective actions" discussed in [6].

2.3 Inside versus Outside Views of Uncertainty

Kahneman and Lovallo [7] separate human judgment processes into two categories: "inside view"-based and "outside view"-based. We will apply these concepts later on in this paper. Applied in the context of cost estimation uncertainty assessment the categories can be described as follows:

- *Inside View*. An inside view-based uncertainty assessment process is based on a *decomposition* of the total cost estimation uncertainty into individual cost estimation uncertainties related to, among other things, specific activities and risks. An inside view-based process therefore requires a thorough understanding of the inside of the project and an ability to combine individual cost uncertainties. An inside view is similar to a "bottom-up" estimation analysis approach.
- *Outside View*. The uncertainty of the software cost estimate may be determined by comparing outside properties of the current development project with previously completed projects, i.e., a process similar to estimation by analogy. The underlying assumption is that projects that have similar characteristics will, on average, behave similarly regarding uncertainty. An outside view is similar to a "top-down" estimation analysis approach.

2.4 Evaluating Uncertainty Assessments

Evaluating cost uncertainty assessments is more difficult than evaluating cost estimation accuracy. Whereas the accuracy of individual cost estimates can be assessed by comparing them to actual efforts, individual cost uncertainty assessments have no obvious corresponding actual values. In the long run, however, a K percent confidence level should correspond to a proportion of correct assessments ("Hit rate") similar to K percent. For example, given a number of cost intervals with 90 percent confidence, we should expect that about 90 percent of these included the actual effort. A mismatch between the confidence level and the "Hit rate" implies that the assessments are inaccurate. If the "Hit rate" is systematically lower than the confidence level, we observe a *bias* toward overconfidence and, if it is higher, we observe a bias toward underconfidence.

The following definition of "Hit rate" is based on uncertainty assessments on the cost prediction interval format, e.g., that it is believed to be "90 percent probable that the actual cost is in the interval [\$80,000; \$125,000]." The "Hit rate," however, may easily be adapted to include one-sided assessments, e.g., that it is believed to be "95 percent probable that the actual cost is less than \$125,000." We measure the hit rate as:

$$HitRate = \frac{1}{n}\sum_i h_i,\ h_i = \begin{cases} 1, & \min_i \le Act_i \le \max_i \\ 0, & Act_i > \max_i \vee Act_i < \min_i, \end{cases}$$

where $\min_i$ and max_i are, respectively, the minimum and maximum values of the prediction interval for the cost estimate of task i, Act_i is the actual cost of task i, and n is the number of estimated tasks.

We provide a more comprehensive discussion of different uncertainty assessment evaluation measures in [8].

3 Search Process

The studies identified were based on the following three distinct search processes (Search completed April 2004):

Search 1 is a comprehensive, systematic, issue-by-issue, manual search for studies on assessment of software cost uncertainty in about 100 journals. The journals were selected on the basis that they included at least one paper on software cost estimation. We have included the list of journals in the Appendix to support other researchers' future research on software cost estimation, i.e., we believe that an exhaustive search for related cost estimation research should include a search in all these journals. We identified 25 journal papers that contained empirical results relevant for the assessment of software cost uncertainty.

Search 2 is a search for guideline relevant papers from other domains based on a manual examination of titles and abstracts of papers in the following psychology and forecasting journals: *Organizational Behavior and Human Decision Processes*, *Journal of Behavioral Decision Making*, *Journal of Personality and Social Psychology*, *Memory and Cognition*, and *Journal of Forecasting*. More than 200 papers were identified and examined. A selection of studies relevant to the main topics of our review was included, based on our informal judgment of relevance and representativeness. For example, we identified more than 20 studies relevant for the effect on motivation on uncertainty assessments, but included only eight studies in the discussion. Professor in psychology K.H. Teigen at the University of Oslo reviewed the included studies and found that they were representative and relevant for the discussion of this paper.

Search 3 is a search for relevant conference papers and uncertainty assessment books in library databases, e.g., searches in Inspec and PsycINFO applying the terms "uncertainty assessment" and "motivation." More than 50 relevant conference papers and books were identified.

The estimation paper library www.simula.no/BESTweb includes most papers identified through *Search 1*, *2*, and *3*. A brief summary of the three search processes is included in Table 1.

There is no explicit quality assessment of the studies included in our review. The implicit quality assessment is that we have emphasized journal papers in our search and that results supported by more than one study are considered better supported by evidence. For obvious reasons, the results from the studies on software cost uncertainty assessments are most emphasized when assessing the strength of evidence of the guidelines.

4 State of Practice

The main results of this section are briefly described in Table 2.

TABLE 1
Search Strategies and Goals

Search Process	Type of Search	Goal
Search 1	Exhaustive, manual search for empirical studies on assessment of software development cost estimation uncertainty published in journals.	Identify *all* journal papers on assessment of software development cost estimation uncertainty.
Search 2	Manual search of selected psychology and forecasting journals for papers on assessment of uncertainty.	Identify a *representative* set of relevant studies from *other domains* than software development.
Search 3	Digital library-based search for relevant studies presented at software engineering conferences, published in books, or in journals not examined in *Search 1* or *2*.	Complement *Search 1* and *2*.

TABLE 2
Main Results

Section	Research Question	Main Result
4.1	Which cost uncertainty assessments strategies are applied by the software industry?	The software industry seems to base most cost uncertainty assessments on unaided, intuition-based expert judgment.
4.2	a) How realistic are expert judgment-based cost uncertainty assessments? b) What are the reasons for unrealistic uncertainty assessments?	a) Software professionals have a strong tendency to be over-confident, i.e., to systematically underestimate the uncertainty of the cost estimates. b) Potential reasons for over-confidence include: Interpretation difficulties of uncertainty concepts, lack of feedback and evaluation, and, hidden agendas.

4.1 Uncertainty Assessment Strategies

There have been a few attempts to build formal software cost uncertainty model, e.g., [9], [10], [11], [12], and several general statistical uncertainty models for estimation and planning purposes, e.g., [13], [14]. In addition, there are many frameworks and tools supporting a structured elicitation and combination of project uncertainties, e.g., [4], [15], [16].

The study presented in [8] suggests that there is little, if any, use of such formal cost uncertainty assessment models among software professionals. Instead, uncertainty assessments seem to be based on unaided, intuition-based human judgment ("expert judgment"), i.e., processes based on the nonexplicit and nonrecoverable mental processes of software professionals. Similar observations are reported in a project survey of Hihn and Habib-Agahi [17]. They found, among other things, that the studied organization had no explicit process for incorporating risk and uncertainty assessments in the cost estimation work. We have found no studies suggesting that formal uncertainty assessment models, frameworks, or tools are in common use by the software industry.

According to [8], software projects typically describe cost uncertainty by applying one of the following means: 1) cost prediction intervals, e.g., that it is "almost certain" or that there is "90 percent probably" that the actual cost will be between \$50,000 and \$175,000, 2) categories of cost uncertainty, e.g., that an estimate is a $+/-$ 10 percent (low uncertainty) or $+/-$ 50 percent (high uncertainty) cost estimate,[2] or 3) Informal uncertainty terminology, e.g., that a cost estimate is a "ballpark" figure or "very rough" estimate.

