Inconsistency of Expert Judgment-based Estimates of Software Development Effort

Stein Grimstad[1,2], Magne Jørgensen[1]

[1]Simula Research Laboratory, P.O. Box 134, 1325 Lysaker, Norway, {steingr, magnej}@simula.no
[2]University of Oslo, Department of Informatics

Abstract
Expert judgment-based effort estimation of software development work is partly based on non-mechanical and unconscious processes. For this reason, a certain degree of intra-person inconsistency is expected, i.e., the same information presented to the same individual at different occasions sometimes lead to different effort estimates. In this paper, we report from an experiment where seven experienced software professionals estimated the same sixty software development tasks over a period of three months. Six of the sixty tasks were estimated twice. We found a high degree of inconsistency in the software professionals' effort estimates. The mean difference of the effort estimates of the same task by the same estimator was as much as 71%! The correlation between the corresponding estimates was 0,7. Highly inconsistent effort estimates will, on average, be inaccurate and difficult to learn from. It is consequently important to focus estimation process improvement on consistency issues and thereby contribute to reduced budget-overruns, improved time-to-market, and better quality software.

Keywords: Software development, effort estimation, expert judgment, inconsistency.

1 Introduction

Effort estimation is an important activity in software development and provides essential input to pricing, planning and budgeting processes (Briand and Wieczorek 2002; Coombs 2003; Sommerville 2004). Unfortunately, many software effort estimates are inaccurate and effort overruns seem to be the rule rather than the exception (Jenkins, Naumann et al. 1984; Heemstra and Kusters 1989; Moløkken-Østvold, Jørgensen et al. 2004). It is unrealistic to expect perfectly accurate estimates, even with the best estimation and development processes, since several of the factors that affect project effort can only be known after the project is completed. On the other hand, it is likely that estimation accuracy can be improved substantially by better estimation processes (Jørgensen and Sjøberg 2001; Aranda and Easterbrook 2005). Improved consistency in the use of effort estimation information and processes, which is the topic of this paper, is one possible approach to achieving more accurate effort estimates.

Greater consistency may, to some degree, be achieved by greater use of formal estimation models. In many other fields in which forecasts are made, such as the making of diagnoses in medicine, expert judgments are typically outperformed by even the simplest prediction models, partly due to the higher degree of consistency of the models

(Meehl 1957). The obvious consequence of this is that we should switch to effort estimation models instead of expert judgment in software development projects. However, the situation in software engineering seems to be different from that in many other disciplines. A recent review of fifteen studies comparing models and experts in software development effort estimation shows that the experts typically performed no worse than the models (Jørgensen 2004). One reason for this may be that it is difficult to develop meaningful estimation models that do not require a high degree of expert judgment as input to the models in the first place; that being so, the difference between models and expert judgment-based effort estimates in software development with regard to consistency may not be large. Understanding the nature and degree of inconsistency in expert judgment may consequently benefit estimation processes based on models, as well as those based on expert judgment.

In this paper we understand “degree of inconsistency” to mean how much an individual's effort estimates of the same software task, based on the same information and made under similar conditions, but made at different times, differ. If a difference in an individual's effort estimates of the same task is caused by changed conditions, e.g., by learning or the possession of new information, the difference it is not necessarily an indication of estimation inconsistency. In psychology, this judgment inconsistency is sometimes referred to as "test-retest reliability".

Forecasting research on inconsistency, e.g., (Stewart 2001), suggests that inconsistency is a major source of error in forecasts based on human judgment and that it makes learning more difficult. The forecasting research has mainly been conducted in laboratory settings, which is not surprising, since people in real-life situations seldom make judgments more than once under the same conditions. In spite of the lack of real-life studies, there are good reasons to believe that there is a high degree of inconsistency in judgments made outside the laboratory. This belief is supported by, among other things, the finding that reducing inconsistency through a mechanical combination of predictions typically leads to more accurate predictions; see, for example, (Taff, Borchering et al. 1991; Höst and Wohlin 1998; Jørgensen and Moløkken 2002). Consequently, a reduction in the degree of inconsistency in software development effort estimation may be important for improving the estimation processes.

