An Effort Prediction Interval Approach Based on the Empirical Distribution of Previous Estimation Accuracy

M. JØRGENSEN (magne.jorgensen@simula.no) & D. I. K. SJØBERG (dag.sjoberg@simula.no)
Simula Research Laboratory, Norway

Abstract

When estimating software development effort, it may be useful to describe the uncertainty of the estimate through an effort prediction interval (PI). An effort PI consists of a minimum and a maximum effort value and a confidence level. We introduce and evaluate a software development effort PI approach that is based on the assumption that the estimation accuracy of earlier software projects predicts the effort PIs of new projects. First, we demonstrate the applicability and different variants of the approach on a data set of 145 software development tasks. Then, we experimentally compare the performance of one variant of the approach with human (software professionals') judgment and regression analysis-based effort PIs on a data set of 15 development tasks. Finally, based on the experiment and analytical considerations, we discuss when to base effort PIs on human judgment, regression analysis, or our approach.

Keywords: Effort estimation, estimation uncertainty, prediction intervals, human judgment.

1 Introduction

An estimate of software development effort is typically uncertain. Hence, information about the level of uncertainty may be useful. For example, it is recommended that the proportion of a software development's project budget devoted to risk management is based on the degree of uncertainty of the effort estimate (McConnel 1998). A description of the uncertainty provides the user of the estimate with a sense of how accurate the estimate is likely to be and protects the estimator from later, unjust criticism that the estimate was wrong in situations where high uncertainty was indicated (Makridakis, Wheelwright et al. 1998). Software development effort estimates cannot be expected to be 100% accurate. A description of the level of uncertainty emphasizes this. An example of a time scheduling and estimation model that takes the uncertainty of an estimate as input is PERT (Project Evaluation and Review Technique). An overview and history of PERT is described in (Moder, Phillips et al. 1995). PERT requires, for each activity, the minimum, the most likely and the maximum effort. The PERT model provides, however, no support in *how* to derive the minimum and maximum effort.

The uncertainty of an effort estimate can be described through a prediction interval (PI). An effort PI is based on a stated certainty level and contains a minimum and a maximum effort value. For example, a project leader may estimate that the most likely effort of a project is 1000 work-hours and that it is 90% certain that the actual effort will be between 500 and 2000 work-hours. Then, the interval [500, 2000] work-hours is the 90% PI of the effort estimate of 1000 work-hours. Frequently, other terms are used instead of PI, e.g., prediction bounds, prediction limits, interval prediction, prediction region and, unfortunately, confidence interval. An important difference between confidence interval and PI is that PI refers to the uncertainty of an estimate, while confidence interval usually refers to the uncertainty associated with the parameters of an estimation model or distribution, e.g., the uncertainty of the mean value of a distribution of effort values. The confidence level of a PI refers to the expected probability that the real value is within the predicted interval (Armstrong 2001a).

While many research papers on estimation of software development effort have been published, e.g., (Boehm 1981; Albrecht and Gaffney 1983; Symons 1991; Briand, El Emam et al. 1998), little research has been conducted on methods to assess the *uncertainty* of the effort estimates through prediction interval models. The only software development research study on this topic we have been able to find is described in (Angelis and Stamelos 2000). That study compares the effort PIs derived from a bootstrap-based model (Efron and Tibshirani 1993) with the prediction intervals derived from regression-based effort estimation models.[1] The same authors present in (Stamelos and Agnelis 2001) an effort PI model for project portfolios based on a bootstrap-model. Other relevant work is, for example, (Padberg 1999), on distribution-based schedule prediction, (Chulani, Boehm et al. 1999) on use of

[1] Unfortunately, the authors compared the *confidence interval of the mean* effort for the regression based model with the *prediction interval of a new estimate* for the analogy, bootstrap based model. This means that the conclusion that the analogy, bootstrap based PIs were better than the regression model PIs is not valid. A re-analysis of their data shows that the PIs of the analogy, bootstrap based and the regression based PIs had a similar performance. This is an example of how easily PIs and confidence intervals are mixed.

Bayesian analysis to support prediction of uncertainty, and (Myrtveit and Stensrud 1999) on estimation performance with and without estimation model support.

In addition to these bootstrap and regression-based effort prediction interval approaches, simple guidelines can be found in the literature, e.g., the prediction interval suggested by NASA's Software Engineering Laboratory (SEL) (NASA 1990), see Table 1, and general guidelines on describing uncertainty and risk of software development effort, e.g., (Kitchenham and Linkman 1997). In other domains, research on the use of formal models to calculate uncertainty of estimates seems more common, e.g., uncertainty of forecasts in economics (Pindyck and Rubinfeld 1997) and measurements in physics (Ramsey 1998).