2. The exact semantic of these categories may be unclear, e.g., how likely should it be that the actual cost is inside $+/-$ 10 percent to be assessed as a $+/-$ 10 percent estimate.

4.2 Overconfidence

The use of unaided expert judgment-based uncertainty assessment strategies would not be problematic if the judgments were accurate. Unfortunately, several studies report systematic underestimation of cost uncertainty, i.e., systematic overconfidence, when based on unaided human judgment:

- Connolly and Dean [18] report that the actual effort used by student programmers to solve programming tasks fell inside their 98 percent confidence effort prediction (minimum-maximum) intervals in only about 60 percent (hit rate) of the cases, i.e., the effort prediction intervals were much too narrow to reflect 98 percent confidence. Explicit attention to, and training in, establishing good minimum and maximum effort values did increase the hit rate to about 70 percent, which was still far from the required 98 percent.
- Jørgensen et al. [8] studied the software development activity estimates of 195 student projects activities and 49 industry project tasks. The effort prediction intervals of the activities of the student projects were based on a 90 percent confidence level, but had a hit rate of only 62 percent. The effort prediction intervals of the tasks of the industrial projects were based on the confidence level "almost certain" and had a hit rate of only 35 percent, i.e., a strong underestimation of uncertainty.
- Jørgensen and Teigen [19] conducted an experiment in which 12 software professional were asked to provide 90 percent confidence effort prediction intervals on 30 previously completed, similar maintenance tasks. In total, 360 effort prediction intervals were provided. The software professionals had

access to a small experience database of similar projects and were informed about the actual effort of a task after each uncertainty assessment. Although "90 percent confident," the professionals' hit rates were, on average, only 64 percent on the first 10 tasks (Tasks 1-10), 70 percent on the next 10 tasks (Tasks 11-20), and 81 percent on the last 10 tasks (Tasks 21-30). In other words, even after 20 tasks with feedback after each task, there was a systematic bias toward overconfidence. In real-world software development situations, we cannot expect so many learning-friendly environments as in this experiment, i.e., situations with many similar uncertainty assessments tasks completed in a short period of time with immediate feedback. The low learning in this learning-friendly environment suggests a much poorer learning, maybe no learning at all, in most real-world environments.

- Jørgensen [20] studied the effort minimum-maximum intervals provided by seven realistically composed estimation teams. The teams assessed the uncertainty of the effort estimate of the same two projects, i.e., in total, 14 project effort prediction intervals. The projects had been completed in the same organization as that to which the software professionals who participated in the study belonged. The average confidence level of the estimation teams was 86 percent, but only 43 percent of the teams' effort prediction intervals included the actual effort.
- Studies of uncertainty estimates based on unaided human judgment have been conducted in other domains for many years. Most of those studies report levels of overconfidence similar to that in the software domain. See, for example, the studies described in [21], [22], [23], [24]. Lichtenstein and Fischhoff [25] report that the level of overconfidence seems to be unaffected by differences in intelligence and expertise, so we should *not* expect the level of overconfidence to be reduced with greater experience. Arkes [26] provides an overview of studies from different domains on overconfidence, strongly supporting the overconfidence claim.

There is, therefore, strong evidence of a systematic bias in human judgment toward underestimation of the uncertainty of software projects. Potential reasons for this overconfidence are, according to [8], as follows:

- *Interpretation difficulties*. Software professionals have understandable problems interpreting the concepts "90 percent confident," "90 percent probable," or "almost certain" when historical data are sparse and statistical skills are poor. In [8], the authors report results from an experiment where the estimators did not, on average, provide different effort minimum-maximum intervals when 70 percent, 90 percent, and 99 percent confident.
- *Lack of feedback and evaluation*. There was no evaluation of the performance of the prediction interval and no analysis that enabled learning from experience in the software companies analyzed in [8], i.e., the software companies were, as far as can be gleaned from our observations, not even aware of the degree of overconfidence of their uncertainty assessments.
- *Hidden agendas*. Software professionals may have goals, e.g., personal goals, other than accurate cost uncertainty assessments. In particular, the desire to be evaluated as a skilled software developer may contribute to overly narrow cost prediction intervals. In [8], the authors report from an experiment where software project managers perceived the developer providing the narrowest effort prediction intervals to be the most skilled software developer. This perception was present even in situations where the managers knew that the effort prediction intervals of those developers were the most overconfident. When cost uncertainty assessments are not evaluated with respect to accuracy, it may be, from an individual's point of view, rational to emphasize alternative goals, such as appearing skilled by presenting overly narrow prediction intervals.

Overconfident uncertainty assessments may lead to poor project plans and, consequently, poor project performance. State-of-practice evidence therefore suggests that there is a need to improve the ways in which we conduct cost uncertainty assessments in software projects.

5 Improvement of Uncertainty Assessments

This section reviews the following replacements of, and extensions to, unaided, judgment-based uncertainty assessments: application of formal uncertainty models (Section 5.1), formalization of judgment-based processes (Section 5.2), mechanical combinations of uncertainty assessments (Section 5.3), group work-based uncertainty assessments (Section 5.4), improving motivation for accuracy (Section 5.5), and improving framing of the uncertainty assessment problem (Section 5.6).

The selection of these topics is based on what typically is studied in uncertainty assessment papers, e.g., we only review potential improvements where we could find attempts of empirical evaluation. This means, for example, that the "rules of thumb" suggested by NASA [27] are not analyzed.

The main results of this section are briefly described in Table 3.

5.1 Formal Uncertainty Models

Formal models are not subject to the same overconfidence biases as software professionals, e.g., they may not be as vulnerable to biases resulting from a desire to appear skilled or to please the customers. There are two main categories of formal uncertainty assessment model that so far have been empirically investigated in software contexts:

- *Indirect models*. Indirect models derive uncertainty assessments as "by-products" of models of most likely effort or cost. Those models include only variables that are relevant for estimation of the most likely cost, i.e., if a variable is relevant for the

TABLE 3
Main Results

Section	Research Question	Main Result
5.1	Is the use of formal uncertainty models likely to improve the software cost uncertainty assessments?	The choice between unaided expert judgment and formal uncertainty models is frequently a choice between over-confidence and inefficient use of relevant uncertainty information. The inefficient use of uncertainty information by the formal uncertainty models implies that they are currently not very useful.
5.2	Is a formalization of the uncertainty assessment process likely to improve the software cost uncertainty assessments?	Yes. Several studies suggest that proper formalizations, e.g., more structured judgment-processes, leads to more realistic cost uncertainty assessments.
5.3	Is a mechanical combination of uncertainty assessments likely to improve the software cost uncertainty assessments?	Maybe. There are few cost uncertainty assessment studies of this topic. The available studies suggest that proper mechanical combination strategies may be hard to derive and that the improvement is less than with group work-based combinations (see Section 5.4).
5.4	Is a group work-based combination of uncertainty assessments likely to improve the software cost uncertainty assessments?	Group work has the potential of improving the uncertainty assessments. There are, however, many pitfalls of group work, e.g., the "polarization effect", that should be addressed to achieve the improvement.
5.5	Is the use of motivational mechanisms, e.g., "monetary incentives for accuracy", likely to improve the software cost uncertainty assessments?	Not necessarily. A prerequisite for improvement from increased motivation may be that the strategy applied is explicit and improves with more effort. Unaided, intuition-based uncertainty assessment strategies are not of that type.
5.6	Is a change of the framing of the uncertainty assessment likely to improve the software cost uncertainty assessments?	There seems to be an improvement in realism resulting from a change of the traditional uncertainty assessment framing. For example, improvement seems to result from the replacement of: "Provide a minimum-maximum effort interval that includes the actual effort with a probability of 90%", with: "Assess the probability that the actual effort is inside the interval [50% of most likely effort; 200% of most likely effort].

uncertainty but not for the estimation of most likely cost, it is not included in the model. The empirically evaluated indirect models are 1) use of cost prediction intervals of regression models of most likely cost [9], [10] and 2) prediction intervals based on bootstrapping of analogy-based cost estimation models [9].