This paper tries to contribute to this goal by providing a better understanding of the size and nature of the inconsistency in software professionals’ expert judgment-based estimation. A better understanding of the degree and nature of effort estimation inconsistency may provide valuable input for the development of improved estimation guidelines, models and processes; the selection of estimation personnel; and the design of training programmes that will lead to a more consistent use of estimation information and processes. There has, as far as we know, not been any previous study that investigates empirically software professionals’ degree of inconsistency in an effort estimation context. This means that we do not know the extent to which severe consequences of inaccurate effort estimates, e.g., budget-overruns, delayed time-to-market, and poor quality software, can be reduced by improving software professionals’ consistency in their use of estimation information and processes.

The research questions of this paper are as follows:

RQ 1: How consistent are software professionals' expert judgment-based effort estimates?

RQ 2: Do more accurate estimators have more consistent expert judgment-based effort estimates than less accurate estimators?

The remainder of the paper is organized as follows: Section 2 briefly discusses related work on inconsistency. Section 3 describes the design of our experiment. Section 4 presents the results. Section 5 discusses the results. Section 6 summarizes.

2 Related Work

Inconsistency in expert judgment has been investigated, and demonstrated, in many research fields; see, for example, (Levi 1989; Lusk and Hammond 1991). One important finding is that there are considerable domain-specific differences. For example, weather forecasters are, on average, far more consistent than stockbrokers (Shanteau, Weiss et al. 2002). Among professions studied with respect to consistency in judgments, none are, in our opinion, sufficiently similar to software development to enable the transfer of research results on consistency. Unfortunately, as stated earlier, we have been unable to find any empirical study of software professionals' individual level of effort estimation consistency.

It is, to some extent, understandable that there is a lack of studies on this subject. Such studies require, among other things, that software professionals estimate the effort of the same task at least twice, that they do not remember the first estimate on the second occasion, and that no significant amount of learning have taken place. These conditions can hardly be met in other situations than carefully designed laboratory conditions.

Most studies in which different software professionals estimate the same software development task report a high variation of effort estimates. In (Kusters, Genuchten et al. 1990), for example, 14 professional software project leaders estimated the effort of the same project. The mean effort of the 14 estimates was 28 man-months. The standard deviation of the estimates was as high as 18 man-months for expert judgment-based estimates and 14 man-months for model-based effort estimates. It is, however, not reasonable to claim that this large variation in effort estimate for the same project is a proper measure of the individuals' level of inconsistency. This would require that we made several unrealistic assumptions, e.g., that the software professionals would build the same software and have the same understanding of the (typically incomplete) specification. In our opinion, analyses based on such assumptions would be highly speculative and we have, consequently, not included these studies as reference points for our own results on individuals' degree of inconsistency regarding effort estimation.

In (Jørgensen, Faugli et al. 2007) we observed that software professionals who were more optimistic on previous effort estimation tasks were the more optimistic ones on subsequent tasks in 68% of the cases. This observation suggests that there are systematic individual differences in software professionals' estimation accuracy. The opposite result, i.e., that there were no systematic difference in estimation accuracy, would suggest that the degree of inconsistency was random and that we should not expect

to observe systematic individual differences in inconsistency in our experiment, i.e., the answer to RQ2 would be negative.

3 Study Design

The problems of examining inconsistency regarding effort estimation in real-life situations motivated our decision to investigate our research questions in a carefully-designed laboratory setting.

3.1 Previous Experiment

About one year before completion of the current experiment, we conducted an experiment in which 20 software professionals each estimated the effort and then completed five development tasks on an existing web-based database system written in Java (Gruschke and Jørgensen 2007). The current experiment is based on the effort estimation of development tasks on the same web-based database system. The research results and study material of that previous study provide essential input to the design of the current experiment.

3.2 Selection of Subjects

We selected three software professionals with high and three software professionals with low estimation accuracy from the previous experiment as subjects for the current experiment. In addition, we selected one software professional with medium estimation accuracy, for a total of seven subjects.