Estimation Points	Effort Prediction Intervals
End of requirements definition and specification	[Estimate / 2.0, Estimate * 2.0]
End of requirement analysis	[Estimate / 1.75, Estimate * 1.75]
End of preliminary design	[Estimate / 1.40, Estimate * 1.40]
End of detailed design	[Estimate / 1.25, Estimate * 1.25]
End of implementation	[Estimate / 1.10, Estimate * 1.10]
End of system testing	[Estimate / 1.05, Estimate * 1.05]

Table 1 NASA SEL's guidelines for estimation of effort prediction interval

It may be difficult for software professionals to understand the underlying assumptions and the calculations of regression and bootstrap-based PIs. On the other hand, the use of very simple "rules of thumb" such as the NASA SEL's guidelines may not reflect different types of projects and development organizations. According to what we have observed in industry, practitioners need a PI approach that is both simple to understand and easy to fit to an organization's own project data.

The lack of simple models on how to calculate effort PIs may be a major cause for the current situation where, as far as we have observed (Jørgensen, Teigen et al. 2002), software development effort PIs are based on human judgment. This is unfortunate, since results from (Connolly and Dean 1997) and (Jørgensen, Teigen et al. 2002) indicate that human judgment-based software development effort PIs are much too narrow to reflect high confidence levels, e.g., a 90% confidence level. This result is supported by similar results from other domains (Tversky and Kahneman 1974; Alpert and Raiffa 1982; Kahnemann, Slovic et al. 1982; Yaniv and Foster 1997). In other words, there may be a strong need for simple PI models that remove the systematic bias of the minimum and maximum effort values estimated by software professionals.

The remainder of this paper is organized as follows. Section 2 describes the approach we suggest. Section 3 exemplifies the use and validity of the approach, and compares it with human judgment and regression-based effort PIs. Finally, Section 4 concludes and suggests further work.

2 The Approach

The general principles of the approach we describe in this paper are based on well-known statistical principles of prediction intervals. Our contribution is the practical implementation and the evaluation of the principles in a software development context. This includes, for example, contributions on how to select a proper accuracy measure (Step 1) and a proper sub-set of projects with similar estimation uncertainty (Step 2). As far as we know, there has been no study implementing and evaluating the approach we describe here in a software estimation context.

The approach contains the following main steps (further explained in Section 2.1-2.4):

1. Select a measure of estimation accuracy that enables a separate analysis of bias and spread.
2. Find a set of projects with the same expected degree of estimation uncertainty as the project to be estimated. This process can, for example, be human judgment-based or based on more formal cluster analysis. A search for projects similar with respect to uncertainty may be different from a search for projects that are of similar type. Important uncertainty factors, may for example be estimation method and stability of requirement specification.
3. Display the distribution of estimation accuracy values for the selected projects. Determine how to calculate the effort PIs based on properties of the distribution.
4. Decide the confidence level, e.g., that it is 90% certain that the actual effort of the project will be within the PI, and then calculate the effort PI.

PI calculations based on the same principles are, for example, the forecasting prediction interval methods described in (Williams and Goodman 1971; Gardner 1988; Taylor and Bunn 1999; Bongaarts and Bulatao 2000). Our approach is based on the assumption that we are able to select projects where the future estimation errors will be similar to those made on the selected projects. Alternative approaches for the calculation of prediction intervals are bootstrapping (Angelis and Stamelos 2000; Pascual, Romo et al. 2001) and regression model based prediction intervals (Box and Jenkins 1976; Wonnacott and Wonnacott 1990; Makridakis, Wheelwright et al. 1998). While

bootstrap and regression require the development of a formal estimation model describing the variation in use of effort, the suggested approach is independent of how the effort estimates were derived; it can be used on expert estimates as well as analogy or regression model based estimates. The main requirement of the selected project estimates is that they have similar degree of uncertainty as the project to be estimated. Although this typically means that the projects' estimates are based on the same estimation process or model, there may be situations where expert estimates and model estimates can be mixed, e.g., a mix of expert and model estimates when there is a significant element of expert judgment in the input to the estimation model. Another advantage of our approach is that it may be relatively easy to use and understand. Our approach may, in particular, be easy to use if the selection of similar projects is based on expert judgment, and the calculation of the PIs is based on the empirical uncertainty distribution, see Section 2.3.2. In comparison, the bootstrap-based approach described in (Angelis and Stamelos 2000) requires advanced tool support for practical use.[2]

In situations without very accurate estimation models we believe that the simplicity of our approach makes it less prone to unstable over-fitting. This is in accordance with the guideline provided by Armstrong (2001b, p. 374-375): *"Select simple methods unless substantial evidence exists that complexity helps. ... One of the most enduring and useful conclusions from research on forecasting is that simple methods are generally as accurate as complex methods. ... Simplicity also aids decision makers' understanding and implementation, reduces the likelihood of mistakes and is less expensive."*

Implementations of the main steps of our approach in a software development context are described in the following sub-sections. Strengths and weaknesses compared with alternative methods are discussed in Section 3.