- *Direct models*. Direct models derive uncertainty assessments from models of estimation uncertainty, i.e., models that only include variables that are important for the uncertainty of the estimation of most likely cost. The evaluated direct models are: use of empirical and parametric distribution of previous estimation accuracy [10] and regression models of the estimation error [11].

The results reported in the available empirical studies suggest that formal models are, as expected, able to remove the bias toward overconfidence, e.g., their use yields a much better correspondence between hit rate and confidence level than do assessment by software professionals. However, there seems to be important limitations related to the "efficiency" of formal uncertainty models.

Angelis and Stamelos [9] evaluated prediction intervals based on regression models and bootstrapping models. Both models were developed for the prediction of most likely effort, i.e., they are *indirect models*. They report that both types of formal models were able to provide unbiased prediction intervals, i.e., about 95 percent of the actual effort values were included in the 95 percent confidence prediction intervals. Unfortunately, for our purposes, the authors did not compare the performance of formal models to that of software professionals. However, the very wide effort prediction intervals provided by the formal models suggest an inefficient use of uncertainty information. For example, the parametric bootstrap model-based provided 95 percent effort prediction intervals with maximum effort typically 10 times the minimum effort. In light of the variation of estimation error of actual projects reported in other studies, e.g., [11], [28], and the experience-based minimum-maximum cost interval span for software projects described in [27], the model-based effort prediction intervals in [9] seem to be unrealistically wide. Much of the interval width may be a result of inaccurate models of most likely effort and lack of integration of important uncertainty information, i.e., most of the uncertainty is "model uncertainty" (poor integration of knowledge) and not "project uncertainty" (inherent uncertainty).

Additional findings supporting the view of inefficient formal uncertainty models are provided in [10]. In that study, the authors compared human judgment-based effort prediction intervals with prediction intervals from regression models of most likely effort (*indirect model*) and empirical distribution of estimation error (*direct model*). The authors concluded that the choice between human judgment and formal models is frequently a choice between the avoidance of prediction intervals that are so wide as to be meaningless and the avoidance of systematic bias toward intervals that are too narrow.

Similar results were found when evaluating regression models of estimation error (*direct model*) [11]. In that paper, the authors explain the poor efficiency as follows: "*An analysis of the model residuals and the estimators' own descriptions of reasons for low/high estimation accuracy suggest that we cannot expect formal models to explain most of the* *estimation accuracy and bias variation, unless the amount of observations and variables is unrealistically high. For example, many important reasons for low estimation accuracy are connected to seldom-occurring events and cost management issues*." That problem of integrating some types of information in formal estimation models is not confined to software studies. Whitecotton et al. [29], for example, found in a study of accounting students that: "*Human intuition was useful for incorporating relevant information outside the scope of the model*."

A potential use of formal uncertainty models not evaluated in this paper is sensitivity analysis, i.e., to use the model as tools to better understand how different project properties and events are interconnected. Then, an inside view-based uncertainty model may be useful, as applied in the system dynamic models described in [30], [31], [32].

The size and the complexity of the software project may have an impact on the usefulness of formal uncertainty models, e.g., formal models may be more useful for uncertainty assessments of large, complex projects. Unfortunately, we have not been able to find empirical studies on this issue.

5.2 Formalization of the Uncertainty Assessment Processes

In some professional domains, e.g., the management of nuclear reactors, there is, according to [33], a trend toward more formalization of human judgment-based uncertainty assessment processes. An important reason for that trend may be that formalization of the process enables review of the process by others. It is frequently not satisfactory to base important management or investment decision on one group's or one individual's "gut feelings" about the uncertainty, i.e., on uncertainty assessments that are impossible to review.

One example of formalization of the judgment process is the process described in [34]. This process is similar to the process of the formal model described in [10]. The main differences to the formal uncertainty model are that that the selection of similar projects is based on human judgment, not on a formal algorithm, e.g., a clustering algorithm, and that there is an element of adjustment of the final values. The benefit of the model is based on, among other things, the finding that people often seem to be better assessors when asking "how frequently X happens" instead of "how probable is X" [35]. While providing frequencies imposes an outside view, examining historical data and providing subjective probabilities frequently implies an inside view, where the individual uncertainties are analyzed and combined. As reported in [36], the inside view easily leads to overconfidence. The above process was evaluated in [34] and [37]. These studies suggest that the formalization led to improvements, although not for all estimation teams. An important result from the studies is the warning against unlimited judgment-based adjustments of the outcome of formalized uncertainty assessment processes. Such post-adjustments may easily reintroduce the strong bias toward overconfidence.

There have been several attempts to formalize the cost uncertainty process based on an inside view of project cost uncertainties, e.g., the frameworks described in [4], [15]. Typically, these frameworks apply simulation techniques, e.g., the Monte Carlo simulation, to implement the complex adding of interconnected uncertainty distributions. The frameworks seem to provide limited support on how to provide individual cost uncertainty distributions and the relationships between the uncertainty distributions. To the best of our knowledge, formalizations based on the inside view have not been properly evaluated with respect to accuracy and practicability, e.g., it is not clear whether software professionals in general are able to use these frameworks or not.

5.3 Mechanical Combinations of Uncertainty Assessments

The benefits of combining predictions from different sources are well documented. For example, based on 30 empirical studies, Armstrong [38] reports that predictions based on the mean value of individual predictions were, on average, 12.5 percent more accurate than the individual predictions themselves. Similarly, empirical studies report promising results from combining software estimates of the most likely cost from different sources, e.g., [39], [40], [41]. This paper focuses on the combination of uncertainty assessments, not on combination of estimates of most likely outcome. It is therefore not obvious that we can transfer these positive combination results to our software cost uncertainty assessment context.

Strategies for, and benefits of, the mechanical combination of software development cost uncertainty assessments have, as far as we know, only been studied in [42]. According to [43], there are not many studies at all, i.e., regardless of domain, on the topic of combining uncertainty assessments. The study described in [42] evaluated three combination strategies: 1) Average of the individual minimum and maximum values, 2) Maximum and minimum of the individual maximum and minimum values, and 3) Group process (discussion)-based prediction intervals. Strategies 1 and 2 are examples of the mechanical combination of uncertainty assessments. Strategy 3 is based on group work (and will be discussed in greater detail in Section 5.4). The empirical study reported in that paper, with software professionals, suggested that Strategy 1 led to little improvement in correspondence compared with the individual cost prediction intervals, mainly because of a strong individual bias toward too narrow prediction intervals that could not be removed by averaging the values. Strategies 2 and 3 both improved the correspondence. However, Strategy 3 used the uncertainty information more efficiently, in that it yielded narrower prediction intervals for the same degree of correspondence between hit rate and confidence.

