Combination of Software Development Effort Prediction Intervals: Why, When and How?

Magne J□rgensen
Simula Research Laboratory
P.O. Box 134, 1325 Lysaker, Norway
magne.jorgensen@simula.no

Kjetil Mol□kken
University of Oslo, Department of Informatics
P.O.Box 1080, 0316 Oslo, Norway
kjetilmo@ifi.uio.no

ABSTRACT

The uncertainty of a software development effort estimate may be described through a prediction interval, e.g., that the most likely use of effort is 1.500 work-hours and that it is 90 % probable (90% confidence level) that the actual use of effort will be between 1.000 (minimum) and 2.000 (maximum) work-hours. Previous studies suggest that software development effort prediction intervals are, on average, much too narrow to reflect high confidence levels, i.e., the uncertainty is under-estimated. This paper analyses when and how a combination of several individual prediction intervals of the same task improves the correspondence between hit rate and confidence level of effort prediction intervals. We analyse three combination strategies: (1) Average of the individual minimum and maximum values, (2) Maximum and minimum of the individual maximum and minimum values, and (3) Group process (discussion) based prediction intervals. Based on an empirical study with software professionals we found that strategy (1) did not lead to much correspondence improvement compared with the individual prediction intervals, mainly because of a, as expected, strong individual bias towards too narrow prediction intervals. Strategy (2) and (3) both improved the correspondence. However, Strategy (3) used the uncertainty information more efficiently, i.e., had narrower prediction intervals for the same hit rate. Our empirical results suggest that group discussion based combination of prediction intervals should be used instead of ìmechanicalî combinations of individual prediction intervals. Clearly, there is no best combination strategy for all prediction interval situations, and the choice of strategy should be based on an investigation of factors that impact the usefulness of a strategy.

Categories and Subject Descriptors

D.2.9 [Management]: Cost Estimation

General Terms

Management, Measurement, Experimentation, Human Factors

Keywords

Uncertainty of effort predictions, Combination, Group Estimates

1. INTRODUCTION[1]

Predictions of the required effort to complete a project cannot be expected to be accurate when the requirements are incompletely described, the development environment is unstable, the project members lack experience from similar projects, or when key project members leave in the middle of the project. These conditions are, however, present in many software projects. Consequently, knowledge about the uncertainty of how much effort will be required to complete a project is important. For example, a wider span of possible use of effort should lead to a larger part of the project budget being allocated to the resolution of unexpected events.

The use of effort prediction intervals is one way of describing the uncertainty. For example, a project manager may describe the uncertainty as ìthere is a 90% (or, very high) probability that the actual use of effort of my project will be in the interval 1000 ñ 2000 work-hoursî. Effort prediction intervals are frequently used in software projects, and, as far as we have observed, the difference between the estimated minimum (best case) effort and the estimated maximum (worst case) effort can be very large.

A problem with expert judgment based effort prediction intervals is that they are typically much too narrow to reflect the actual uncertainty. For example, we analysed 18 software development projects and found that only 64% of the activity effort prediction intervals included the actual effort, although the intervals were based on a 90% confidence, i.e., a hit rate of 64% instead of the desired 90%. Similarly, Conolly and Dean [1] report that the actual effort used to solve programming tasks fell in only 60% of the tasks inside the 98% confidence effort prediction intervals. Explicit attention to and training in establishing good minimum and maximum effort values did increase the proportion inside the PI to about 70%, which was still far from the required 98% confidence. Several studies on human judgement report similar results on prediction intervals in other domains, see Arkes [2] for a recent overview and discussion.

The use of effort prediction interval models, e.g., the models described in [3, 4], may lead to better correspondence between confidence level and hit rate. However, a comparison of model and expert judgement-based prediction intervals carried out by one of the authors [5], indicates that the formal models are prone to use the available uncertainty information less efficiently. This

[1] This paper was reduced from 8 pages. One state-of-practice and one other empirical study were included in the full paper. If you want the full paper, send a mail to magne.jorgensen@simula.

means that model-based prediction intervals may be less useful than those based on expert judgement when planning software projects and that it may be difficult to replace judgement-based effort uncertainty intervals.

