Evidence-Based Guidelines for Assessment of Software Development Cost Uncertainty

M. JØRGENSEN
magne.jorgensen@simula.no
Simula Research Laboratory,
P.O.Box 134, 1325 Lysaker, Norway
Tel.: +47 924 333 55
Fax: + 47 67 82 82 01

Abstract Several studies suggest that uncertainty assessments of software development costs are strongly biased towards over-confidence, i.e., that software cost estimates typically are believed to be more accurate than they really are. This over-confidence 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 a review of 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 models, 3) Apply structured and explicit judgment-based processes that improves with more effort and feedback, 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. 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.

Keywords: Uncertainty of software development cost, uncertainty assessment strategies, project planning, cost estimation, evidence-based guidelines.

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 (Ross and Staw 1986, p 282).

Accurate assessment of the uncertainty of software development cost estimates[1] is important (i) when deciding whether or not to embark upon a project, to support the bidding process, and (ii) to support decisions about how large the project's contingency budget should be (McConnel 1998). To illustrate the use of uncertainty assessments, consider a project concerning which the project leader believes that the median cost (the estimate) is about $100 000, i.e., he believes that there is a 50% 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

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

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 Figure 1, then, to be about 70% sure of not exceeding the budget, the project should have a contingency buffer of $25 000, i.e., $125 000 - $100 000.

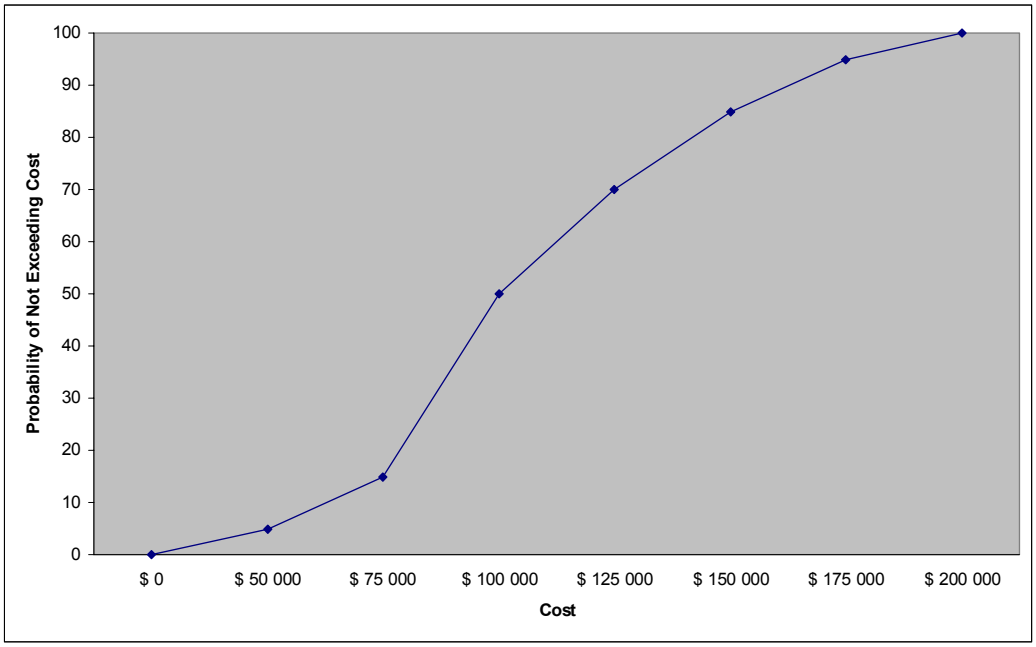

Figure 1: Cost Uncertainty Distribution

Clearly, the use of cost uncertainty assessments that reflect the underlying uncertainty will improve the budgeting process. Unfortunately, as documented in Section 3, it is easy to be over-confident about the accuracy of cost estimates. For this reason, we need to improve the way in which we conduct software cost estimation uncertainty assessments.

The paper is organized as follows: Section 2 briefly describes the state-of-practice of software cost uncertainty assessment strategies. Section 3 motivates the need for improved uncertainty assessment strategies via a review of empirical studies on unaided, human judgment-based cost uncertainty assessments. Section 4 introduces categories and concepts useful for an examination of different uncertainty assessment strategies. Section 5 describes and evaluates several extensions to, and replacements of, unaided human judgment-based strategies for assessing uncertainty. Section 6 summarizes the evaluation in preliminary evidence-based guidelines for uncertainty assessments. Section 7 concludes the paper.

2 State of Practice

Our observations (Jørgensen, Teigen et al. 2004) indicate that 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% 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% (low uncertainty) or +/-50%

(high uncertainty) cost estimate[2], or (3) Informal uncertainty terminology, e.g., that a cost estimate is a "ballpark" figure or "very rough" estimate. Of these means of describing cost uncertainty, only the use of statistical prediction intervals, i.e., minimum-maximum intervals together with a probability-based confidence level, has a well-defined semantic. Lack of precision of interpretation clearly limits the usefulness and evaluation potential of an uncertainty assessment strategy. A software developer may, for example, interpret "a ballpark figure" differently from a software manager or the customer.

To date, our observations have not revealed any use of formal cost uncertainty assessment models among software companies (Jørgensen, Teigen et al. 2004). Instead, we have observed uncertainty assessments based on unaided, intuition-based human judgment ("expert judgment"), i.e., processes based on the non-explicit and non-recoverable mental processes of software professionals. Similar observations are reported in a project survey of Hihn and Habib-Agahi (1991). They found, among other things, that the studied organization had no explicit process for incorporating risk and uncertainty assessments in the cost estimation work.

The use of unaided expert judgment to assess the uncertainty of cost estimates is understandable, given the alternative approaches provided by project management and software engineering text-books. The following two examples illustrate the problems of applying the techniques typically outlined in text-books on software project planning.

Example 1: A frequently recommended project cost uncertainty analysis technique is the "PERT System of Three Time Estimates" (PERT = Project Evaluation and Review Technique). PERT was originally developed for schedule estimation, but is also applicable to cost estimation. A description and discussion of PERT is provided by Moder et al. (1995). PERT requires that the project planner provide, for each activity, a triple assessment: minimum, most likely and maximum cost. The triple is assumed to describe the cost uncertainty distribution of that activity. PERT supports the addition of these distributions into a cost uncertainty distribution for the whole project. There are, in our opinion, three major limitations that affect the usefulness of PERT: (1) PERT requires as *input* the uncertainty of each project activity estimate. This means that PERT implicitly assumes that the project planners, e.g., the software professionals, are able to provide proper minimum and maximum activity cost estimates. The validity of this assumption is debatable, as we will document in Section 3. (2) Some of the assumptions that enable the addition of uncertainty distributions, e.g., independence between activity uncertainty distributions, are unrealistic in software contexts. Surprisingly, there is frequently not much support on how to treat relationships between individual uncertainties when applying the PERT technique. For example, "The Guide to the Project Management Body of Knowledge" (2000) applies the "PERT System of Three Times Estimates" as part of their quantitative risk analysis, but provides no support on how to manage dependencies between the individual uncertainties. (3) PERT provides limited support on how to include cost uncertainties not related to a particular activity, e.g., uncertainty related to unexpected events with effect on most activities.