All subjects are experienced software consultants with Masters degrees. They were paid for their participation. None of them had received any estimation training between participation in the previous and the current experiment. Clearly, the observation of only seven software professionals is a threat to the robustness of the results. However, for practical reasons we had to choose between a study of few subjects solving many estimation tasks or many subjects solving few tasks. We considered that our research questions were better answered with the first study design option.

Our selection of subjects ensured that they had relevant previous experience with estimation and completion of similar tasks. In addition, the nonrandom selection of the 20 subjects from the previous experiment was supposed to strengthen the analysis of whether or not the most accurate estimators were also the most consistent (RQ 2), because the difference in previous estimation accuracy among the subjects is likely to be larger, compared to that of a random selection.

3.3 Estimation Tasks

The requirement specifications are based on actual change requests from the users of the system and written in natural language. The length of the specifications varied from a few lines to a full page. An example is given below.

The current system implementation accesses the database directly. Rewrite all database code to use Hibernate for database access.

Estimate the most likely work-effort it would require for you to implement and unit test this task.

Estimate of the most likely work effort _____ (work-hours)

Two senior software developers went through the requirement specifications to ensure that there were no obvious errors in the descriptions and that it was reasonable to believe that the tasks were familiar to the subjects. The two senior software developers believed that the requirement specifications represented a typical specification met in the software industry. The only notable difference was that the specifications were more precise than the average, e.g. that there was less irrelevant information than is typical in many real-world specifications. This means that the specifications are, to some extent, estimation consistency-friendly[1]. Consequently, the degree of inconsistency may be greater in real-life situations than in our experiment.

We debriefed the subjects when they had completed all tasks. In the debriefing, the subjects stated that they had perceived the tasks as realistic and all but one subject found the estimation tasks typical for tasks they normally estimate. The outlying subject's level of inconsistency was average with respect to the studied group.

3.4 Treatment

The subjects participated in three half-day sessions with approximately one month between each session. At the start of each session, the subjects received a booklet that contained 20 estimation tasks. The subjects were instructed to estimate the tasks in the same order as in the booklet. They were not allowed to go back and change previous, already completed, estimates. All seven subjects estimated the same tasks and in the same order. The tasks were estimated by expert judgment. The subjects had access to the system documentation, but not to the source code.

Six of the tasks (TT1-TT6) were used to measure the degree of inconsistency. These test tasks were estimated twice, e.g., the 10^{th} task estimated in Session 2 was identical to the 14^{th} task estimated in Session 1; see Table 1. Most tasks (48 out of 60) were estimated only once. This, together with the long period of time between the sessions, would imply, we assumed, that the subjects did not realize that they had estimated a test task before.

[1] Forecasting research suggests that when information is presented in a way that clearly emphasizes the most relevant information, consistency improves (Stewart 2001)

Table 1 Tasks (T1-60) and Test Tasks (TT1-TT6)

Session 1	*Session 2*	*Session 3*
T1	T21	T41
T2 (TT2)	T22	T42
T3	**T23 (TT5)**	T43
T4 (TT4)	**T24 (TT6)**	**T44 (TT2)**
T5	T25	T45
T6	T26	**T46 (TT4)**
T7	T27	T47
T8	T28	T48
T9	T29	T49
T10	**T30 (TT1)**	T50
T11	T31	T51
T12	T32	T52
T13	T33	T53
T14 (TT1)	**T34 (TT3)**	T54
T15	T35	T55
T16 (TT3)	T36	T56
T17	T37	T57
T18	T38	T58
T19	T39	**T59 (TT5)**
T20	T40	**T60 (TT6)**

We informed the subjects that the duration of each session was stipulated to be about four work-hours, but that they could use more time, and would be paid for it, if needed. In the debriefing, six subjects reported that the time they had used on estimation was the similar to, or greater than, the time they typically used to estimate similar tasks, while one had spent less time than usual. Time pressure may increase inconsistency (Rothstein 1986). However, we believe that the impact of time pressure on the subject who spent less than time than usual was low. This belief is supported by the observation that this subject was, on average, the third most consistent estimator.