We have not found any study on the benefits of combining human judgment and model-based software development cost uncertainty assessments. A lack of research on this topic is unfortunate, since combinations of model and expert judgment have been shown to frequently outperform both models alone and experts alone in other domains; see, for example, [44]. Models and experts have complementary strengths and a combination of the strengths of each approach may lead to significant benefits.

5.4 Group Work-Based Combinations of Uncertainty Assessments

Estimates and uncertainty assessments may be derived from group discussions. The structure imposed on the group work may vary a lot, from formal Delphi-based processes [45] to more unstructured processes [46].

There has been some scepticism regarding the use of groups to assess risk or uncertainty. Many of them are based on the awareness of the "group-think"-effect [47], i.e., that group members feel a pressure to have the same opinions as, and think similarly to, the other members of the group. The social pressure from groups may even operate at an unconscious level, according to [48]. Studies show that there may be a "risky shift" in groups, i.e., that the group as a whole is much more willing to make risky decisions than each individual member [49]. More recent studies suggest that the more general effect of group-work is the "polarization effect" [50], i.e., groups with a majority of members who are prone to making risky judgments become more prone to making such judgments, while groups with a majority of members who are averse to making risky judgments become more risk averse. The effects of group-work may be difficult to predict. For example, the study reported in [51] found that groups' estimates became more conservative (risk averse) because the groups' members believed that the other groups' members' estimates were too optimistic.

Not all studies report unwanted effects from group work. There are, for example, several studies that report good results from the use of groups to estimate and plan projects, e.g., [52], [53]. It is therefore difficult to provide a general conclusion on the effect of group work with respect to the accuracy of uncertainty assessment, based on a general review of previous related studies. The effect obviously depends on the composition of the group, the group-work processes, and the assessment context. The following empirical study-based relationships should therefore be interpreted carefully:

- Group-work may typically lead to the identification of more activities of software projects [46] and, as a consequence, to more realistic estimates of most likely cost than individual estimates. Greater realism in estimates of most likely cost typically contributes to greater realism in cost uncertainty assessments.
- Discussion between people with different types of work may lead to the identification of project work in the interface between these types of work [46], i.e., group-work may lead to a consideration of more information relevant to uncertainty. This does not necessarily have much impact on the realism of cost uncertainty assessment if an inside view-based strategy is applied because of the combinatory complexity. More information has been found to increase the overconfidence when the additional information is irrelevant or only slightly relevant [29], [54]. Group work may also lead to the identification of a higher number of previously completed similar projects, contributing to a larger database of project analogies relevant to uncertainty assessments.
- Group work may lead to a higher degree of overconfidence if the group members assess the cost uncertainty of their own development work [55]. Then, the desire of appearing skilled by exhibiting high confidence may hinder accuracy in cost uncertainty assessments [8].
- Group work may lead to a higher degree of evaluation. This can lead to more, no change in, or less overconfidence in uncertainty assessments, dependent on the strategy applied. One example of increased evaluation is the use of a "devil's advocate," i.e., a person allocated to the role of arguing for alternative views [56]. The use of a "devil's advocate" may force the group to defend its position and consider arguments that do not support the current uncertainty assessment, e.g., the group may have to face questions like, "Most other similar project have had large unexpected problems. Is it likely that our project will be different?"

5.5 Improved Motivation

There are a variety of motivation-based uncertainty assessment improvement strategies, e.g., "identification of individual performance," "evaluation and feedback," "provision of arguments for the uncertainty assessment calculations," and "monetary incentives for accuracy." All of them are based on the belief that the use of motivation-based strategies lead to greater concern about performance and, hence, better performance. The effect of motivational mechanisms is, however, complex. For example, several studies suggest that higher motivation may result in a *fall* in performance on difficult tasks, e.g., [57], [58]. The common explanation for decreased performance is that higher motivation may lead to greater use of dominant responses, i.e., less reflection and more "instinct" [59]. This means that a possible effect of increased motivation is even more overconfident software cost uncertainty assessments, e.g., that the urge to provide narrow prediction intervals so as to be evaluated as skilled increases with increased accountability. However, other studies, e.g., [60], show no effect or a positive effect from increased motivation on performance. Lerner and Tetlock [61] summarize the findings in a review of accountability-studies: "*Two decades of research now reveal that (a) only highly specialized subtypes of accountability lead to increased cognitive effort; (b) more cognitive effort is not inherently beneficial; it sometimes makes matters even worse;...*" The pessimistic view regarding motivational mechanisms is not undisputed. A comprehensive review on financial incentives [62], suggest that incentives, in general, have positive effects.

We were unable to find any published software study on the effect of motivational strategies on accuracy of software cost uncertainty assessment, and only one study [63] on the effect of higher motivation on cost estimation accuracy. That study, [63], found significant, positive correlation between increased accountability through performance evaluation and improved estimation accuracy. In fact, they state that performance evaluation was the *only* indicator of improved estimation accuracy: "*Only one managerial practice,* *the use of the estimate in performance evaluations of software managers and professionals, presages greater accuracy. By implication, the research suggests somewhat ironically that the most effective approach to improve estimating accuracy may be to make estimators, developers, and managers more accountable for the estimate even though it may be impossible to direct them explicitly on how to produce a more accurate one*." There are important differences between the estimation of most likely cost and the assessment of the uncertainty of a cost estimate. One issue is, however, similar, the "self-fulfilling prophecy" effect. That effect suggests that an initially overconfident cost estimate or uncertainty assessment may actually become realistic if the project members perceive it as a goal. For example, a high motivation for not exceeding the estimated maximum cost may imply that the project simplifies the functionality of the software and work smarter to avoid a very large cost overrun. Case studies and experiments illustrating this "self-fulfilling prophecy" are described in [5]. A possible explanation for the benefits of higher accountability in [63] is the "self-fulfilling prophecy" effect.

Our preliminary summary of the motivation-effect studies is that there seem, in the main, to be two conditions that enable benefit to be derived from improved motivation toward uncertainty assessment: 1) There must be an explicit uncertainty assessment process where the accuracy improves with more effort or 2) There must be a substantial flexibility in the software development process or product, to enable the effect of the "self-fulfilling prophecy." How we design motivational mechanisms is obviously important. Below, we present empirically validated findings that may be useful for the design and tailoring of motivational mechanisms:

- The motivational mechanisms should be directed toward the process more than the outcome [64]. There may be many reasons for a poor uncertainty assessment and not all of them can be attributed to poor assessment work. Consequently, rewarding the outcome may easily lead to the reward of poor and punishment of good uncertainty assessment work.
- The viewpoints of the audience, e.g., the software managers or customers, should *not* be known at the time of assessment [61]. Otherwise, the assessor may easily be even more biased to confirm with the audience with increased motivation. If the audience's viewpoint is not known, increased motivation seems to increase the assessors' preemptive self-criticism, which typically lead to better performance.
- There should be no unfortunate mixture of motivational mechanisms leading to conflicting goals, see the discussion in [65]. Optimally, the only goal of the uncertainty assessment should be realism.
- There should be no use of external incentives, e.g., financial rewards, if people have a strong *intrinsic* motivation, i.e., they perform activities for their own sake. In this case, the use of external incentives may destroy the intrinsic motivation and lead to poorer performance [66].
- It may be beneficial to instruct software professionals to explain and defend their cost uncertainty assessments. This motivational mechanism may be particularly useful if no (or strongly delayed) feedback related to outcome will be provided [67], as the case is in many software project uncertainty assessments. For example, if there are no organizational mechanisms for providing feedback on the minimum-maximum interval provided by a software developer, it is even more important to require explicit and valid argumentation for the uncertainty assessment.