Combining prediction intervals from several software professionals may improve the correspondence between hit rate and confidence level. Surprisingly, we have not been able to find any software development study on this topic. Most research on effort predictions seems to focus on establishing and comparing formal models providing single point effort predictions, e.g., [6, 7].

The benefits of combining predictions from different sources are well documented. For example, Armstrong [8] reports, based on 30 empirical studies, that predictions based on the average of individual predictions were on average 12.5% more accurate than a randomly selected individual prediction. Similarly, Hˆst and Wohlin [9] report good results from averaging individual software development effort predictions in student projects. Taff et al. [10] report benefits from software development effort predictions based on group consensus. In this paper we are interested the combination of uncertainty predictions, i.e., prediction intervals, which may reveal different combination strengths and weaknesses compared with combination of estimates of most likely values.

The remaining part of this paper is organized as follows: Section 2 describes an empirical study comparing different prediction interval combination approaches. Section 3 discusses the results.

2. AN EMPIRICAL STUDY

2.1 Measures

The common performance measure of software development effort prediction intervals is the ìhit rateî [1, 3], i.e., the proportion of the prediction intervals that include the actual effort. There should be a close correspondence between the hit rate of a number of prediction intervals and the chosen confidence level, i.e., if the prediction intervals are based on the confidence level *K* %, then we expect a hit rate of *K* % (*K* out of 100 actual values inside the minimum - maximum interval). Murphy and Winkler [11] call this correspondence ësecondary validityí, while the ëprimary validityí is defined as the correspondence between an individual prediction interval and the underlying probability distribution. In software effort predictions we do not have access to the underlying effort probability distribution of individual development tasks and must, for this reason rely on ësecondary validityí measures based on a, preferably high, number of evaluated prediction intervals. We discuss the strengths and weaknesses of use of ësecondary validityí prediction interval measures in [12].

To reduce some of the weaknesses of the hit rate measure, e.g., that there are sub-optimal strategies to achieve a high hit rate [13], we are interested in the relative width of the effort prediction interval (PIWidth). We define the interval width as $PIWidth_i = (Max_i - Min_i) / Pred_i$, where $Pred_i$ is the predicted effort of task *i*, and Max_i and Min_i are the predicted maximum and minimum effort of task *i*. When two combination strategies achieve a similar hit rate for the same confidence level, then a lower average relative width of the prediction intervals indicates a better use of uncertainty information [14]. Using PIWidth we are therefore able to exclude strategies that provide unnecessary wide effort prediction intervals. It is, however, important to accept that the goal is to provide effort prediction intervals that reflect the underlying uncertainty of the project work, not, for example, prediction intervals that ìeaseî the project leaderís decisions. When the uncertainty is high, the effort prediction intervals should be wide.

2.2 The Study

2.2.1 Design

20 software professionals from the same, medium large, web-development company were paid to participate in the study. The participants were divided into five estimation teams. The roles covered by each prediction team were: ìEngagement managerî (customer relations and contract-responsible manager), project manager, software developer and user interaction designer. As input to the effort prediction process the participants got the specification of a, at that point of time, on-going software development project conducted by the organization. The project was, of course, not known by the participants in our study. The prediction process of the experiment was similar to that typically used by the organization and the participants were told to act as if this was a real effort prediction task.

The actual effort of the specified project turned out to be about 2.400 work-hours. This actual effort value should, however, be carefully interpreted. A software project has several possible outcomes and a repeated completion of the same project, even if we assume no learning, would lead to a distribution of actual effort values. For example, the actual effort of the specified project was impacted by unexpected events that a new project probably would not meet. On the other hand, a new project may meet other unexpected events. The analysis of whether the actual effort of the project is within the effort prediction interval is nevertheless meaningful. As far as we understood from interviews with the project leader, the project events were not exceptional. Consequently, a 90% confidence effort prediction interval should include the unexpected events of the project. The effort originally estimated by the project was 1.240 work-hours, i.e., the original estimate was much too low.

The prediction process was divided into two parts. In Part 1 each participant predicted the most likely effort and a 90% confidence effort prediction interval without discussing with the other members of the prediction team, i.e., Part 1 resulted in the individual effort predictions and prediction intervals. Then, in Part 2, the prediction teams discussed their individual predictions and agreed on the teamís predictions (the *CombTeam* predictions). The team process was similar to, although less formal than, the ìEstimeetingî process described in [10].