Example 2: One of the most popular university text-books on software engineering is (Sommerville 2001). That text-book suggests that existing formal cost estimation models can be used to assess uncertainty: *"The estimator should develop a*

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

range of estimates (worst, expected and best) rather than a single estimate. The costing formula should be applied to all of these." As we understand this recommendation, it implies that we should use the worst, expected and best value of the parameters of the costing formula, e.g., the "COCOMO II" costing formula (Boehm, Abts et al. 2000), to achieve the worst, expected, and best case cost. Sommerville's recommendation has, as far as we know, not been empirically evaluated, is not in common use, and may have major limitations, e.g.: (1) The size of the software is an important input to most formula-based estimation models. It is not obvious that it is easier to assess the minimum and maximum size of software more accurately than the minimum and maximum cost. In fact, the opposite, i.e., that size may be more difficult to assess than cost, was found in the study by Hihn and Habib-Agahi (1991). (2) The costing formulae were developed to estimate the most likely cost, so the variables included the models are not necessarily those important for the uncertainty of a cost estimate, (3) Most software organizations do not use costing formulae (Moløkken and Jørgensen 2003; Jørgensen 2004b).

There have been a few attempts to build other types of formal cost uncertainty model, e.g., (Humphrey and Singpuwally 1991; Angelis and Stamelos 2000; Jørgensen and Sjøberg 2003; Jørgensen 2004a), and several general statistical uncertainty models for estimation and planning purposes, e.g. (Dagum, Galper et al. 1995; Chapman and Ward 2000). In addition, there are many frameworks and tools supporting a structured elicitation and combination of project uncertainties, e.g., (Duffey and van Dorp 1999; Elkjaer 2000; Kitchenham, Pickard et al. 2003). Software organizations should deal with cost uncertainty at the project portfolio level, as well as at the individual level. Stamelos and Angelis (2001) developed a model for that purpose. To the best of our knowledge, none of these models, frameworks or tools is in common use and there seem to be strong limitations to their usefulness. For example, most formal uncertainty models seem to require more data, or data on other formats, than most software organizations are able to provide.

3 Over-Confidence

As suggested in Section 2, unaided human judgement-based cost uncertainty assessments may be the approach most frequently applied by software professionals. This would not be problematic if those uncertainty judgments were accurate. Unfortunately, several studies reports systematic underestimation of cost uncertainty, i.e., systematic over-confidence, when based on unaided human judgment:

- Connolly and Dean (1997) report that the actual effort used by student programmers to solve programming tasks fell inside their 98% confidence effort prediction (minimum-maximum) intervals in only about 60% of the cases, i.e., the effort prediction intervals were much too narrow to reflect 98% confidence. Explicit attention to, and training in, establishing good minimum and maximum effort values did increase the proportion inside the prediction intervals to about 70%, which was still far from the required 98%.
- Jørgensen, Teigen and Moløkken (2004) studied the software development activity estimates of 195 student projects activities and 49 industry project activities. The effort prediction intervals of the activities of the student projects were based on a 90% confidence level, and included only 62% of the actual effort values. The effort prediction intervals of the activities of the industrial projects

were based on the confidence level "almost certain", and included only 35% of the actual effort values, i.e., a strong underestimation of uncertainty.

- Jørgensen and Teigen (2002) conducted an experiment in which 12 software professional were asked to provide 90% confidence effort prediction intervals on 30 previously completed 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% confident", the professionals included, on average, only 64% of the actual effort values on the first 10 tasks (Task 1-10), 70% on the next 10 task (Task 11-20), and, 81% on the last 10 tasks (Task 21-30). In other words, even after 20 tasks with feedback after each task, there was a systematic bias towards over-confidence.
- Jørgensen (2004c) studied the cost prediction 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 fourteen 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. Only 43% 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 over-confidence similar to that in the software domain. See, for example the studies described in (Tversky and Kahneman 1974; Alpert and Raiffa 1982; Kahnemann, Slovic et al. 1982; Yaniv and Foster 1997). Lichtenstein and Fischhoff (1977) report that the level of over-confidence seems to be unaffected by differences in intelligence and expertise, so we should *not* expect the level of over-confidence to be reduced with greater experience. Arkes (2001) provides a recent overview of studies from different domains on over-confidence, strongly supporting the over-confidence claim.

There is, therefore, strong evidence of a systematic bias in human judgment towards underestimation of the uncertainty of software projects. Potential reasons for this over-confidence are as follows:

- *Interpretation difficulties*. Software professionals have understandable problems interpreting the concepts "90% confident", "90% probable" or "almost certain" when historical data are sparse and statistical skills are poor. In (Jørgensen, Teigen et al. 2004), we report results from an experiment where the estimators did not, on average, provide different effort minimum-maximum intervals when 70%, 90% and 99% confident. Clearly, if a software estimator is 99% confident to include the actual effort in an effort minimum-maximum interval, this interval should, on average, be much wider than when 90% or 70% 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 (Jørgensen, Teigen et al. 2004), i.e., the software companies were, as far as can be gleaned from our observations, not even aware of the degree of over-confidence of their uncertainty assessments. One important obstacle to the efficient evaluation of uncertainty assessment seemed to be that the activity structure used for estimation typically was different from that used for the logging of actual effort, i.e., the organizations' own administrative systems did not enable such evaluations.
- *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 (Jørgensen, Teigen et al. 2004) we 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 over-confident. 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.

The available evidence therefore suggests that software companies rely on unaided human judgment alone in the assessment of cost estimation uncertainty. These uncertainty assessments are, on average, strongly over-confident. Over-confident uncertainty assessments may lead to poor project plans, and consequently, poor project performance. There is consequently a need to improve the ways in which we conduct uncertainty assessments in software projects.

4 Uncertainty Assessments

For the purpose of discussing strengths and weaknesses of uncertainty assessment strategies, as we do in Section 5, we believe it is useful to clarify important uncertainty concepts in terms of probability theory (Section 4.1), to introduce some categories of uncertainty (Section 4.2), and two broad categories of uncertainty assessment process (Section 4.3). To support those software professionals who assess cost estimation uncertainty, it is useful to understand the underlying mental processes of judgment-based assessments. We discuss the (limited) knowledge about this topic in Section 4.4. Section 4.5 introduces measures of uncertainty assessments.

4.1 *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% confidence effort prediction interval [$20 000; $40000]. This prediction interval means that it believed that it is a 90% probability that the actual cost is higher than or equal to $20000, and lower than or equal to $40000.

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, therefore introduce *frequency* of a class of similar events as a (fallible) 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%, which is interpreted as the average objective probability of including the actual cost in the cost minimum-maximum interval. 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% probability of including the actual costs, we say that the project manager is over-confident.