5.6 Improved Framing of the Uncertainty Assessment Problem

As far as we have observed, the majority of the studies on human judgment, e.g., the study by Hora et al. [68], report that overconfidence is robust to differences in framing. There are, however, studies that report that the framing can be essential in some contexts, e.g., regarding stochastic problems [69]. Hence, there is considerable variation in the results of studies on human judgment.

The only framing study in the context of software development is, as far as we know, [34]. In that study, the authors showed that the framing did, indeed, have an important impact. The field study involved 70 software projects in two different software development organizations. Two different uncertainty assessment framings were compared: 1) The traditional "Provide a minimum-maximum effort interval that includes the actual effort with a probability of 90 percent" and 2) The alternative "Assess the probability that the actual effort is inside the interval [50 percent of most likely effort; 200 percent of most likely effort]." The choice of framing had a surprisingly large effect. Those who received the traditional minimum-maximum effort interval framing showed the usual pattern of overconfidence. Those who received the alternative framing, however, achieved a very close correspondence between confidence level and "hit rate." The effect of this shift in framing has been confirmed in other domains [70].

6 Preliminary Guidelines

This section condenses what we believe are major uncertainty assessment results of Sections 4 and 5 into seven guidelines. Although the guidelines reflect the results of the studies discussed in this paper, it is difficult to avoid subjectivity in their selection and formulation. In addition, it is highly likely that, in the near future, uncertainty assessment results will be published that should prompt changes in the guidelines. It is therefore important to emphasize that the guidelines should be considered *preliminary* and need to be revised regularly in the light of new evidence. The preliminary guidelines are, to a large extent, research hypotheses subject to further studies and possible falsification.

The validity scope of the guidelines is not well described. In general, this is not well described in the underlying studies and we need better studies to be able to be more precise about the scope validity, e.g., to which degree a guideline mainly is valid for large, complex software projects and not so much for small, simple software projects.

We have included a classification of the strength of the evidence of a guideline, i.e., "strong," "medium," and "weak." This classification is informal and based on our subjective assessment. "Strong" evidence indicates that we have found many relevant studies and that the great majority of evidence is in favor of the guideline, "medium" strong evidence means either that we found few relevant supporting studies or that we identified many relevant studies with most evidence in favor of and some against the guideline, and "weak" evidence means that the number of studies found was very low or that the results of the studies found were only weakly in favor of the guideline.

Guideline 1. Do not Rely Solely on Unaided, Intuition-Based Processes.

Explanation. Unaided, intuition-based software cost estimation uncertainty assessments are on average systematically biased toward overconfidence.

Evidence [8], [18], [19], [20], [21], [23], [24], [25], [26]. See Section 4 for discussion of the evidence.

Strength of evidence. Strong.

Guideline 2. Do not Replace Expert Judgment with Formal Models.

Explanation. Judgments made as a result of using formal models may have a better correspondence between confidence level and accuracy of uncertainty assessments than the unaided judgments of software professionals. On the other hand, formal models seem to apply uncertainty assessments less efficiently compared with software professionals. Current formal cost prediction interval models yield intervals that are so wide as to be meaningless, to compensate for lack of uncertainty information specific for a single project.

Evidence [9], [10], [11], [29]. See Section 5.1 for discussion of the evidence.

Strength of evidence. Medium.

Guideline 3. Apply Structured and Explicit Judgment-Based Processes.

Explanation. As much as possible of the uncertainty assessment process should be structured and explicit (formalized) to enable review of the quality of the assessment process, learning and evaluation of reasons for overconfidence.

Evidence [20], [22], [29], [35], [36], [37], [54], [71], [72]. See Section 4 and Section 5.2 for discussion of evidence.

Strength of evidence. Strong.

Guideline 4. Apply Strategies Based on an Outside View of the Project.

Explanation. Apply uncertainty assessment strategies based on an *outside* view of the project, e.g., strategies that compare uncertainty properties of the current project with the estimation accuracy of previously completed projects. Inside view-based uncertainty assessment strategies seem to require formalizations of uncertainty relationships that are too complex to be useful in most software projects and should only be used if there are no relevant historical data. Notice that this does *not* mean that it is unimportant to assess the inside uncertainties, e.g., the project risks and the minimum-maximum effort intervals of individual activities. That type of information about uncertainty may be very important for project planning and management. What we suggest is that assessments of the *total* cost estimation uncertainty are based on an outside view of the project.

Evidence. The evidence is connected to the evidence for Guideline 3, i.e., strategies applying an outside view seem to have several of the properties recommended in Guideline 3. For additional evidence, see [7], [37] and the discussion in Section 5.2.

Strength of evidence. Medium.

Guideline 5. Combine Uncertainty Assessments From Different Sources Through Group Work, Not Through Mechanical Combination.

Explanation. Group work where the participants have different types of background seems to be a useful combination strategy for cost uncertainty assessment. Be aware of "group-think" in coherent groups where goals other than accuracy become important. Mechanical combination of uncertainty assessment, i.e., not through group work, may be more problematic. For example, while the strategy "take the average of individual estimates" is an obvious, and well-documented, strategy for combining most likely estimates, there may not be any obvious strategy when combining uncertainty assessments. Mechanical combinations may nevertheless, given a proper combination algorithm, be better than single source uncertainty assessments.

Evidence [8], [29], [42], [46], [54], [55], [56]. See Sections 5.3 and 5.4 for a discussion of the evidence.

Strength of evidence. Weak. There is much evidence both in favor of and against the guideline. We have included the guideline mainly because the available software cost uncertainty assessment results are in favor of it.

Guideline 6. Use Motivational Mechanisms With Care and Only If It Is Likely That More Effort Leads to Improved Assessments.

Explanation. Studies suggest that increased motivation may have a negative impact, or no impact at all, on accuracy. For example, higher motivation may have a negative effect if the increased motivation leads to use of "more instinct and less reflection" and no effect if the underlying assessment processes is unconscious. Positive effects from motivational mechanisms are more likely if the uncertainty assessment process improves with greater effort, the mechanisms are mainly directed toward the process not the outcome, the evaluators' viewpoints are not known, there are no conflicting evaluation goals, and the intrinsic motivation for accuracy is low.

Evidence [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67]. See Section 5.5 for a discussion of the evidence.

Strength of evidence. Medium.

Guideline 7. Frame the Assessment Problem to Fit the Structure of the Uncertainty Relevant Information and the Assessment Process.

