CHAPTER 3

Magne Jørgensen

Overconfidence in the Accuracy of Work Effort Predictions: The Role of Interval Width

Effort Prediction Intervals

Assume that you are supposed to complete the task of copying a book on your workplace's copying machine. You have never performed exactly this task before but have some experience with similar copy machine tasks. Part of your previous experience is the observation that the effort you need to complete copying tasks is not very predictable. Sometimes seemingly small and easy copying tasks require much more effort than predicted, e.g., due to errors you make, paper jams, and people disturbing you. Clearly, it is difficult to provide accurate predictions of the effort you would need to complete the task in this context. This difficulty may lead to the thought of presenting the predicted effort usage as an effort interval. You could for example predict that the copying task requires between one and two work-hours. This effort interval information may be valuable as input to plans on when you will, with high certainty, be available for subsequent tasks and when others can use the copying machine. But, what is the meaning of this effort interval information? It certainly does not mean that it is impossible to use less than one or more than two work-hours. Even if you extended the maximum effort value to 100 work-hours, it could happen that you would need even more effort. To make the statement more precise, you may therefore be asked to provide the minimum and maximum effort values that make it, for example, 98% likely to include the actual use of effort. Here, the probability of 98% denotes your confidence level, and the minimum-maximum effort interval with the corresponding confidence level indicates your effort prediction interval. In frequency terms, if you base your assessments on 98% effort prediction intervals, you should observe that, on the long run, 98 out of 100 actual effort values are inside your minimum-maximum effort intervals. The problem of uncertainty assessments is, however, far from solved: How do you know when your minimum-maximum effort values reflect a confidence level of 98%? Do you possess any proper, conscious or unconscious, strategy to translate your previous effort prediction experience and knowledge about the current task into this kind of effort prediction interval?

Previous studies suggest that people are not good at these kinds of uncertainty assessments and that in general they provide too narrow effort intervals for the requested confidence levels, which are typically in the range of 90–99%. In software development, which is the domain where I collected the empirical evidence for this article, the first study documenting the strong tendency towards overconfidence was conducted by Connolly and Dean (1997). They found that the actual effort of students completing a small software programming assignment fell outside the students' 98% confidence effort prediction interval in 43% of the cases (Study 1, Assignment 1). In Jørgensen, Teigen, and Moløkken (2004) (Study A) it is documented that overconfidence can also be a problem in field settings with software professionals. The 90% confidence effort prediction intervals, provided for software development activities, included the actual use of effort in only 35% of the cases. Real-life conditions and more experience do not necessarily remove the tendency towards too narrow confidence intervals. Similar results are reported by, for example, McKenzie et al. (2008).

To the best of my knowledge, the most common format for eliciting the effort prediction intervals is: *Please provide the minimum and maximum effort so that you are X% confident to include the actual effort*. This elicitation format, typically with confidence level (X%) of 90% or 98%, is recommended in typical project management textbooks and part of the frequently used PERT method for project planning (Moder, Phillips, & Davis, 1995). This elicitation format may have been introduced to facilitate the mathematical calculations and does not seem to be based on results on how different elicitation formats support human judgment. Hardly any textbook on project management refers to the in psychology well-known tendency towards overconfidence when using this elicitation format. The results reported in Jørgensen et al. (2004) (Study C) provide one example of a problematic side of the traditional elicitation format. We randomly divided software developers (computer science students) into four groups and, dependent on the group, instructed them to be 50%, 70%, 90% or 99% confident to include the actual effort in their minimum-maximum effort intervals. We found that the average widths of the intervals were not significantly different in the different groups! It was the somewhat undefined concept of minimum and maximum that seemed to dominate their assessments, not the instructed confidence levels for which they seemed to have no strategy to integrate properly. In a recent study (unpublished), we replicated this study with software professionals and found essentially the same result. A possible implication of these findings is that people may be perceived as overconfident, well calibrated or underconfident depending on whether the requested interval probabilities represent high, medium or low confidence, respectively. The main reason for repeatedly observing