Both subjective and objective probabilities may be described as based on two different sources: (i) The *inherent* uncertainty regarding cost usage in a software project, and, (ii) 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.

Sometimes we use the description 'assessment of uncertainty of cost estimates' and sometimes 'assessment of uncertainty of costs'. These two descriptions are similar description of the same subjective distribution. They are both based on the same concept of probability-based uncertainty assessments and can be derived from the same cost uncertainty distribution, e.g., the distribution depicted in Figure 1. The only difference is that "assessment of uncertainty of cost estimates" relates the uncertainty to the estimate of most likely cost (or median cost), and the other does not.

4.2 *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 (Pich, Loch et al. 2002; Kitchenham, Pickard et al. 2003). The categories are: *Normal variance, Known risks, Unforeseen events,* and, *Flexibility of outcome and process*. In comparison with previous frameworks the main difference is the introduction of the uncertainty category *Flexibility of outcome and process*. Our interpretation of the categories is as follows:

- *Normal variance*: Uncertainty of known activities resulting from what is considered to be "normal variation"[3] in a project's software development performance. Normal variation is typically a result of many small uncertainties.
    - *Example of treatment*: Minimum-maximum cost intervals of project activities may be applied to describe the normal variation.
- *Known risks*: Uncertainty that results from the potential occurrence of foreseen events (positive and negative) that are analyzed at the time of estimation. These event (if occurring) impacts on the project's performance significantly, i.e., the impact is considered to be outside the "normal variance".
    - *Example of treatment*: Many projects develop and maintain a list of risks as part of their management processes, which typically includes information about the risks, their probability, their potential impact on the project, and plans for how the risk is to be managed.
- *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". According to Asher (1993) and van Genuchten (1991) over-optimism regarding the degree of unforeseen events may be the main contributor to cost over-runs in development projects.
    - *Example of treatment*: Project leaders may introduce a cost buffer (a contingency buffer) allocated to unexpected events.

[3] This concept is similar to the "normal variance"-concept of "Statistical Process Control" within the Total Quality Management approach, see Montgomery, D. C. (2000). Introduction to Statistical Quality Control. New York, Wiley.

- *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 reducing the project specification, e.g., 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 (Jørgensen and Sjøberg 2001a). The impact of this category is similar to the impact on project management of the "option of corrective actions" discussed in (Huchzermeier and Loch 2001).
    - *Example of treatment*: The size of the project's contingency cost buffer should reflect the flexibility of the process and outcome. A software prototyping project may, for example, not need any contingency buffer at all if it is accepted that the product can be completed with much less functionality and much lower quality than was originally planned.

When software professionals search for relevant information about uncertainty, they may search for all of these types of uncertainties. However, although we may be able to identify and collect most of the relevant information about uncertainty,, by, for example, following the processes described in (Hall 1998), it is not obvious how to combine the information into an assessment of the *total* uncertainty of a project's cost estimate.

4.3 *Inside vs Outside Views*

Kahneman and Lovallo (1993) separate human judgment processes into two categories: "inside view"-based and "outside view"-based. Applied in the context of cost estimation uncertainty assessment the categories can be described and evaluated 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, 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. This may be problematic in many cases. For example, a correct combination of activity cost prediction intervals (minimum-maximum intervals) requires a formalization of the relations between the prediction intervals and an application of complex statistical theory to add the uncertainty distributions. Frequently, to support this process, simulation techniques are applied, e.g., (Elkjaer 2000). Unaided human judgment, i.e., software professionals "intuition" of the uncertainty of the project's cost estimate, can hardly be based on complex statistical calculations or computational-intensive simulation-based techniques. It may, however, still be possible for software professional's predictions to be based on an "inside view" of the project, provided that proper heuristics exist and are applied (Gigerenzer and Todd 1999; Jørgensen and Sjøberg 2001b). One advantage of inside view cost uncertainty assessments is that they can be based on analyses that are conducted as part of the project planning process, e.g., as part of the risk analysis process. As pointed out in (Edwards and Moores 1994), integration with the planning process may be an important factor for success in the use of prediction methods.
- *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-based uncertainty assessment may frequently be simpler than an inside view-based one. Instead of developing a model for connecting the internal uncertainties of a project, as in the inside view, the main step is to collect previously completed projects with similar uncertainty properties, e.g., projects of similar size, applying similar estimation methods, similar skill of the estimators, etc.. When a set of similar projects is identified, the level of uncertainty of these projects, e.g., measures of the estimation error, may be used to derive the uncertainty of the current project. A prerequisite for accurate uncertainty assessments based on an outside view is that project properties important for the level of uncertainties are known and that there is relevant information about previous projects available. The stored information about completed projects may, however, in many organizations be sparse and not easily accessible, as described in (Jørgensen 2004c). In such a case, an informal recall of previous projects from memory, or the application of an inside view, may be the only alternatives.

4.4 *Judgment of Uncertainty*

For the support, as opposed to the replacement, of human judgment-based uncertainty assessments it is useful to understand the ways in which software professionals typically judge cost estimation uncertainty, i.e., the mental processes underlying uncertainty judgments. Unfortunately, there seems to be no commonly accepted theory on processes of expert judgment upon which we can base this understanding. According to Brown and Siegler (1993) psychological research on real-world quantitative expert estimation *"has not culminated in any theory of estimation, not even in a coherent framework for thinking about the process"*.

One reason for the lack of theory is that human judgment processes may be unconscious ("tacit") and difficult to access. This is illustrated in (Jørgensen 2004c). In that study we recorded, transcribed and analyzed the estimation discussions of seven professional development teams. Each team estimated most likely, minimum and maximum effort of two projects. In total, we analyzed 180 pages of transcribed text. *None* of the seven teams formulated processes or arguments that were even close to an explicit uncertainty assessment process. In fact, the most explicit uncertainty assessment process we found was the "rule-of-thumb" used by one of the project managers: *"maximum effort is typically two times the minimum effort"*. The other types of uncertainty arguments were more or less based on "gut feeling", e.g., *"I want to decrease the minimum and increase the maximum effort"* or "*the maximum [effort] is the most pessimistic outcome I believe in"*.

The mental processes leading to software professionals' uncertainty assessments are, therefore, typically hidden and can only be analyzed indirectly, e.g., by the analysis of indicators that affect the assessments (Stanovich 1991, p 117-121) . One example of a potential indicator of cost estimation uncertainty is the development skill (the "know-how") of the software professionals estimating the project: the higher the "know-how", the lower the cost estimation uncertainty. Unfortunately, without explicit uncertainty assessment processes and systematic feedback, establishing a proper weighting of indicators may be problematic and some indicators may be strongly over-weighted. For example, in (Jørgensen 2004c) we report: *"The level of over-confidence was higher in situations where at least one of the team members assessed his/her knowledge to be very high…."* In other words, our results suggest

that the cost uncertainty assessment was, directly or indirectly, affected by the perceived level of skill, and that its importance was over-weighted.