Explanation. Information may be of little use in cases where the information structure does not fit the assessment problem framing and process. For example, a traditional method of assessing uncertainty is to ask a software developer to provide a 90 percent confidence effort prediction interval, i.e., a minimum-maximum interval where the estimator believes that there is a 90 percent probability of including the actual effort of development tasks. Then, it can be demonstrated that the information about the previous estimation error of similar projects is difficult to apply. If, on the other hand, the software developer is asked to assess the probability of not exceeding the budget with more than X percent, he or she may investigate the previous projects and find that, for example, 20 percent of the projects exceeded the budget by more than X percent, i.e., the historical information together with the assessment process fit the framing of the assessment problem. Two subguidelines belonging to this guideline are: 1) Do not ask for high confidence (90 percent, or worse, 98 percent) effort prediction intervals. Lower confidence intervals are much more likely to be realistic. 2) Suggest (or ask for) a prediction interval before you ask for confidence estimates (alternative framing). People are much better assessing their confidence corresponding to a given minimum-maximum interval than constructing an interval corresponding to a given level of confidence.

Evidence [34], [69], [73], [74], [75]. See Section 5.6 for a discussion of the evidence.

Strength of evidence. Medium strong for software cost uncertainty assessments.

7 Conclusions and Future Work

Software project cost estimation uncertainty assessments are frequently based on expert judgment, i.e., unaided, intuition-based processes. Such uncertainty assessments have been shown to be systematically overconfident and may lead to poor project management. There have been many studies on how to improve uncertainty assessments, both in the software development domain and other domains. The review presented in this paper synthesizes findings of empirical uncertainty assessment studies into seven practical, evidence-based guidelines. The guidelines suggest, among other things, that the most promising strategies are not based on formal models, but on supporting the expert processes, and that there are several important prerequisites for deriving benefits from motivational mechanisms. The guidelines are preliminary and there is a strong need to evaluate them in different software development contexts.

In spite of the preliminary state of the guidelines, it is our belief that the current version constitutes a practical and useful, although incomplete, guide for software organizations when designing their own cost uncertainty assessment process.

As for future work, we believe that, in particular, the following three uncertainty assessment related topics in particular deserve more research focus:

- Development and evaluation of processes that systematically train software professionals in uncertainty assessments. We did not find any software related study on this topic.
- Development and evaluation of different types of formalizations of judgment-based uncertainty assessment processes. This focus is, we believe, more promising than a total replacement of expert judgment with formal uncertainty assessment models.
- Better understanding of the mental processes underlying uncertainty assessment, e.g., what type of experience and processes are involved when a software professional claims to be 90 percent sure that the actual effort will not exceed 1,500 work-hours?

Appendix

Journals Searched for Studies on Software Cost Uncertainty Assessment

ACM Transactions on Computer Personnel, *Ada User Journal*, *Advances in Computers*, *Advances in Information Systems*, *American Programmer/Cutter IT Journal*, *Annals of Software Engineering*, *Applied Computing Review*, *Australian Journal of Information Systems*, *Automated Software Engineering*, *Communications of the ACM*, *Computers & Operations Research*, *Computing and Control Engineering Journal*, *Concurrent Engineering: Research and Applications*, *Datamation*, *Embedded Systems Programming*, *Empirical Software Engineering*, *Engineering Economist*, *Engineering Intelligent Systems for Electrical Engineering and Communications*, *European Journal of Information Systems*, *Expert Systems*, *Expert Systems with Applications*, *GEC Journal of Research*, *Human Factors*, *IBM Systems Journal*, *ICL Technical Journal*, *IEE Proceedings Software*, *IEE Proceedings Software Engineering*, *IEEE Aerospace and Electronic Systems Magazine*, *IEEE Computer*, *IEEE Multimedia*, *IEEE Software*, *IEEE Transactions on Computers*, *IEEE Transactions on Software Engineering*, *IEEE Transactions on Systems, Man and Cybernetics*, *IIE Transactions*, *Industrial Management & Data Systems*, *Information and Management*, *Information and Software Technology*, *Information Resources Management Journal*, *Information Strategy: The Executive's Journal*, *Information Systems Journal*, *Information Systems Management*, *Information Systems Research*, *Information Technology & Management*, *International Journal of Project Management*, *International Journal of Software Engineering and Knowledge Engineering*, *International Journal of Systems Science*, *Journal of Computer and Software Engineering*, *Journal of Computer Information Systems*, *Journal of Defense Software Engineering*, *Journal of End User Computing*, *Journal of Experimental and Theoretical Artificial Intelligence*, *Journal of Information Technology*, *Journal of Management Information Systems*, *Journal of Parametrics*, *Journal of Software Maintenance and Evolution: Research and Practice*, *Journal of Systems and Software*, *Journal of Systems Management*, *Management Science*, *MIS Quarterly*, *New Review of Hypermedia and Multemedia*, *Pakistan Journal of Information and Technology*, *Programming and Computer Software*, *R.F.-Design*, *Scandinavian Journal of Information Systems*, *SIGPLAN Notices*, *Software—Practice and Experience*, *Software Engineering Journal*, *Software Engineering Notes*, *Software Quality Journal*, *Software World*, *Technometrics*, *Texas Instru-*

ments Technical Journal, *The Australian Computer Journal*, *Transactions of the Information Processing Society of Japan*, *Vitro Technical Journal*.

ACKNOWLEDGMENTS

The author would like to thank Dr. Kjetil Moløkken at Simula Research Laboratory, the Professor in Psychology at the University of Oslo, Karl Halvor Teigen, and Dr. Chris Wright at Cambridge Language Consultants for their useful suggestions and interesting discussions.