overconfidence in numerous studies in various domains may consequently be that people tend to ignore probabilities together with a request for high confidence intervals, not that they have a disposition towards overconfidence. In other words, people are ignorant of probabilities, rather than overconfident. Without a proper strategy enabling the transfer of previous experience into probability based prediction intervals, factors such as the vague and context-dependent semantic of "minimum" and "maximum", the desire to provide informative prediction intervals (Yaniv & Foster, 1997) and the wish to be perceived as competent (Jørgensen et al., 2004) may dominate the uncertainty assessments.

An Alternative Elicitation Format

The inability to reflect variation in confidence levels when using the traditional elicitation format was one important reason for Karl Halvor Teigen and myself (Jørgensen & Teigen, 2002) to propose and evaluate an alternative elicitation format. The underlying thought was that we should change the elicitation format so that it would be easier to translate previous prediction experience into uncertainty assessments and as a consequence enable more realistic uncertainty assessment. We assumed a context with experience from completion of a set of related tasks with corresponding feedback on prediction accuracy that enabled people to know the approximate distribution of the prediction error. In several relevant contexts I have confirmed that software developers are indeed able to recall such distributions, e.g., to recall how often they previously had overrun the predicted effort by more than 50%. We further assumed that this prediction error distribution would be meaningful as input to the uncertainty assessment of the task to be predicted in most situations. If, for example, 20% of the previously completed, similar tasks had been overrun by more than 50%, it would be reasonable to think that the likelihood of spending 50% more effort than the predicted effort would be about 20%. These assumptions, we argued, indicate that it would be beneficial for requests to be framed as follows: *How likely is it that the actual effort will be X% lower (and/or Y% higher) than the predicted most likely effort?*

This alternative elicitation format was evaluated both experimentally and in the field, and found to yield more realistic uncertainty assessments. There were contexts with long sequences of tasks where the alternative format led to near perfectly calibrated uncertainty assessments (Jørgensen & Teigen, 2002) and contexts where it improved the assessments substantially (Jørgensen et al., 2004), but our studies also revealed individual and team differences in abilities to benefit from it (Jørgensen & Moløkken, 2004). Similar positive

effects of the alternative format for almanac type of questions is reported by, for example, Winman et al. (2004).

A limitation related to the generality of the previous evaluations of the alternative elicitation format in the effort prediction interval context was the use of a pretty wide effort interval, typically an interval that represented a minimum of 50% and a maximum of 200% of the predicted effort. None of the studies examined to what extent the alternative elicitation format would give similar benefits in situations with more narrow intervals. More narrow intervals are usually the more relevant ones, both in the context of project planning, budgeting and bidding. A project manager may, for example, benefit from knowing how likely it is to exceed his budget when it is based on a work-effort 25% higher than the predicted most likely effort usage of a project. This does not mean that very wide effort intervals are uninteresting. They may, for example, be useful to assess how likely it is that a project is strongly under-estimated, which in turn may be useful in the risk management of the project.

To examine the alternative elicitation format in the context of more narrow intervals, two studies with varying interval widths were completed. In the ideal case, where the uncertainty assessments are based purely on the error distribution of similar predictions, we would find that the degree of realism would not be influenced by narrower intervals. If, however, there were elements biasing the uncertainty assessments, we would expect to observe the more overconfidence the narrower the minimum-maximum effort intervals. An increase in overconfidence with narrower intervals is consistent with our finding that high confidence is used to indicate and assess software development skill. In Jørgensen et al. (2004) (Study D) we created a scenario where the software managers were asked to evaluate the skills of two software developers, based only on information about the prediction accuracy and effort uncertainty assessment realism. We found that the managers tended to assess the overconfident developer as the one with the most knowledge about the tasks completed, even when informed that both developers had the same prediction accuracy and that the least confident developer had the most realistic uncertainty assessments. An increase in overconfidence with narrower interval width is also consistent with, in many ways, a rational effort uncertainty assessment strategy where the probabilities regress towards a central probability value. The less one knows about the real uncertainty, the more the uncertainty assessment may regress towards this central value. This strategy would contribute to excessive probability values on narrow, and too low values on wide effort intervals. Finally, there may also be an effect from a preference towards providing uncertainty assessments perceived as informative (Yaniv & Foster, 1997). As an illustration, the statements that it is