Awareness of properties of unaided, judgment-based cost uncertainty assessment is important when analyzing and predicting benefits from supporting strategies. For example, more effort in the risk analysis process, introduction of financial incentives for accuracy, more experienced developers, and increased awareness about the over-confidence bias, do not necessarily lead to more accurate judgments. In fact, studies suggest that unconscious assessments may be even more over-confident when a greater amount of information is available (Oskamp 1965; Whitecotton, Sanders et al. 1998). According to (Wilson and Brekke 1994) mental contamination, e.g., the over-confidence bias, is difficult to avoid when the processes are unconscious or uncontrollable.

4.5 *Measures of Cost Uncertainty Assessments*

A proper evaluation of uncertainty assessments requires, as stated in Section 2, that it is clear how to interpret the measures. From that follows that it is easier to evaluate "90% confident of not exceeding $125 000" than it is to evaluate "almost sure of no high cost overruns". The measures introduced in this section are based on the understanding of probability and frequency as described in Section 4.1. For the purpose of this paper we apply mainly the measures "Hit rate" (frequency) and "Relative Width" (RWidth) of cost uncertainty intervals. We provide a more comprehensive list of measures and discussion in (Jørgensen, Teigen et al. 2004).

Evaluating cost uncertainty assessment is more difficult than evaluating cost estimation accuracy. Whereas the accuracy of individual cost estimates can be assessed by comparing them to actual effort, individual cost uncertainty assessments have no obvious corresponding actual values. In the long run, however, a *K*% confidence level should correspond to a proportion of correct assessments ("Hit rate") similar to K%. For example, given a number of cost intervals with 90% confidence, we should expect that about 90% 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 lower than the confidence level, we observe overconfidence and if it is higher we observe under-confidence.

The following definitions of "Hit rate" and "RWidth" are based on uncertainty assessments on the cost prediction interval formal, e.g., that it is believed to be "90% 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% probably that the actual cost is less than $ 125 000".

Hit rate: We measure the hit rate as:

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

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.

RWidth: Of two sets of cost prediction intervals with the same "Hit rate", the set with narrower intervals is more informative, and also indicative of a higher level of expertise, or more efficient use of the uncertainty information, than the wider

intervals, see discussion on inherent and lack-of-knowledge based uncertainty in Section 4.1. For example, a person who is only guessing may end up with an adequate hit rate if 90% of his or her 90% prediction intervals are extremely wide. To compare prediction intervals for tasks of different magnitudes we apply the *relative* width of the interval:

$RWidth_i = (Max_i - Min_i) / Est_i$, where Est_i is the estimated (most likely) cost of task i.

5 Uncertainty Assessment Strategies

This section reviews the following replacements of, and extensions to, unaided, intuition-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). There is no clear-cut boundary between the strategies and actual uncertainty assessment processes may include elements of many strategies. We only review strategies where we could find attempts of empirical evaluation. This means, for example, that the suggestions of Sommerville described in Section 2 and the "rules of thumb" suggested by NASA (1990) and others are not discussed.

As stated earlier, the goal of this paper is to review changes in the cost uncertainty assessment process of software projects, potentially leading to improvements. This, obviously, requires an understanding of what we mean by improvement. For the purpose of this paper we define improvements as *cost estimation uncertainty assessments that accord better with the actual (objective) uncertainty than unaided, intuition-based uncertainty assessments*. As described in Section 2, unaided, intuition-based uncertainty assessments may be the most common cost uncertainty assessment strategy.

5.1 *Formal Uncertainty Models*

As reported in Section 3, several studies report systematic underestimation of uncertainty when the assessments are based on unaided human judgment. Formal models are not subject to the same 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 uncertainty but not for the estimation of most likely cost it is not included in the model. The empirically evaluated indirect models are (i) use of cost prediction intervals of regression models of most likely cost (Angelis and Stamelos 2000; Jørgensen and Sjøberg 2003) and (ii) prediction intervals based on bootstrapping of analogy-based cost estimation models (Angelis and Stamelos 2000).
- *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 (Jørgensen and Sjøberg 2003), and regression models of the estimation error (Jørgensen 2004a).

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

Angelis and Stamelos (2000) 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 model were able to provide unbiased prediction intervals, i.e., about 95% of the actual effort values were included in the 95% 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% 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., (Kitchenham, Pfleeger et al. 2002; Jørgensen 2004a), and the experience-based minimum-maximum cost interval span for software projects described in (NASA 1990), the model-based effort prediction intervals in (Angelis and Stamelos 2000) seems to be unrealistically wide. Much of the 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" (lack of knowledge) and not "project uncertainty" (inherent uncertainty).

Additional findings supporting the view of inefficient formal uncertainty models are provided in (Jørgensen and Sjøberg 2003). In that study we 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*). We 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 towards intervals that are too narrow.

Similar results were found when evaluating regression models of estimation error (*direct model*) (Jørgensen 2004a). In that paper we 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. (1998), 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 model may be useful, as applied in the system dynamic models described in (Abdel-Hamid

and Madnik 1986; Abdel-Hamid 1990; Sengupta and Abdel-Hamid 1996). However, system dynamics tools and notations have been available for years, without much use in the software industry. A potential reason for that is that inside view-based uncertainty relationships of software projects are too complex to formalize.

Whether or not formal uncertainty assessment models will be able to replace uncertainty assessments based on software professionals' judgment depends on many factors, e.g., the availability of historical data and uncertainty relevant information, and the skills of the software professionals. The currently available evidence suggests that use of human judgment-based cost uncertainty assessment is a reasonable choice, given the poor efficiency of the formal models. In particular, human judgment may be necessary in situations where there are so-called "broken-leg" indicators that are important for the assessment of uncertainty, i.e., seldom-occurring events that cannot easily be included in a formal model.

5.2 *Formalization of the Uncertainty Assessment Processes*

In some professional domains, e.g., the management of nuclear reactors, there is, according to (Otway and von Winterfeldt 1992), a trend towards 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 to instruct software professionals to follow these steps:

1) Identify previously completed projects with similar uncertainty characteristics to the one to be assessed, e.g., with similar size, technology, etc. Identify at least 10 completed projects, if necessary through lowering the requirements to the similarity of the included projects.
2) Recall, by using historical data or memory, the estimation errors, i.e., as substitutes of the estimation uncertainty, of these projects.
3) Draw up a distribution of estimation errors based on the estimation error of similar, previously completed, projects.
4) Apply the distribution to assess the uncertainty of the current project, on the assumption that the accuracy of future cost estimates will follow closely the pattern of the historical estimates. For example, if 2 out of 10 projects had more than 40% cost over-run, assume that there is a 20% probability of not exceeding the most likely effort by more than 40%.
5) Adjust the uncertainty assessment. However, only allow adjustments if it is possible to provide an explicit and valid argument, based on differences from the set of previously completed projects.

These steps are an implementation of the simplest variant of the formal uncertainty models described in (Jørgensen and Sjøberg 2003). The main differences to the application of 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 seem to be better assessors when asking "how frequently X happens" instead of "how probable is X"

(Gigerenzer 1994; Sloman, Over et al. 2003). While providing frequencies imposes an outside view, examining historical data, providing subjective probabilities frequently implies an inside view, where the individual uncertainties are analyzed and combined. As reported in (Kahneman and Tversky 1982), the inside view easily lead to over-confidence.