3.5 Measurement

To compare the estimation accuracy of the subjects in the previous experiment, we applied MRE (Conte, Dunsmore et al. 1986) (Magnitude of Relative Error). MRE is a commonly used measure for estimation accuracy, and is calculated by the following formula:

$$MRE = \frac{\left| actual\,effort - estimated\,effort \right|}{actual\,effort} * 100\%$$

As noted above, six test tasks were estimated twice. Consequently, there are 42 pairs (seven subjects * six tasks) of corresponding estimates that can be used to measure inconsistency. A pair consists of two estimates of the same development task, by the same subject, in two different sessions.

We measure the relative inconsistency (RIncons) of a pair of estimates provided by subject i on test task j by the following formula:

$$RIncons(Si,TTj) = \left(\left(\frac{\max(Est1(Si,TTj), Est2(Si,TTj))}{\min(Est1(Si,TTj), Est2(Si,TTj))} \right) - 1 \right) * 100\%$$

where Si is subject i, and TTj is test task j. Est1 is the first estimate of TT j, and Est2 is the second estimate. Simplified, we measure the relative inconsistency as the ratio of the highest to the lowest effort estimate of the same task for the same subject. If, for example, S_1 estimated that he required 10 work-hours to solve TT_4 in Session 1, and 15 work-hours for the same task one month later, $RIncons(S_1,TT_4)$ = ((max(15,10)/min(15,10) – 1) * 100% = (15/10 – 1)* 100% = 50%

The relative degree of inconsistency has the advantage that it measures degree of inconsistency independently of the size of the estimates. However, for small tasks relative degree of inconsistency can be misleading, i.e., relative degree of inconsistency can be high although the absolute difference between the estimates is of no practical importance. Therefore, we also used a measure of absolute degree of inconsistency (AIncons). We define AIncons as:

$$AIncons(Si,TTj) = |Est1(Si,TTj) - Est2(Si,TTj)|$$

4 Results

4.1 Descriptive Statistics

The estimates of the six test tasks (TT1-TT6) are presented in Table 2. We did not identify any obvious outliers, e.g., due to mistyping, in the data.

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

Task	Subject 1		Subject 2		Subject 3		Subject 4		Subject 5		Subject 6		Subject 7	
	Est1	Est2	Est1	Est2	Est1	Est2	Est1	Est2	Est1	Est2	Est1	Est2	Est1	Est2
TT1	32	30	6	13	5	5	7,5	7	8	25	7	20	18	11
TT2	8	8	6	2,5	5	2	5	6	4	6	6	8	5	5,5
TT3	32	28	7	11	4	5	7	4	16	8	7	10	15	9
TT4	4	4	1	2,5	2	2	2	4	3	2	2	1	2	1,5
TT5	16	40	10	15	6	16	7	5,5	80	30	40	40	7	8
TT6	6	10	4	3	1	3	1,5	1,5	1	1	3	1	2	1

4.2 Research Question 1

We addressed Research Question 1 through examination of the difference of subjects' estimates of the same task applying the measures RIncons and AIncons; see Table 3 and Figure 1.

Table 3 RIncons (%) and AIncons (work-hours)

Subject Id	Mean		Median		Max		Min		Stdv	
	RIncons	AIncons	RIncons	AIncons	RIncons	AIncons	RIncons	AIncons	RIncons	AIncons
1	40	5,67	11	3,00	150	24,0	0	0,00	59,6	9,16
2	91	3,67	87	3,75	150	7,00	33	1,00	50,4	2,23
3	90	2,67	88	1,50	200	10,0	0	0,00	91,7	3,78
4	38	1,33	24	1,25	100	3,00	0	0,00	40,1	1,08
5	97	13,0	75	5,00	213	50,0	0	0,00	80,0	19,2
6	94	3,50	71	2,00	200	13,0	0	0,00	83,4	4,76
7	48	2,67	49	1,00	100	7,00	0	0,50	34,9	2,99
Average	71	4,64	50	2,00					66,4	8,66