60% and that it is 90% likely not to exceed the predicted effort by more than 25% are both of value for a project manager. The two statements may nevertheless differ in how useful they are perceived to be. It is, for example, easier to create a project plan based on a high confidence input about the effort usage rather than medium to low confidence input. When uncertain about the uncertainty, other concerns than realism may dominate the assessments.

The Effect of Decreasing Effort Interval Width

The following two studies evaluate the effect of decreasing interval width on the level of overconfidence. In none of the studies were the participants trained in the use of the effort prediction error distribution in the assessment of effort uncertainty or instructed to use this kind of information. This means that the results may be different with better-trained participants. If the software professionals used previous effort prediction error as the main input to the effort uncertainty assessments, we would, as stated earlier, expect to see no increase in overconfidence with decreased interval width.

The first study evaluates the effect in a context where software professionals predict the effort and assess the effort uncertainty of own work. The second study relied on the effort predictions and prediction error feedback resulting from the first study, but lets developers not involved in the software development assess the uncertainty of the effort predictions. Predictions of outcome and uncertainty of own work are likely to be more exposed to biases, e.g., motivational biases due to a wish to "look competent," than predictions of others' work. During the first study, participants gained much knowledge about how to complete the task, while the participants in the second study had only shallow knowledge about the task completion and had to rely on other information to assess the effort prediction uncertainty, such as type of task and previous effort prediction accuracy of similar tasks. The situation in the first study may therefore activate more "inside" and the second study more "outside" views on the task (Kahneman & Lovallo, 1993). Outside views may be more oriented towards "looking back" (history-based) and, consequently, less likely to be exposed to human biases than the "looking forward," planning-oriented predictions of own work based on detailed decomposition of the work.

ASSESSING THE ACCURACY OF OWN PREDICTIONS OF OWN WORK (STUDY 1)

The 20 participants in the first study were software professionals with extensive experience in predicting the effort of and completing software development tasks. The software professionals were paid close to ordinary fees for their work and asked to treat the

development work as ordinary consultancy work. The tasks were performed on a real-life system and the developers were not aware that other developers completed the same tasks. The total effort on all five tasks per developer was on average 38 work-hours.

All software professionals were instructed to predict the effort they would use, provide an assessment of the uncertainty of the effort prediction, applying the alternative elicitation format; and complete the same five tasks in the same sequence. Other results from the same study have previously been reported in Jørgensen and Gruschke (2009), with a focus on the effect of the so-called lessons learned sessions to improve effort predictions and uncertainty assessments. The study found no or little observed effect from lessons learned sessions (neither on prediction accuracy nor effort uncertainty assessment). For the purpose of the analysis in this section, I therefore combine the data from the ten developers exposed to lessons learned sessions and from the ten developers receiving outcome feedback only.