The above process was evaluated in a medium-large Norwegian company (Jørgensen and Moløkken 2004). Nineteen realistically composed estimation teams of three or four software professionals estimated the most likely cost and provided effort prediction intervals of the same software project. Ten of the teams, Group A teams, received no instructions, i.e., there was no formalization of their unaided, intuition-based uncertainty assessment process. The remaining nine teams, Group B teams, followed a process similar to that above, with one important difference. We made no restrictions on the adjustments, i.e., step 5. We found that the Group B teams provided much wider cost prediction intervals when evaluated after step 4) (mean RWidth of 1.1 versus 0.65), but not after step 5)! Allowing unrestricted adjustment of the cost prediction intervals had as a consequence that the Group B team reduced their estimated maximum costs to the same levels as the Group A teams. Interestingly, the Group B teams retained the history-based minimum cost values, which were systematically higher than those of the Group A teams. In other words, the Group B teams accepted that the historical fact that they had hardly ever used less than 75% of the estimated cost on similar projects should have an impact on the minimum cost value. They did, however, not interpret the historical error distribution in the same way when determining the maximum value, i.e., they thought their estimate was much better than the estimation error of similar projects indicated that it would be. Based on the earlier findings on systematic over-confidence when assessing cost uncertainty, we interpret this finding as a warning against unlimited adjustments of the outcome of formalized uncertainty assessment processes. This may easily introduce the strong bias towards over-confidence. Overall, however, we believe that the suggested formalized five-step approach is promising way to combine the benefits of models and experts.

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 (Elkjaer 2000; Kitchenham, Pickard et al. 2003). Typically, these frameworks apply simulation techniques, e.g., the Monte Carlo simulation, to implement the complex adding of inter-connected uncertainty distributions. Unfortunately, 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 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, Armstrong (2001) reports, based on 30 empirical studies, that predictions based on the mean value of individual predictions were on average 12.5% 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., (Höst and Wohlin 1998; Myrtveit and Stensrud 1999; MacDonell and Shepperd 2003).

Strategies for, and benefits of, the mechanical combination of software development cost *uncertainty* assessments have, as far as we know, only been studied in (Jørgensen and Moløkken 2002). According to (Taylor and Bunn 1999) there are not many studies at all, i.e., regardless of domain, on the topic of combining uncertainty assessments. The study described in (Jørgensen and Moløkken 2002) evaluated three combination strategies: (1) Average of the individual minimum and maximum values, and (2) Maximum and minimum of the individual maximum and minimum values, and (3) Group process (discussion) based prediction intervals. (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 towards 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, (Blattberg and Hoch 1990). 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*

As far as we have observed, software organizations' work on estimation and uncertainty assessments is typically conducted in groups and follows one of the following two variants: (1) The project leader collects and integrates estimates and uncertainty assessments from the project members, without any group discussion. (2) The estimates and uncertainty assessments are derived from group discussion facilitated by the project leader. The structure imposed on the group work may vary a lot, from formal Delphi-based processes (Rowe and Wright 2001), to more unstructured processes (Moløkken and Jørgensen 2004).

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 (Janis 1972), 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 (Epley and Gilovich 1999). 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 (over-confident) decisions than each individual member (Kogan and Wallach 1964). More recent studies suggest that the more general effect of group-work is the "polarization effect" (Davis, Kameda et al. 1992), 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 (Maines 1996) 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., (Kernaghan and Cooke 1986; Taff, Borchering et al. 1991). It is therefore not possible 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 (Moløkken 2002) and, as a consequence, to more realistic estimates of most likely cost than individual estimates. Greater realism in estimates of most likely cost may contribute 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 (Moløkken 2002), 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 over-confidence when the additional information is irrelevant or only slightly relevant (Oskamp 1965; Whitecotton, Sanders et al. 1998).
- Group work may 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. This may lead to improved cost uncertainty assessments if the uncertainty assessment is based on, for example, the uncertainty assessment process described in Section 5.2.
- Group work may lead to a higher degree of over-confidence if the group members assess the cost uncertainty of their own development work (Newby-Clark, Ross et al. 2000). Then, the desire of appearing skilled by exhibiting high confidence may hinder accuracy in cost uncertainty assessments (Jørgensen, Teigen et al. 2004).
- Group work may lead to a higher degree of evaluation. This can lead to more, no change in, or less over-confidence in uncertainty assessments, dependent on the strategy applied, see Section 5.5.
- Group work may benefit from the use of a "devil's advocate", i.e., a person allocated to the role of arguing for alternative views (Schwenk and Cosier 1980). 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?"

An example of a proper group work-based uncertainty assessment process, developed for dam building purposes, which implements many of the relationships reported in empirical studies, is described in (Baecher). Adapted to our purposes the main steps of that process may be described as follows:

1) Identify the cost uncertainties to be assessed.
2) Select a panel of experts displaying a balanced spectrum of expertise. The experts should be able to argue their point of view and be open to other points of view.

3) Refine assessment issues in discussions with the panel.
4) Expose the experts to a short training and motivation session on concepts, objectives, methods, and, common errors made.
5) Elicit the uncertainty assessment of individual experts on issues pertinent to their individual expertise.
6) Allow the group members to interact, supported by a facilitator, to explore hypotheses, points of view, etc., with the goal of aggregating the assessments and resolving the breadth of opinion.
7) Document the process well and communicate the results back to the panel of experts.

This process may be useful for deriving the benefits of using groups, while avoiding most of the pitfalls. However, work should be conducted to evaluate the process.

5.5 *Improved Motivation*

There are a variety of motivation-based uncertainty assessment 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., (Sieber 1974; Armstrong, Denniston Jr. et al. 1975; Cosier and Rose 1977). 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" (Pelham and Neter 1995). 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., (Grether and Plott 1979), show no effect, or a positive effect, from increased motivation on performance. Lerner and Tetlock (1999) 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 (Camerer and Hogarth 1999), suggest that incentives, in general, have positive effects. The complexity of identifying the conditions for benefits from motivational mechanisms to evaluate performance in job situations is well illustrated in (Lindsay and Ehrenberg 1993).

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 (Lederer and Prasad 2000) on the effect of higher motivation on cost estimation accuracy. That study, (Lederer and Prasad 2000), found positive effects of increased accountability through performance evaluation. In fact, they found that performance evaluation was the *only* means of improving 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 (Jørgensen and Sjøberg 2001a). In our opinion, a likely explanation for the benefits of higher accountability in (Lederer and Prasad 2000) 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 towards uncertainty assessment: 1) There must be an explicit uncertainty assessment process where the accuracy improves with more effort, or, 2) There must be 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 several empirically validated findings that are useful for the design and tailoring of motivational mechanisms:

- The motivational mechanisms should be directed towards the process more than the outcome (Siegel-Jacobs and Yates 1996). 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 (Tetlock 1993a). 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 (Jørgensen 2004b). Optimally, the only goal of the uncertainty assessment should be accuracy and 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 (Lepper, Greene et al. 1973).
- 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 (Hagafors and Brehmer 1983), 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*

Several studies on human judgment, e.g., the study by Hora et al. (1992), report that over-confidence is robust to differences in framing. Other studies, however, report that the framing can be essential in stochastic problems, e.g. (Sedlmeier 1999). Hence, there is considerable variation in the results of studies on human judgment.