REFERENCES

[1] J. Ross and B.M. Staw, "Expo 86: An Escalation Prototype," *Administrative Science Quarterly*, vol. 31, pp. 274-297, 1986.
[2] S. McConnell, *Software Project Survival Guide*. Microsoft Press, 1998.
[3] M.T. Pich, C.H. Loch, and A. De Meyer, "On Uncertainty, Ambiguity and Complexity In Project Management," *Management Science*, vol. 48, pp. 1008-1023, 2002.
[4] B.A. Kitchenham, "Modeling Software Bidding Risks," *IEEE Trans. Software Eng.*, vol. 29, no. 6, pp. 542-554, 2003.
[5] M. Jørgensen and D.I.K. Sjøberg, "Impact of Effort Estimates on Software Project Work," *Information and Software Technology*, vol. 43, no. 15, pp. 939-948, 2001.
[6] A. Huchzermeier and C.H. Loch, "Project Management under Risk: Using the Real Options Approach to Evaluate Flexibility in R&D," *Management Science*, vol. 47, no. 1, pp. 85-101, 2001.
[7] D. Kahneman and D. Lovallo, "Timid Choices and Bold Forecasts: A Cognitive Perspective on Risk Taking," *Management Science*, vol. 39, no. 1, pp. 17-31, 1993.
[8] M. Jørgensen, K.H. Teigen, and K. Moløkken, "Better Sure Than Safe? Overconfidence in Judgement Based Software Development Effort Prediction Intervals," *J. Systems and Software*, vol. 70, nos. 1-2, pp. 79-93, 2004.
[9] L. Angelis and I. Stamelos, "A Simulation Tool for Efficient Analogy Based Cost Estimation," *Empirical Software Eng.*, vol. 5, no. 1, pp. 35-68, 2000.
[10] M. Jørgensen and D.I.K. Sjøberg, "An Effort Prediction Interval Approach Based on the Empirical Distribution of Previous Estimation Accuracy," *Information and Software Technology*, vol. 45, no. 3, pp. 123-136, 2003.
[11] M. Jørgensen, "Regression Models of Software Development Effort Estimation Accuracy and Bias," *Empirical Software Eng.*, vol. 9, no. 4, pp. 297-314, 2004.
[12] W.S. Humphrey and N.D. Singpuwally, "Predicting (Individual) Software Productivity," *IEEE Trans. Software Eng.*, vol. 17, no. 2, pp. 196-207, 1991.
[13] C. Chapman and S. Ward, "Estimation and Evaluation of Uncertainty: A Minimalist First Pass Approach," *Int'l J. Project Management*, vol. 18, no. 6, pp. 369-383, 2000.
[14] P.P. Dagum, et al., "Uncertain Reasoning and Forecasting," *Int'l J. Forecasting*, vol. 11, no. 1, pp. 73-87, 1995.
[15] M. Elkjaer, "Stochastic Budget Simulation," *Int'l J. Project Management*, vol. 18, no. 2, pp. 139-147, 2000.
[16] M.R. Duffey and J.R. van Dorp, "Risk Analysis for Large Engineering Projects: Modeling Cost Uncertainty for Ship Production Activities," *J. Eng. Valuation and Cost Analysis*, vol. 2, no. 4, pp. 285-301, 1999.
[17] J. Hihn and H. Habib-Agahi, "Cost Estimation of Software Intensive Projects: A Survey of Current Practices," *Proc. Int'l Conf. Software Eng.*, pp. 276-287, 1991.
[18] T. Connolly and D. Dean, "Decomposed versus Holistic Estimates of Effort Required for Software Writing Tasks," *Management Science*, vol. 43, no. 7, pp. 1029-1045, 1997.
[19] M. Jørgensen and K.H. Teigen, "Uncertainty Intervals versus Interval Uncertainty: An Alternative Method for Eliciting Effort Prediction Intervals in Software Development Projects," *Proc. Int'l Conf. Project Management (ProMAC)*, pp. 343-352, 2002.
[20] M. Jørgensen, "Top-Down and Bottom-Up Expert Estimation of Software Development Effort," *Information and Software Technology*, vol. 46, no. 1, pp. 3-16, 2004.
[21] M. Alpert and H. Raiffa, "A Progress Report on the Training of Probability Assessors," *Judgment Under Uncertainty: Heuristics and Biases*, D. Kahneman, et al., eds., pp. 294-305, 1982.
[22] D. Kahneman, PP. Slovic, and A. Tversky, *Judgment Under Uncertainty: Heuristics and Biases*. Cambridge, U.K.: Cambridge Univ. Press, 1982.
[23] A. Tversky and D. Kahneman, "Judgment Under Uncertainty: Heuristics and Biases," *Science*, vol. 185, pp. 1124-1131, 1974.
[24] I. Yaniv and D.P.P. Foster, "Precision and Accuracy of Judgmental Estimation," *J. Behavioral Decision Making*, vol. 10, pp. 21-32, 1997.
[25] S. Lichtenstein and B. Fischhoff, "Do Those Who Know More Also Know More About How Much They Know?" *Organizational Behaviour and Human Decision Processes*, vol. 20, no. 2, pp. 159-183, 1977.
[26] H.R. Arkes, "Overconfidence In Judgmental Forecasting," *Principles of Forecasting: A Handbook For Researchers and Practitioners*, J.S. Armstrong, ed., pp. 495-415, 2001.
[27] NASA, *Manager's Handbook for Software Development*. NASA Software Eng. Laboratory, Greenbelt, Md.: Goddard Space Flight Center, 1990.
[28] B. Kitchenham, et al., "An Empirical Study of Maintenance and Development Estimation Accuracy," *J. Systems and Software*, vol. 64, no. 1, pp. 57-77, 2002.
[29] S.M. Whitecotton, 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*, vol. 76, no. 3, pp. 325-348, 1998.
[30] T. Abdel-Hamid, "Investigating the Cost/Schedule Trade-Off in Software Development," *IEEE Software*, vol. 7, no. 1, pp. 97-105, 1990.
[31] T.K. Abdel-Hamid and S.E. Madnik, "Impact of Schedule Estimation on Software Project Behavior," *IEEE Software*, vol. 3, no. 4, pp. 70-75, 1986.
[32] K. Sengupta and T.K. Abdel-Hamid, "The Impact of Unreliable Information on the Management of Software Projects: A Dynamic Decision Perspective," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 26, no. 2, pp. 177-189, 1996.
[33] H. Otway and D. von Winterfeldt, "Expert Judgment in Risk Analysis and Management: Process, Context, and Pitfalls," *Risk Analysis*, vol. 12, no. 1, pp. 83-93, 1992.
[34] M. Jørgensen, "Realism in Assessment of Effort Estimation Uncertainty: It Matters How You Ask," *IEEE Trans. Software Eng.*, vol. 30, no. 4, pp. 209-217, 2004.
[35] G. Gigerenzer, "Why the Distinction Between Single-Event Probabilities and Frequencies is Important for Psycology (and Vice Versa)," *Subjective Probabilities*, G. Wright and P.P. Ayton, eds., 1994.
[36] D. Kahneman and A. Tversky, "Variants of Uncertainty," *Cognition*, vol. 11, pp. 143-157, 1982.
[37] M. Jørgensen and K. Moløkken, "Eliminating Overconfidence in Software Development Effort Estimates," *Proc. Int'l Conf. Product Focused Software Process Improvement*, pp. 17-29, 2004.
[38] J.S. Armstrong, "Combining Forecasts," *Principles of Forecasting: A Handbook for Researchers and Practitioners*, J.S. Armstrong, ed., pp. 417-440, 2001.
[39] M. Höst and C. Wohlin, "An Experimental Study of Individual Subjective Effort Estimations and Combinations of the Estimates," *Proc. Int'l Conf. Software Eng.*, pp. 332-339, 1998.
[40] S.G. MacDonell, "Software Source Code Sizing Using Fuzzy Logic Modeling," *Information and Software Technology*, vol. 45, no. 7, pp. 389-404, 2003.
[41] I. Myrtveit and E. Stensrud, "A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models," *IEEE Trans. Software Eng.*, vol. 25, no. 4, pp. 510-525, 1999.
[42] M. Jørgensen and K. Moløkken, "Combination of Software Development Effort Prediction Intervals: Why, When and How?" *Proc. 14th IEEE Conf. Software Eng. and Knowledge Eng. (SEKE '02)*, pp. 425-428, 2002.
[43] J.W. Taylor and D.W. Bunn, "Investigating Improvements in the Accuracy of Prediction Intervals for Combinations of Forecasts: A Simulation Study," *Int'l J. Forecasting*, vol. 15, no. 3, pp. 325-339, 1999.
[44] R.C. Blattberg and S.J. Hoch, "Database Models and Managerial Intuition: 50 Percent Model + 50 Percent Manager," *Management Science*, vol. 36, no. 8, pp. 887-899, 1990.