A developer started his (there were only male participants) work by reading the specification and analysing the first task. Then, he predicted the effort he most likely would need to complete the task and provided his confidence (a probability between 0 and 100%) in including the actual effort in the intervals E1 = [90%; 110%], E2 = [60%; 150%] and E3 = [50%; 200%] of the predicted effort. The spreadsheet used for the uncertainty assessment presented the intervals both as percentages and as the effort value it represented in work-hours. E1 represents a narrow, E2 a medium and E3 a wide effort interval. If, for example, the developer predicted the effort of the first task to be 10 work-hours, he would consequently be asked to assess how likely it was the actual effort would be between 9 [90%] and 11 [110%] work-hours (E1), between 6 [60%] and 15 [150%] work-hours (E2), and between 5 [50%] and 20 [200%] work-hours (E3). Subsequently, he completed the first task, i.e., he designed, programmed, tested and documented the tasks. In the role of the client, we checked the task solution for correctness, i.e., that the developed software fulfilled the requirements specified in the task description, and if correct, the developer was given feedback on his prediction accuracy and uncertainty assessment realism. The feedback could, for example, be that the actual effort was overrun by 60%, and that the effort prediction was inside the uncertainty interval E3, but outside the intervals E1 and E2. If the task solution was incorrect, the developer had to correct errors until accepted by the client. The same steps were repeated for all five tasks.

The effort predictions were on average (median) 13% too low, which reflects a tendency towards over-optimism. The medium absolute prediction error was 46%, which reflect that the work efforts of the tasks were difficult to predict accurately. Table 1 shows the

mean level of confidence and the "hit rate", i.e., the proportion of effort predictions included in the effort interval, for all three effort uncertainty intervals. As can be seen, the deviation between confidence level and hit rate increases with decreasing width of the effort intervals. This tendency exists even when we only look at the two last tasks, i.e., the tasks where the developers had more opportunities to adjust their confidence levels based on highly relevant and recent effort prediction experience.

The mean confidence levels are about the same for the first three and last two tasks. This suggests that the reduction in overconfidence in the two last tasks is mainly due to more accurate predictions of most likely use of effort, not to learning from previous effort prediction error feedback. This also indicates that the increase in uncertainty assessment realism with wider prediction intervals to some extent is a result of ignorance of probabilities (confidence levels), rather than a tendency towards overconfidence in the accuracy of own predictions.

	Tasks	**E1 = [90%; 110%]**	**E2 = [60%; 150%]**	**E3 = [50%; 200%]**
Mean confidence	Task 1 – Task 5	56%	81%	94%
Mean hit rate	Task 1 – Task 5	18%	47%	70%
Mean confidence	Task 1 – Task 3	55%	81%	95%
Mean hit rate	Task 1 – Task 3	10%	42%	65%
Mean confidence	Task 4 – Task 5	57%	81%	92%
Mean hit rate	Task 4 – Task 5	30%	55%	78%

Table 1. Correspondence between Confidence Level and Hit Rate

In previous studies we found that most (Jørgensen & Teigen, 2002) or at least some (Jørgensen & Moløkken, 2004) of the software developers, when exposed to repeated tasks, started using a strategy that makes use of previous prediction accuracy experience in their uncertainty assessment when applying the alternative elicitation format. An examination of the individual developers' predictions indicates that this is the case here, too. The low number

of tasks, however, makes the analysis of individual developers not very robust. The main finding is consequently that we did not replicate the previous promising results with the alternative format, especially not for the narrower effort intervals. In previous studies we either had longer sequences of predictions or we explicitly made the participants develop a distribution of previous estimation error to remind them of this distribution's relevance. This may explain some of the difference in results.

ASSESSING THE ACCURACY OF OTHER DEVELOPERS' EFFORT PREDICTIONS (STUDY 2)

A second study, in the context of the alternative elicitation format, was designed to test the effect of interval width in a situation with less reason for overconfidence as a means of being perceived as skilled or providing information value to the receiver, i.e., when assessing the uncertainty of other participants' predictions. In this study, 83 software professionals were asked to assess the estimation accuracy of the effort prediction made by the 20 software developers in the first study. Each of the software professionals was randomly allocated to the effort predictions and feedback related to 1 of the 20 original developers. The software professionals had only parts of the information possessed by the developers who originally predicted and completed the tasks. They shared the task descriptions and information about the previous prediction accuracy, but the participants in the second study did not have as much time for analysis, and much less knowledge about the software development context. The uncertainty assessments were only completed for Tasks 4 and 5 (the two last tasks), since they otherwise would lack relevant historical data. The same effort intervals (E1, E2 and E3) were used. Table 2 shows the mean confidence levels together with the mean hit rate of the effort intervals.