The only study on human judgment in the context of software development is, as far as we know, (Jørgensen and Teigen 2002). In that study we showed that the framing did, indeed, have an important impact in situations with immediate feedback and many similar cost uncertainty assessment tasks. The experiment involved 29 software professionals who estimated the most likely cost and the uncertainty of that cost estimate. Each participant estimated 30 tasks, applying an experience base of previously completed tasks. Feedback about actual cost was provided after each task estimate. Two different uncertainty assessment framings were compared. Following an estimation of the most likely effort half of the participants (Group A) were asked to: (1) *Provide a minimum-maximum effort interval that include the actual effort with a probability of 90%* (traditional framing), and half of the participants (Group B) were asked to (2) *Assess the probability that the actual effort is inside the interval 50% of most likely effort - 200% of most likely effort* (alternative framing). The alternative framing is similar to that proposed by (Seaver, Winterfeldt von et al. 1978). The groups performed surprisingly differently. Those who received the traditional framing (Group A) showed the usual pattern of over-confidence, e.g., the average "hit rate" was 58% on the first 10 tasks when it should have been about 90%. Those who received the alternative framing (Group B), however, achieved a very close correspondence between probabilities of including the actual effort (confidence level) and the "hit rate", e.g., average "hit rate" for the first 10 tasks was 83%, which was exactly the same as their average "confidence level".

We should not be led to believe that changes in the framing solve all the inherent problem of uncertainty assessment. The experimental design may, to some extent, have been unrealistic and biased in favor of the alternative framing. For example, it is not common that a software organization has an experience database with many previous projects, and provides immediate feedback. It is, nevertheless, interesting to analyze how easily those who received the alternative framing used the historical data and the feedback to provide accurate confidence levels of a pre-set minimum-maximum interval compared to those who received the traditional framing and provided minimum-maximum values of a pre-set confidence level. At least, we can interpret the results as suggesting that a match between the format of the available information and the presentation of the uncertainty assessment problem simplifies the mental work and improves the quality.

6 Preliminary Guidelines

This section condenses what we believe are the major uncertainty assessment results 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.

Guidelines:

1. **Do not Rely Solely on Unaided, Intuition-Based Processes**: There is strong evidence suggesting that unaided, intuition-based software cost estimation uncertainty assessments are inaccurate and systematically biased towards over-confidence. **Evidence and argumentation**: Section 3.
2. **Do not Replace Expert Judgment With Formal Models**: 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. However, formal models may not apply uncertainty assessments as efficiently as software professionals. For example, formal cost prediction interval models seem to yield intervals that are so wide as to be meaningless, to compensate for lack of uncertainty information specific for a single project. It may be beneficial to combine uncertainty assessments from formal models and human judgment, but this approach has so far not been evaluated properly. **Evidence and argumentation**: Section 5.1.
3. **Apply Structured and Explicit Judgment-Based Processes that Improves with More Effort and Feedback**: The process should be structured and explicit to enable review of the quality of assessment process. In addition, a process should be selected in such a way that the accuracy improves with more assessment effort and feedback. The process described in Section 5.2 is, we believe, an example of an explicit process that improves with more effort, e.g., with the collection of more historical data. Inside view-based strategies, e.g., identification and assessment of a project's activity based uncertainties, seem to be vulnerable to higher over-confidence with the collection of more information. **Evidence and argumentation**: Sections 4, 5.2 and 5.3.
4. **Apply Strategies Based on an Outside View of the Project**: 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 and argumentation**: Section 4.
5. **Combine Uncertainty Assessments From Different Sources Through Group Work, Not Through Mechanical Combination:** 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. **Evidence and argumentation**: Sections 5.3 and 5.4.

6. **Use Motivational Mechanisms With Care and Only If It Is Likely That More Effort Lead to Improved Assessments.** 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 towards 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 and argumentation**: Section 5.5.
7. **Frame the Assessment Problem to Fit the Structure of the Uncertainty Relevant Information and the Assessment Process.** 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% confidence effort prediction interval, i.e., a minimum-maximum interval where the estimator believe that there is a 90% probability of including the actual effort, of a 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 P%, he or she may investigate the previous projects and find that, for example, 20% of the projects exceeded the budget by more than P%, i.e., the historical information together with the assessment process fit the framing of the assessment problem. **Evidence and argumentation**: Section 5.6.

7 Conclusions

Software project cost estimation uncertainty assessment may frequently be based on expert judgment, i.e., unaided, intuition-based processes. Unfortunately, such uncertainty assessments have been shown to be systematically over-confident, i.e., they underestimate the uncertainty. Over-confident cost estimation uncertainty assessment may lead to poor project management. There have been several studies on how to improve uncertainty assessments, both in the software domain and other domains. The review presented in this paper synthesizes the 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. There is also a need to investigate other strategies that have the potential for improving cost estimation uncertainty assessments, e.g., the role of training and the use of cognitive de-biasing techniques.

In spite of the preliminary state of the guidelines, it is our belief that the current version constitutes a practical and useful guide for software organizations when designing their own cost uncertainty assessment process. In addition, we believe that future research on software cost estimation uncertainty assessment may benefit from the categories we introduced as preparation to the review, e.g., the distinction between inside and outside view-based strategies for cost uncertainty assessment.

Acknowledgement: Thanks to Mr. Kjetil Moløkken at Simula Research Laboratory, the Professor in Psychology at the University of Oslo, Karl Halvor Teigen, and Mr. Chris Wright at Cambridge Language Consultants, for their useful suggestions and interesting discussions.

References

Abdel-Hamid, T. (1990). "Investigating the cost/schedule trade-off in software development." IEEE Software **7**(1): 97-105.

Abdel-Hamid, T. K. and Madnik, S. E. (1986). "Impact of schedule estimation on software project behavior." IEEE Software **3**(4): 70-75.

Alpert, M. and Raiffa, H. (1982). A progress report on the training of probability assessors. Judgment under uncertainty: Heuristics and biases. D. Kahneman, P. Slovic & A. Tversky. Cambridge, Cambridge University Press**:** 294-305.

Angelis, L. and Stamelos, I. (2000). "A simulation tool for efficient analogy based cost estimation." Journal of Empirical Software Engineering **5**(1): 35-68.

Arkes, H. R. (2001). Overconfidence in judgmental forecasting. Principles of forecasting: A handbook for researchers and practitioners. J. S. Armstrong. Boston, Kluwer Academic Publishers**:** 495-515.