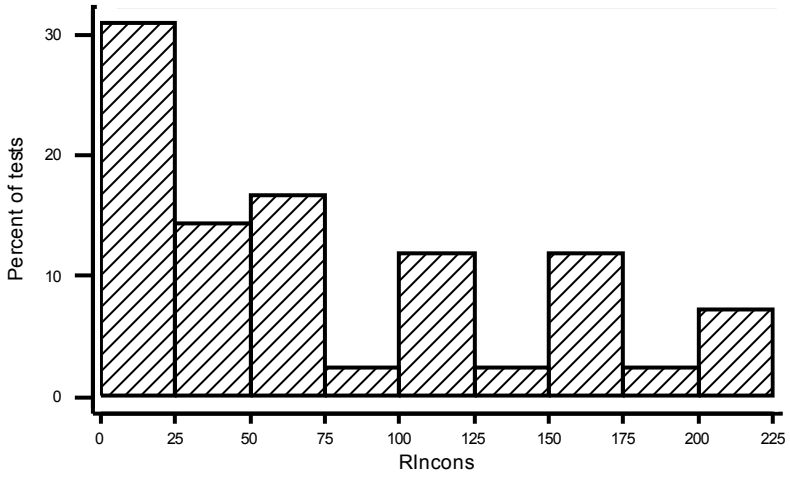

Figure 1 Histogram of relative inconsistency (RIncons)

Important observations include:

- *The degree of inconsistency is high.* Mean (median) RIncons is 71% (50%). RIncons is larger than 25% in 28 of the 42 tests. The correlation between the first and the second effort estimate of the same task is 0,7.
- *There are large individual differences in inconsistency.* The lowest mean (median) RIncons of the subjects is 40% (11%), while the highest is 97% (75%).
- *None of the subjects is consistent on all six tasks.* All subjects have RIncons of 100% or more on at least one occasion.
- *None of the tasks are consistently estimated by all subjects*. The lowest mean (median) RIncons for any test task is 54% (33%), while the highest is 86% (67%).

RIncons was, on average, higher for larger than for smaller tasks. While mean RIncons for tests tasks with both effort estimates larger than eight hours is 93%, it is 68% for test tasks with one of the estimates equal to or smaller than eight hours. This suggests that: i) the high degree of inconsistency is not explained by the fact that some of the estimated tasks are quite small, and ii) the degree of inconsistency is at least as high for large as for smaller development tasks. However, the degree of effort estimation inconsistency for large projects remains to be studied.

4.3 Research Question 2

The second research question concerns whether more accurate estimators are more consistent in their expert judgment-based effort estimates than less accurate estimators. This research question is investigated by analyzing the connection between the subjects' estimation accuracy (MRE) of the previous experiment and the degree of inconsistency (RIncons) of the current experiment. Figure 2 shows the median RIncons and the median MRE. As can be seen in Figure 2, the connection between MRE and RIncons is not strong. The correlation (r) between median MRE and median RIncons is -0,3. However, this result should be interpreted with great care, because the number of subjects is low and the impact from a few extreme observations is strong. If we, for example, remove Subject 1 (median MRE of 81% and median RIncons of 11%) from the

analysis, the correlation is 0,4 in the expected direction. Subject 1 seemed to have a different estimation process than the others. The fact that one subject affects the correlation illustrates the lack of robustness of the results from this part of the study.

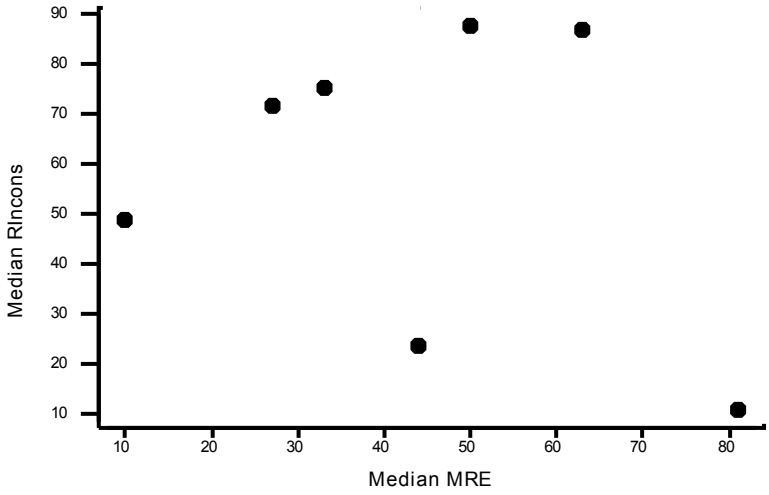

Figure 2 Inconsistency (median RIncons) vs. Accuracy (median MRE)

If it is possible to generalize from the displayed in Figure 2, it may be that we cannot select accurate effort estimators based on tests of degree of estimation consistency. However, this does not entail that estimation accuracy and inconsistency are unrelated. Clearly, individual software professionals who improve their estimation consistency are likely to improve their estimation accuracy, as well.