Effort intervals	**E1 = [90%;110%]**	**E2 = [60%; 150%]**	**E3 = [50%; 200%]**
Mean confidence	28%	56%	81%
Mean hit rate	30%	55%	78%

Table 2. Correspondence between Confidence and Hit Rate (Task 4 – Task 5)

The software professionals were now, on average, very well calibrated. This suggests that the software professionals managed to use the previous prediction error and adjust realistically for learning effects in this context.

Conclusions

People seem to find it difficult to translate previous effort prediction experience into minimum and maximum effort values representing, for example, 98% confidence. Typically, studies find hit rates lower than those reflecting the requested confidence levels. This is frequently used as an indication that people are overconfident in the accuracy of their own predictions. This section refers to evidence suggesting that a description of this as overconfidence may be misleading and that probability ignorance is a more likely underlying cause. People typically possess no proper strategy to assess the probability of including the actual value in their intervals, and unsurprisingly ignore the confidence level and emphasize other factors, such as a wish to be perceived as skilled and informative. In order to avoid probability ignorance, an alternative elicitation format based on assessing the probability of including the actual effort in a given interval is proposed. Previous studies suggest that this format has the potential of enabling a translating previous prediction experience into confidence levels (probabilities) and, as a consequence, more realistic effort uncertainty assessments.

The studies in this article show however that the alternative elicitation format does not always remove the overconfidence. When decreasing the interval width in situations with assessment of the uncertainty of own effort predictions of own work, the degree of overconfidence increased substantially. Some of the positive effect of the previous studies applying the alternative elicitation format may consequently have been a result of the quite wide effort intervals used and not necessarily a better facilitation of the use of previous prediction experience. On the other hand, the second study showed an, on average, near perfect calibration when assessing the uncertainty of other people's effort predictions and using the alternative elicitation format. Achieving realistic assessments of effort uncertainty seems consequently to require both proper elicitation formats and removal of factors that lead to overconfidence. In particular, the use of other people than those involved in the prediction of effort and/or completion of the work to assess the uncertainty of the effort predictions may be a key factor to benefiting from the alternative elicitation format.

References

Connolly, T., & Dean, D. (1997). Decomposed versus holistic estimates of effort required for software writing tasks. *Management Science. 43,* 1029–1045.

Jørgensen, M., & Gruschke, T. M. (2009). The impact of lessons-learned sessions on effort estimation and uncertainty assessments. *IEEE Transactions on Software Engineering*, *35*, 368–383.

Jørgensen, M., & Moløkken, K. (2004). *Eliminating over-confidence in software development effort estimates*. Conference on Product Focused Software Process Improvement, 174-184, Japan, Springer-verlag, Lecture Notes on Computer Science.

Jørgensen, M., & 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), 343-352, Singapore.

Jørgensen, M., Teigen, K. H., & Moløkken, K. (2004). Better sure than safe? Over-confidence in judgment based software development effort prediction intervals. *Journal of Systems and Software, 70*, 79–93.

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

McKenzie, C. R. M., Liersch, M., & Yaniv, I. (2008). Overconfidence in interval estimates: What does expertise buy you? *Organizational Behavior and Human Decision Processes, 107*, 179–191.

Moder, J. J., Phillips, C. R., & Davis, E. W. (1995). *Project management with CPM, PERT and precedence diagramming*. Middleton, WI: Blitz Publishing Company.

Winman, A., Hanson, P., & Jusling, P. (2004). Subjective probability intervals: how to reduce overconfidence by interval evaluation. *Journal of Experimental Psychology: Learning, Memory and Cognition, 30*, 1167–1175.

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