[45] G. Rowe and G. Wright, "Expert Opinions in Forecasting: The Role of the Delphi Process," *Principles of Forecasting: A Handbook for Researchers and Practitioners,* J.S. Armstrong, ed., pp. 125-144, 2001.
[46] K. Moløkken-Østvold and M. Jørgensen, "Group Processes in Software Effort Estimation," *Empirical Software Eng.,* vol. 9, no. 4, pp. 315-334, 2004.
[47] I.L. Janis, *Victims of Groupthink.* Boston: Houghton Mifflin, 1972.
[48] N. Epley and T. Gilovich, "Just Going Along: Nonconscious Priming and Conformity to Social Pressure," *J. Experimental Social Psychology,* vol. 35, pp. 578-589, 1999.
[49] N. Kogan and M.A. Wallach, *Risk Taking: A Study In Cognition and Personality.* New York: Rinehart and Winston, 1964.
[50] J.H. Davis, T. Kameda, and M.F. Stasson, "Group Risk Taking: Selected Topics," *Risk-Taking Behavior,* J.F. Yates, ed., pp. 164-199, 1992.
[51] L.A. Maines, "An Experimental Examination of Subjective Forecast Combination," *Int'l J. Forecasting,* vol. 12, no. 2, pp. 223-233, 1996.
[52] L.M. Taff, J.W. Borchering, and J.W.R. Hudgins, "Estimeetings: Development Estimates and a Front-End Process for a Large Project," *IEEE Trans. Software Eng.,* vol. 17, no. 8, pp. 839-849, 1991.
[53] J.A. Kernaghan and R.A. Cooke, "The Contribution of the Group Process to Successful Project Planning in R&D Settings," *IEEE Trans. Eng. Management,* vol. 33, no. 3, pp. 134-140, 1986.
[54] S. Oskamp, "Overconfidence in Case-Study Judgments," *J. Consulting Psychology,* vol. 29, no. 3, pp. 261-265, 1965.
[55] I.R. Newby-Clark, et al., "People Focus on Optimistic Scenarios and Disregard Pessimistic Scenarios When Predicting Task Completion Times," *J. Experimental Psychology: Applied,* vol. 6, no. 3, pp. 171-182, 2000.
[56] C.R. Schwenk and R. Cosier, "Effects of the Expert, Devil's Advocate, and Dialectical Inquiry Methods on Prediction Performance," *Organizational Behaviour and Human Decision Processes,* vol. 26, no. 3, pp. 409-424, 1980.
[57] R.A. Cosier and G.L. Rose, "Cognitive Conflict and Goal Conflict Effects on Task Performance," *Organizational Behaviour and Human Performance,* vol. 19, no. 2, pp. 378-391, 1977.
[58] J.E. Sieber, "Effects of Decision Importance on Ability to Generate Warranted Subjective Uncertainty," *J. Personality and Social Psychology,* vol. 30, no. 5, pp. 688-694, 1974.
[59] B.W. Pelham and E. Neter, "The Effect of Motivation of Judgment Depends on the Difficulty of the Judgment," *J. Personality and Social Psychology,* vol. 68, no. 4, pp. 581-594, 1995.
[60] D.M. Grether and C.R. Plott, "Economic Theory of Choice and the Preference Reversal Phenomenon," *Am. Economic Rev.,* vol. 69, pp. 623-638, 1979.
[61] J.S. Lerner and P.P.E. Tetlock, "Accounting for the Effect of Accountability," *Psychological Bull.,* vol. 125, no. 2, pp. 255-275, 1999.
[62] C.F. Camerer and R.M. Hogarth, "The Effects of Financial Incentives in Experiments: A Review and Capital-Labor-Production Framework," *J. Risk and Uncertainty,* vol. 19, no. 1, pp. 7-42, 1999.
[63] A.L. Lederer and J. Prasad, "Software Management and Cost Estimating Error," *J. Systems and Software,* vol. 50, no. 1, pp. 33-42, 2000.
[64] K. Siegel-Jacobs and J.F. Yates, "Effects of Procedural and Outcome Accountability on Judgment Quality," *Organizational Behavior and Human Decision Processes,* vol. 65, pp. 1-17, 1996.
[65] M. Jørgensen, "A Review of Studies on Expert Estimation of Software Development Effort," *J. Systems and Software,* vol. 70, nos. 1-2, pp. 37-60, 2004.
[66] M.R. Lepper, D. Greene, and R.E. Nisbett, "Undermining Children's Intrinsic Interest with Extrinsic Rewards: A Test of the 'Overjustification' Hypothesis," *J. Personality and Social Psychology,* vol. 28, pp. 129-137, 1973.
[67] R. Hagafors and B. Brehmer, "Does Having to Justify One's Judgments Change Nature of the Judgment Process?" *Organizational Behaviour and Human Decision Processes,* vol. 31, no. 2, pp. 223-232, 1983.
[68] S.C. Hora, J.A. Hora, and N.G. Dodd, "Assessment of Probability Distributions for Continous Random Variables: A Comparison of the Bisection and Fixed Value Methods," *Organizational Behavior and Human Decision Processes,* vol. 51, pp. 223-232, 1992.
[69] P.P. Sedlmeier, *Improving Statistical Reasoning: Theoretical Models and Practical Implications.* Mahwah, N.J.: Erlbaum, 1999.
[70] K.H. Teigen and M. Jørgensen, "When 90 Percent Confidence Intervals are 50 Percent Certain: On the Credibility of Credible Intervals," *Applied Cognitive Psychology,* 2005.
[71] T.D. Wilson and N. Brekke, "Mental Contamination and Mental Correction: Unwanted Influences on Judgments and Evaluation," *Psychological Bull.,* vol. 116, no. 1, pp. 117-142, 1994.
[72] S.A. Sloman and E. Al, "Frequency Illusions and Other Fallacies," *Organizational Behavior and Human Decisions Processes,* vol. 91, pp. 296-309, 2003.
[73] S.C. Hora, J.A. Hora, and N.G. Dodd, "Assessment of Probability Distributions for Continous Random Variables: A Comparison of the Bisection and Fixed Value Methods," *Organizational Behavior and Human Decisions Processes,* vol. 51, pp. 223-232, 1992.
[74] K.H. Teigen and M. Jørgensen, "When 90 Percent Confidence Intervals are 50 Percent Certain: On the Credibility of Credible Intervals," *Applied Cognitive Psychology,* vol. 19, pp. 455-475, 2005.
[75] A. Winman, PP. Hanson, and PP. Juslin, "Subjective Probability Intervals: How to Reduce Overconfidence by Interval Estimation," *J. Experimental Psychology: Learning, Memory, and Cognition,* vol. 30, pp. 1167-1175, 2004.

Magne Jørgensen received the Diplom Ingeneur degree in Wirtschaftswissenschaften from the University of Karlsruhe, Germany, in 1988 and the Dr. Scient. degree in Informatics from the University of Oslo, Norway, in 1994. He has 10 years' industry experience as a software developer, project leader, and manager. He is now a professor of software engineering at the University of Oslo and a member of the software engineering research group of Simula Research Laboratory in Oslo, Norway. He has published many articles on software cost estimation in international journals. He has supported work on the improvement of software project estimation and has been responsible for estimation training in several software companies.