Armstrong, J. S. (2001). Combining Forecasts. Principles of forecasting: A handbook for researchers and practitioners. J. S. Armstrong, Kluwer Academic Publishers**:** 417-439.

Armstrong, J. S., Denniston Jr., W. B., et al. (1975). "The use of the decomposition principle in making judgments." Organizational Behavior and Human Decision Processes. **14**(2): 257-263.

Asher, W. (1993). "The ambiguous nature of forecasts in project evaluation: Diagnosing the over-optimism of rate-of-return analysis." International Journal of Forecasting **9**(1): 109-115.

Baecher, G. B. Expert elicitation in geotechnical risk assessments, USAC Draft Report, Univ. of Maryland, College Park, MD.

Blattberg, R. C. and Hoch, S. J. (1990). "Database models and managerial intuition: 50% model + 50% manager." Management Science **36**(8): 887-899.

Boehm, B., Abts, C., et al. (2000). Software cost estimation with Cocomo II. New Jersey, Prentice-Hall.

Brown, N. R. and Siegler, R. S. (1993). "The role of availability in the estimation of national populations." Memory and Cognition **20**: 406-412.

Camerer, C. F. and Hogarth, R. M. (1999). Journal of Risk and Uncertainy **19**(7-42).

Chapman, C. and Ward, S. (2000). "Estimation and evaluation of uncertainty: a minimalist first pass approach." International Journal of Project Management **18**(6): 369-383.

Connolly, T. and Dean, D. (1997). "Decomposed versus holistic estimates of effort required for software writing tasks." Management Science **43**(7): 1029-1045.

Cosier, R. A. and Rose, G. L. (1977). "Cognitive conflict and goal conflict effects on task performance." Organizational Behaviour and Human Performance **19**(2): 378-391.

Dagum, P., Galper, A., et al. (1995). "Uncertain reasoning and forecasting." International Journal of Forecasting **11**(1): 73 - 87.

Davis, J. H., Kameda, T., et al. (1992). Group risk taking: Selected topics. Risk-taking behavior. J. F. Yates. New York, Wiley**:** 164-199.

Duffey, M. R. and van Dorp, J. R. (1999). "Risk analysis for large engineering projects: Modeling cost uncertainty for ship production activities." Journal of Engineering Valuation and Cost Analysis **2**(4): 285-301.

Edwards, J. S. and Moores, T. T. (1994). "A conflict between the use of estimating and planning tools in the management of information systems." European Journal of Information Systems **3**(2): 139-147.

Elkjaer, M. (2000). "Stochastic Budget Simulation." International Journal of Project Management **18**(2): 139-147.

Epley, N. and Gilovich, T. (1999). "Just going along: Nonconscious priming and conformity to social pressure." Journal of Experimental Social Psychology **35**: 578-589.

Gigerenzer, G. (1994). Why the distinction between single-event probabilities and frequencies is important for psychology (and vice versa). Subjective probabilities. G. Wright and P. Ayton. Chichester, Wiley.

Gigerenzer, G. and Todd, P. M. (1999). Simple heuristics that make us smart. New York, Oxford University Press.

Grether, D. M. and Plott, C. R. (1979). "Economic theory of choice and the preference reversal phenomenon." American Economic Review **69**: 623-638.

Hagafors, R. and Brehmer, B. (1983). "Does having to justify one's judgments change the nature of the judgment process?" Organizational Behaviour and Human Decision Processes **31**(2): 223-232.

Hall, E. (1998). Managing risk: methods for software system development, Addison-Wesley.

Hihn, J. and Habib-Agahi, H. (1991). Cost estimation of software intensive projects: A survey of current practices. International Conference on Software Engineering, IEEE Comput. Soc. Press, Los Alamitos, CA, USA: 276-287.

Hora, S. C., Hora, J. A., et al. (1992). "Assessment of probability distributions for continuous random variables: a comparison of the bisection and fixed value methods." Organizational Behavior and Human Decision Processes **51**: 133-155.

Huchzermeier, A. and Loch, C. H. (2001). "Project Management Under Risk: Using the Real Options Approach to Evaluate Flexibility in R&D." Management Science **47**(1): 85 - 101.

Humphrey, W. S. and Singpuwally, N. D. (1991). "Predicting (individual) software productivity." IEEE Transactions on Software Engineering **17**(2): 196-207.

Höst, M. and Wohlin, C. (1998). An experimental study of individual subjective effort estimations and combinations of the estimates. International Conference on Software Engineering, Kyoto, Japan, IEEE Comput. Soc, Los Alamitos, CA, USA: 332-339.

Institute, P. M. (2000). A guide to the project management body of knowledge, PMI Publishing Division.

Janis, I. L. (1972). Victims of groupthink. Boston, Houghton Mifflin.

Jørgensen, M. (2004a). "Regression Models of Software Development Effort Estimation Accuracy and Bias." To appear in: Journal of Empirical Software Engineering.

Jørgensen, M. (2004b). "A Review of Studies on Expert Estimation of Software Development Effort." To appear in: Journal of Systems and Software.

Jørgensen, M. (2004c). "Top-Down and Bottom-Up Expert Estimation of Software Development Effort." To appear in: Journal of Information and Software Technology.

Jørgensen, M. and Moløkken, K. (2002). Combination of software development effort prediction intervals: Why, when and how? Proceedings: Conference on Software Engineering and Knowledge Engineering, Italy.

Jørgensen, M. and Moløkken, K. (2004). Eliminating Over-Confidence in Software Development Effort Estimates. Submitted to: PROFES 2004, Japan.

Jørgensen, M. and Sjøberg, D. I. K. (2001a). "Impact of effort estimates on software project work." Information and Software Technology **43**(15): 939-948.

Jørgensen, M. and Sjøberg, D. I. K. (2001b). "Software process improvement and human judgement heuristics." Scandinavian Journal of Information Systems **13**: 99-121.

Jørgensen, M. and Sjøberg, D. I. K. (2003). "An effort prediction interval approach based on the empirical distribution of previous estimation accuracy." Journal of Information and Software Technology **45**(3): 123-136.

Jørgensen, M. and Teigen, K. H. (2002). Uncertainty Intervals versus Interval Uncertainty: An Alternative Method for Eliciting Effort Prediction Intervals in Software Development Projects. International Conference on Project Management (ProMAC), Singapore: 343-352.

Jørgensen, M., Teigen, K. H., et al. (2004). "Better Sure than Safe? Overconfidence in Judgment Based Software Development Effort Prediction Intervals." To appear in: Journal of System and Software.

Kahneman, D. and Lovallo, D. (1993). "Timid choices and bold forecasts: A cognitive perspective on risk taking." Management Science **39**(1): 17-31.

Kahneman, D. and Tversky, A. (1982). "Variants of uncertainty." Cognition **11**: 143-157.

Kahnemann, D., Slovic, P., and Tversky, A. (1982). Judgement under uncertainty: Heuristics and biases, Cambridge University Press.