5 Discussion

The main result of our study is the observed high degree of inconsistency in expert judgment-based estimates. Our study was conducted in the laboratory, but we believe it is possible that real-life inconsistency is at least as large in many real-world situations. Arguments supporting our belief include the observation that inconsistency increased with larger tasks in our experiment, that the subjects perceived the experimental setting as realistic, and that the estimation tasks in the experiment in many ways were consistency-friendly. In addition, the typical higher complexity of larger projects may induce a higher degree of inconsistency.

However, the six tasks that our subjects estimated twice are rather small. Such task estimates are often aggregated into estimates of some high level activity, e.g. a project estimate. We analysed the impact of aggregations on the level of inconsistency by aggregating the subjects' first estimates of the six test tasks, their second estimates of the same six tasks, and then calculating inconsistency based on these aggregations. We found, as expected, a reduction of level of inconsistency from 50% to 36% on the aggregated task. The level of inconsistency is, however, still high. It is an open question

whether aggregation of more than six tasks would have reduced the level of inconsistency even more. This depends, among other factors, on the degree of bias in the inconsistency.

One implication of our results is that expert judgment-based effort estimates will never be very accurate if the inconsistency problem is not properly addressed. To illustrate the impact of level of inconsistency on estimation accuracy in our study, assume that the actual effort of a task in our experiment is the mean value of the two estimates of that task, i.e., that the main source of estimation error is inconsistency and not, for example, a bias towards optimism. Then, the mean MRE is 0,2. This MRE-value indicates that the best possible level of estimation accuracy is about 20% given the observed level of inconsistency, i.e., given a median RIncons of 50%.

Clearly, this calculation has its limitations. There are, for example, factors other than the ability to provide realistic estimates that affect estimation accuracy, e.g., the ability to "develop to cost" by simplifying the process or product (Grimstad and Jørgensen 2006). Such complex relationships between estimates and actual efforts make it difficult to isolate the impact of inconsistency on estimation error. Nevertheless, the high degree of inconsistency that we have measured indicates that inconsistency can, to some degree, explain the observed high estimation error in many surveys, i.e., the 30-40% development effort overrun reported in (Moløkken and Jørgensen 2003). It may also explain parts of the interestimator disagreement observed in estimation studies where different software professionals estimate the same project; see the example in Section 2.

We did not find evidence that more accurate estimators are more consistent. Some possible explanations are that: 1) the power of our study was too low to examine this relationship, 2) factors other than the ability to provide realistic estimates had a strong impact on the measured estimation accuracy, e.g., variations in the ability to develop to cost, and 3) while consistency is a necessary condition for accurate estimates, it is not a sufficient condition. There are estimation methods that are perfectly consistent, but have no predictive value. If, for example, a subject responds "10 hours" every time asked for an estimate, he or she would be perfectly consistent in spite of inaccurate effort estimates. Shantau, Weiss et al. (2002) proposed the use of the Cochran-Weiss-Shanteu (CWS) measure of expertise. This measure takes as input discrimination (ability to differentiate between cases) and inconsistency. CWS has been successfully used to identify experts in some domains, e.g., in personnel selection. We applied this measure on our data without success. The measure did not identify the most accurate estimators any better than using only inconsistency.

Many explanations have been proposed for the inconsistency of human judgment in other fields; see, for example, (Harvey 1995; Stewart 2001; Simonsohn 2007). Two examples of (partly overlapping) types of explanations are these: i) inconsistency is caused by the effects of presumably irrelevant variations in the decision situation, and ii) inconsistency is caused by cognitive limitations. An example of the first category is found in (Simonsohn 2007), where the weather influenced the weight that reviewers of college applications placed on academic attributes. An example of the second category is described in (Morera and Budescu 2001), where it is shown that a decomposing variant of a multiple-criteria decision-making technique was more consistent than a holistic variant of the same technique. However, the variety of explanations and theories also suggest that our knowledge of the causes of inconsistency is limited.