We applied three combination strategies:

- *CombAverage*: The combined minimum is the average of all the minimum values of the participants in a team, and the combined maximum is the average of all the maximum values. The RWidth of a teamís prediction interval is: [average(Max)-average(Min)]/average(ML).
- *CombWidest*: The combined minimum is the minimum of all the minimum values of the participants in a team, and the combined maximum is the maximum of the maximum values. The RWidth of a teamís prediction interval is: [max(Max)-min(Min)]/average(ML).

Table 1. The effort predictions and prediction intervals

ROLE	Engagement Manager			Project Manager			User Interaction Designer			Software Developer			Team		
	Min	ML	Max	Min	ML	Max	Min	ML	Max	Min	ML	Max	Min	ML	Max
Team A	660	1200	1740	770	960	1150	1000	1500	2000	900	960	1200	900	1100	1500
Team B	1000	1550	4000	1400	1820	2200	1010	1140	1592	400	585	700	1200	1500	2500
Team C	1400	1850	2300	268	300	332	910	1260	1630	110	220	400	1300	1550	1900
Team D	470	547	630	850	914	1100	580	620	840	440	660	920	1205	1339	1473
Team E	1143	2286	3429	802	984	1181	1000	1500	2000	720	900	1080	1126	2251	3377

- *CombTeam*: The combined minimum and maximum are the minimum and maximum values the estimation team agreed on. The RWidth of a teamís prediction interval is: (Max ñ Min) / ML, based on the **Team** values (see Table 1).

2.2.2 Results

Table 1 displays the effort predictions and prediction intervals of all individuals and teams. (ML = Most likely effort, Min = Minimum effort, Max = Maximum effort, RWidth= Relative Width of a prediction interval). Table 1 shows that only 2 out of 20 participants (5% hit rate) had the actual effort (2.400 work-hours) inside their prediction intervals.

Table 2. The prediction intervals of the combination strategies

	CombAverage				CombWidest				CombTeam			
Team	Min	Max	RWidth	Hit	Min	Max	RWidth	Hit	Min	Max	RWidth	Hit
A	833	1523	0,60	0	660	2000	1,16	0	900	1500	0,55	0
B	953	2123	0,92	0	400	4000	2,83	1	1200	2500	0,87	1
C	672	1166	0,54	0	110	2300	2,41	0	1300	1900	0,39	0
D	585	873	0,42	0	440	1100	0,96	0	1205	1473	0,2	0
E	916	1923	0,71	0	720	3429	1,91	1	1126	3377	1,0	1

Table 2 shows the combined prediction intervals. We see that the strategies *CombWidest* and *CombTeam* had the same hit rate (2 hits out of 5, i.e., 40%) and that *CombAverage* hat a hit rate of 0%. Consequently, only *CombWidest* and *CombTeam* improved the correspondence between confidence level (90%) and hit rate compared with the individual prediction intervals (40% vs 5% hit rate). All hit rates were much too low to reflect a 90% confidence. More observations are, however, needed to analyse the exact size of the over-confidence among the participants in our study. The *CombTeam* strategy achieved its hit rate with narrower prediction intervals compared with *CombWidest,* i.e., *CombTeam* seems to use the uncertainty information more efficiently or include new uncertainty information from the group discussions.

Other interesting observations derivable from Table 1 and 2 include:

- The strong bias towards too low maximum values of the individual prediction intervals (the actual effort was 2.400 work-hours) was the main reason for the low performance of the *CombAverage* strategy.
- Typically, the groups (*CombTeam*) agreed on a minimum effort close to the maximum of the individual minimum effort predictions and a maximum between the average and the maximum, of the maximum values, i.e., the groups were less optimistic than the average individual, as a result of information sharing and group dynamics effects.
- Although the software developers and the project managers, presumably, knew more about *how* to develop the specified software than the engagement mangers and the user interaction designers, they had the least realistic prediction intervals, i.e., detailed, technical knowledge is not necessarily a good indicator of prediction skills. It is likely that the other participants in the team knew about the over-confident behaviour of those two roles. Otherwise there should have been a stronger weighting on the software developersí prediction intervals. This finding is in accordance with the results described by Maines [15], where team-based combination of predictions were strongly impacted by the belief that the analystsí predictions typically were optimistic.