Kernaghan, J. A. and Cooke, R. A. (1986). "The contribution of the group process to successful project planning in R&D settings." IEEE Transactions on Engineering Management **33**(3): 134-140.

Kitchenham, B. and Linkman, S. (1997). "Estimates, uncertainty, and risk." IEEE Software **14**(3): 69-74.

Kitchenham, B., Pfleeger, S. L., et al. (2002). "A case study of maintenance estimation accuracy." To appear in: Journal of Systems and Software.

Kitchenham, B., Pickard, L., et al. (2003). "Modelling software bidding risks." To appear in: IEEE Transactions on Software Engineering.

Kogan, N. and Wallach, M. A. (1964). Risk taking: a study in cognition and personality. New York, Holt, Rinehart and Winston.

Lederer, A. L. and Prasad, J. (2000). "Software management and cost estimating error." Journal of Systems and Software **50**(1): 33-42.

Lepper, M. R., Greene, D., et al. (1973). "Undermining children's intrinsic interest with extrinsic reward: A test of the "overjustification" hypothesis." Journal of Personality and Social Psychology **28**: 129-137.

Lerner, J. S. and Tetlock, P. E. (1999). "Accounting for the effects of accountability." Psychological bulletin **125**(2): 255-275.

Lichtenstein, S. and Fischhoff, B. (1977). "Do those who know more also know more about how much they know?" Organizational Behaviour and Human Decision Processes. **20**(2): 159-183.

Lindsay, R. M. and Ehrenberg, A. S. C. (1993). "The design of replicated studies." The American Statistician **47**(3): 217-228.

MacDonell, S. G. and Shepperd, J. M. (2003). "Combining techniques to optimize effort predictions in software project management." Journal of Systems and Software **66**: 91-98.

Maines, L. A. (1996). "An experimental examination of subjective forecast combination." International Journal of Forecasting **12**(2): 223-233.

McConnel, S. (1998). Software project survival guide, Microsoft Press.

Moder, J. J., Phillips, C. R., et al. (1995). Project management with CPM, PERT and precedence diagramming. Wisconsin, U.S.A, Blitz Publishing Company.

Moløkken, K. (2002). Expert estimation of Web-development effort: Individual biases and group processes (Masters Thesis). Department of Informatics, University of Oslo.

Moløkken, K. and Jørgensen, M. (2003). A review of surveys on software effort estimation. IEEE International Symposium on Empirical Software Engineering (ISESE 2003), Rome, Italy.

Moløkken, K. and Jørgensen, M. (2004). "Expert estimation of the effort of web-development projects: Why are software professionals in technical roles more optimistic than those in non-technical roles." To appear in Journal of Empirical Software Engineering.

Montgomery, D. C. (2000). Introduction to Statistical Quality Control. New York, Wiley.

Myrtveit, I. and Stensrud, E. (1999). "A controlled experiment to assess the benefits of estimating with analogy and regression models." IEEE Transactions on Software Engineering **25**(4): 510-525.

NASA (1990). Manager's handbook for software development. Goddard Space Flight Center, Greenbelt, MD, NASA Software Engineering Laboratory.

Newby-Clark, I. R., Ross, M., et al. (2000). "People focus on optimistic scenarios and disregard pessimistic scenarios when predicting task completion times." Journal of Experimental Psychology: Applied **6**(3): 171-182.

Oskamp, S. (1965). "Overconfidence in case-study judgments." Journal of Consulting Psychology **29**(3): 261 - 265.

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

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

Pich, M. T., Loch, C. H., et al. (2002). "On Uncertainty, Ambiguity and Complexity in Project Management." Management Science **48**: 1008-1023.

Reifer, D. J. (1990). "ASSET-R: A function point sizing tool for scientific and real-time systems." Journal of Systems and Software **11**(3): 159-172.

Ross, J. and Staw, B. M. (1986). "Expo 86: An escalation prototype." Administrative Science Quarterly **31**: 274-297.

Rowe, G. and Wright, G. (2001). Expert opinions in forecasting: The role of the Delphi process. Principles of forecasting: A handbook for researchers and practitioners. J. S. Armstrong. Boston, Kluwer Academic Publishers**:** 125-144.

Schwenk, C. R. and Cosier, R. (1980). "Effects of the expert, devil's advocate, and dialectical inquiry methods on prediction performance." Organizational Behaviour and Human Decision Processes **26**(3): 409-424.

Seaver, D. A., Winterfeldt von, D., et al. (1978). "Eliciting subjective probability distributions on continuous variables." Organizational Behaviour and Human Decision Processes. **21**(3): 379-391.

Sedlmeier, P. (1999). Improving statistical reasoning: Theoretical models and practical implications. Mahwah, NJ, Erlbaum.

Sengupta, K. and Abdel-Hamid, T. K. (1996). "The impact of unreliable information on the management of software projects: A dynamic decision perspective." IEEE Transactions on Systems, man, and cybernetics **26**(2): 177-189.

Sieber, J. E. (1974). "Effects of decision importance on ability to generate warranted subjective uncertainty." Journal of Personality and Social Psychology **30**(5): 688-694.

Siegel-Jacobs, K. and Yates, J. F. (1996). "Effects of procedural and outcome accountability on judgment quality." Organizational Behavior and Human Decision Processes. **65**: 1-17.

Sloman, S. A., Over, D., et al. (2003). "Frequency illusions and other fallacies." Organizational Behavior and Human Decision Processes. **91**: 296-309.

Sommerville, I. (2001). Software Engineering. Harlow, England, Pearson Education Limited.

Stamelos, I. and Angelis, L. (2001). "Managing uncertainty in project portfolio cost estimation." Information and Software Technology **43**(13): 759-768.

Stanovich, K. E. (1991). Who is rational? Studies of individual differences in reasoning. Mahwah, New Jersey, Lawrence Erlbaum Associates.

Taff, L. M., Borchering, J. W., et al. (1991). "Estimeetings: development estimates and a front-end process for a large project." IEEE Transactions on Software Engineering **17**(8): 839-849.

Taylor, J. W. and Bunn, D. W. (1999). "Investigating Improvements in the Accuracy of Prediction Intervals for Combinations of Forecasts: A Simulation Study." International Journal of Forecasting **15**(3): 325-339.

Tversky, A. and Kahneman, D. (1974). "Judgment under uncertainty: Heuristics and biases." Science **185**: 1124-1130.

van Genuchten, M. (1991). "Why is software late? An empirical study of reasons for delay in software development." IEEE Transactions on Software Engineering **17**(6): 582-590.

Whitecotton, S. M., Sanders, D. E., et al. (1998). "Improving predictive accuracy with a combination of human intuition and mechanical decision aids." Organizational Behaviour and Human Decision Processes **76**(3): 325-348.

Wilson, T. D. and Brekke, N. (1994). "Mental contamination and mental correction: unwanted influences on judgments and evaluation." Psychological Bulletin **116**(1): 117-142.

Yaniv, I. and Foster, D. P. (1997). "Precision and accuracy of judgmental estimation." Journal of Behavioral Decision Making **10**: 21-32.