Although we know little about the causes of inconsistency, we do know something about how to reduce it. Two of the advices for improving consistency that are among the most supported in forecasting research (Stewart 2001) seem especially applicable to software effort estimation:

- *Combine estimates.* Combining effort estimates from several independent estimators is perhaps the most well-established method for improving consistency, and this approach has been investigated in software engineering contexts in several studies (Taff, Borchering et al. 1991; Höst and Wohlin 1998; Jørgensen and Moløkken 2002). Given our observation of a high level of inconsistency, we believe that more use of combination-based effort estimation would lead to substantial improvements in estimation accuracy. The empirical evidence in (Jørgensen 2007) supports this belief in the benefits of combining effort estimates. One useful approach would be to combine estimates from models and experts as advised in (Chulani, Boehm et al. 1999).
- *Present only the most important estimation information.* Empirical evidence reported in (Jørgensen and Sjøberg 2004; Grimstad and Jørgensen 2007) shows that the presence of information of no or low relevance for estimation purposes can have a strong impact on the estimates, in spite of the estimator knowing and accepting the lack of relevance.

Clearly, more research on inconsistency in estimation processes is needed. The degree of inconsistency we have measured needs to be validated against studies in other contexts and with larger samples and other populations, and the impact on estimation accuracy needs to be investigated further. We also need research on how inconsistency in estimation processes can be reduced, e.g., with respect to the effect of different types of estimation checklists or guidelines.

6 Summary

We reported on an experiment conducted to investigate the degree of inconsistency in expert judgment-based software development effort estimation. This is a topic that has received little attention in research on software estimation.

In the experiment, seven experienced software professionals estimated the most likely work-effort of the same 60 software development tasks. The subjects estimated six of the tasks twice, with at least one month between each estimate of the same task. We found that a subject's estimates of the same task differed substantially (mean difference 71%, median difference 50%). While it is no surprise that software effort estimators are inconsistent, we find this high level of inconsistency surprising. Effort was made to make the environment consistency-friendly, e.g. the specifications did not contain irrelevant information. It is therefore possible that inconsistency is even higher in many real-world settings. We can hardly ever expect accurate effort estimates when inconsistency is this high. Consequently, when attempting to improve processes of software effort estimation, and thereby contribute to reduced budget-overruns, improved time-to-market, and better quality software, it is important to focus on issues pertaining to consistency.

The difference in estimation accuracy of the subjects on previously completed development tasks did not predict degree of estimation inconsistency in the experiment. Possible explanations include the following: i) The power of our study was too low to examine this relationship, e.g., the correlation is as expected when removing one extreme observation from our experiment. ii) The estimation accuracy of individuals was affected by factors other than the degree of consistency, e.g., by the ability to develop to cost iii) A high degree of consistency may be a necessary, but not a sufficient condition for accurate estimates. Even if there should be a lack of positive correlation between software developers' median estimation accuracies and median level of inconsistency, this would not gainsay our recommendation of implementing processes that will reduce estimation inconsistency. Clearly, individual software professionals who improve their estimation consistency are likely to improve their estimation accuracy.

References

Aranda, J. and S. Easterbrook (2005). Anchoring and Adjustment in Software Estimation. European software engineering conference, Lisbon, Portugal, ACM Press.

Briand, L. C. and I. Wieczorek (2002). Resource estimation in software engineering. Encyclopedia of software engineering. J. J. Marcinak. New York, John Wiley & Sons**:** 1160-1196.

Chulani, S., B. Boehm, et al. (1999). "Bayesian analysis of empirical software engineering cost models." IEEE Transactions on Software Engineering **25**(4): 573-583.

Conte, S. D., H. E. Dunsmore, et al. (1986). Software engineering metrics and models. Menlo Park, California, Benjamin Cummings.