3. DISCUSSION

This paper describes, to our knowledge, the first empirical results on the benefits of combined software development effort prediction intervals. Clearly, one should not use our results as proof of the superiority of one combination strategy, e.g., the *CombTeam* strategy, compared with another. Instead, the results should be used to increase the awareness of the factors that have an impact of the expected benefits of a combination strategy, e.g., the existence of a systematic bias among people with the same background. It may frequently be better to have estimation teams including several different roles, than estimation team including only technical roles, since the benefit from combining relates to the removal of systematic bias. The systematic higher bias towards over-optimism we found among the software developers and project leaders compared with the engagement managers and user interaction designers supports this suggestion.

We plan to conduct new studies where we manipulate the group prediction process regarding bias, previous prediction accuracy, background, and inter-correlation between prediction sources. In particular, we will evaluate the performance of the strategies actually used by software professionals compared with alternative strategies.

References

1. Connolly, T. and D. Dean, Decomposed versus holistic estimates of effort required for software writing tasks. *Management Science*, 1997. **43**(7): p. 1029-1045.
2. Arkes, H.R., *Overconfidence in judgmental forecasting*, in *Principles of forecasting: A handbook for researchers and practitioners*, J.S. Armstrong, Editor. 2001, Kluwer Academic Publishers. p. 495-515.
3. Angelis, L. and I. Stamelos, A simulation tool for efficient analogy based cost estimation. *Empirical software engineering*, 2000. **5**: p. 35-68.
4. J□rgensen, M. and D.I.K. Sj□berg. *A simple effort prediction interval method*. Proceedings of *Achieving Quality in Information Systems (AQuIS)*. 2002. Venize.
5. J□rgensen, M. and D.I.K. Sj□berg, An Effort Prediction Interval Approach Based on the Empirical Distribution of Previous Estimation Accuracy. *Submitted to Journal of Information Software and Technology*, 2002.
6. Briand, L.C., K. El Emam, and F. Bomarius. *COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment*. Proceedings of *Int. Conf. on Software Engineering*. 1998. Kyoto, Japan. p. 390-399.
7. J□rgensen, M., Experience with the accuracy of software maintenance task effort prediction models. *IEEE Transactions on Software Engineering*, 1995. **21**: p. 674-681.
8. Armstrong, J.S., *Combining Forecasts*, in *Principles of forecasting: A handbook for researchers and practitioners*, J.S. Armstrong, Editor. 2001, Kluwer Academic Publishers. p. 417-439.
9. Hˆ st, M. and C. Wohlin. *An experimental study of individual subjective effort estimations and combinations of the estimates*. Proceedings of *International Conference on Software Engineering*. 1998. Kyoto, Japan. p. 322-339
10. Taff, L.M., J.W. Borchering, and J.W.R. Hudgins, Estimeetings: Development estimates and a front-end process for a large project. *IEEE Transactions on Software Engineering*, 1991. **17**(8): p. 839-849.
11. Murphy, A.H. and R.L. Winkler, Scoring rules in probability assessment and evaluation. *Acta Psychologica*, 1970. **34**: p. 273-286.
12. J□rgensen, M. and K.H. Teigen. *Uncertainty Intervals versus Interval Uncertainty: An Alternative Method for Eliciting Effort Prediction Intervals in Software Development Projects*. Proceedings of *International Conference on Project Management (ProMac) (Accepted for publication)*. 2002. Singapore.
13. J□rgensen, M., K.H. Teigen, and K. Mol□kken, Better sure than safe? Overconfidence in judgment based software development effort prediction intervals. *Submitted to Journal of Systems and Software*, 2002.
14. Yaniv, I. and D.P. Foster, Precision and accuracy of judgmental estimation. *Journal of behavioral decision making*, 1997. **10**: p. 21-32.
15. Maines, L.A., An experimental examination of subjective forecast combination. *International Journal of Forecasting*, 1996. **12**: p. 223-233.