Coombs, P. (2003). IT Project Estimation - A Practical Guide to the Costing of Software. Cambridge, Cambridge University Press.

Grimstad, S. and M. Jørgensen (2006). A Framework for Analysis of Software Cost Estimation Error. ISESE, Rio de Janeiro, Brazil, ACM Press.

Grimstad, S. and M. Jørgensen (2007). The Impact of Irrelevant Information on Software Effort Estimates. forthcoming in 18th Australian Conference on Software Engineering, Melbourne, Australia.

Gruschke, T. M. and M. Jørgensen (2007). "How much does feedback improve software cost estimation? An Empirical Study." Paper in progress.

Harvey, N. (1995). "Why are judgments less consistent in less predictable task situations?" Organizational Behaviour and Human Decision Processes **63**: 247-263.

Heemstra, F. J. and R. J. Kusters (1989). Controlling Software Development Costs: A Field Study. International Conference on Organisation and Information Systems, Bled, Yugoslavia.

Höst, M. and C. Wohlin (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.

Jenkins, A. M., J. D. Naumann, et al. (1984). "Empirical investigation of systems development practices and results." Information and Management **7**(2): 73-82.
Jørgensen, M. (2004). "A review of studies on expert estimation of software development effort." Journal of Systems and Software **70**(1-2): 37-60.
Jørgensen, M. (2007). " Estimation of Software Development Work Effort:Evidence on Expert Judgment and Formal Models." forthcoming in International Journal of Forecasting.
Jørgensen, M., B. Faugli, et al. (2007). "Characteristics of Software Engineers with Optimistic Predictions." forthcoming in Journal of Systems and Software.
Jørgensen, M. and K. Moløkken (2002). Combination of software development effort prediction intervals: Why, when and how? Conference on Software Engineering and Knowledge Engineering, Italy.
Jørgensen, M. and D. I. K. Sjøberg (2001). "Impact of effort estimates on software project work." Information and Software Technology **43**(15): 939-948.
Jørgensen, M. and D. I. K. Sjøberg (2004). "The impact of customer expectation on software development effort estimates." International Journal of Project Management **22**: 317-325.
Kusters, R. J., M. J. I. M. Genuchten, et al. (1990). "Are software cost-estimation models accurate?" Information and Software Technology **32**(3): 187-190.
Levi, K. (1989). "Expert systems should be more accurate than human experts: Evaluation procedures from human judgment and decision making." IEEE Transactions on Systems, Man, and Cybernetics **19**(3): 647-657.
Lusk, C. M. and K. R. Hammond (1991). "Judgment in a dynamic task: Microburst forecasting." Journal of Behavioral Decision Making **4**: 55-73.
Meehl, P. E. (1957). "When shall we use our heads instead of the formula?" Journal of Counseling Psychology **4**(4): 268-273.
Moløkken, K. and M. Jørgensen (2003). A review of software surveys on software effort estimation. International Symposium on Empirical Software Engineering, Rome, Italy, Simula Res. Lab. Lysaker Norway.
Moløkken-Østvold, K., M. Jørgensen, et al. (2004). A Survey on Software Estimation in the Norwegian Industry. Metrics '04, Chicago, Illinois.
Morera, O. F. and D. V. Budescu (2001). "Random Error Reduction in Analytic Hierarchies: A Comparison of Holistic and Decompositional Decision Strategies." Journal of Behavioral Decision Making **14**: 223-242.
Rothstein, H. G. (1986). "The effects of time pressure on judgment in multiple cue probability learning." Organizational Behaviour and Human Decision Processes **37**: 83-92.
Shanteau, J., D. J. Weiss, et al. (2002). "Performance-based assessment of expertise: How can you tell if someone is an expert?" European Journal of Operations Research **136**: 253-263.
Simonsohn, U. (2007). "Clouds Make Nerds Look Good: The Impact of Environmental Cues on Attribute Weighting." forthcoming in Journal of Behavioral Decision Making.
Sommerville (2004). Software Engineering, Addison-Wesley.
Stewart, T. R. (2001). Improving Reliability of Judgmental Forecasts. Principles of Forecasting. J. S. Armstrong. Boston, Kluwer Academic Publishers**:** 81-106.

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