2.1 Accuracy Measures

The accuracy measure should enable an analysis of bias and spread of the estimates, see Figure 1. This enables separate analyses of systematic and random variance of the estimation accuracy, e.g., in which degree low accuracy effort estimates are caused by systematic over-optimism (high bias) or an inherent uncertainty (high spread). In the evaluation in Section 3.2 we illustrate the importance of this separation of bias and spread.

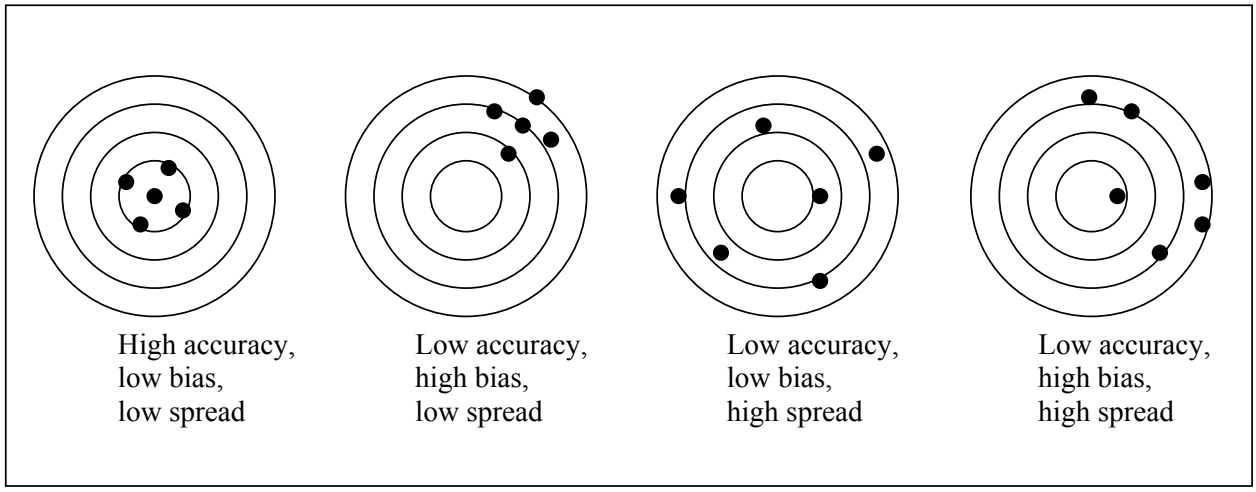

Figure 1 Relationship between accuracy, bias and spread

Based on the wish for a separate analysis of bias and spread we decided to measure estimation accuracy as the Balanced Relative Error (BRE), where Act is the actual effort and Est is the estimated effort:

$$\textbf{F1: } BRE = \begin{cases} \dfrac{Act - Est}{Act}, & Act \le Est \\ \dfrac{Act - Est}{Est}, & Act > Est \end{cases}$$

This measure has been used in time series (forecasting) studies, e.g., the "Accuracy ratio" (Q) (Armstrong 1985), and in work on unbiased, regression-based, software development effort estimation models (Miyazaki, Takanou et

[2] The authors have recently released a tool to support their method (Stamelos, Angelis et al. 2001).

al. 1991; Miyazaki, Terakado et al. 1994). In software effort estimation studies, the most common estimation accuracy measure is the Magnitude of Relative Error (MRE) (Conte, Dunsmore et al. 1986), defined as:

F2: MRE = |Act-Est| / Act

The MRE measure is not optimal for our PI calculation purpose, because a separation of bias and spread is not possible. The mean BRE can be interpreted as the bias of the estimates; the standard deviation of BRE as a measure of the spread. The same separation between bias and spread is in general impossible on the MRE values. Removing the absolute value of |Act-Est|, i.e., MRE' = (Act-Est)/Act, enables this separation. However, we still prefer the BRE measure, because a symmetric BRE distribution is more likely than a symmetric MRE' distribution in situations with large estimation inaccuracies. BRE allows, for example, a three times too high and a three times too low effort estimate to yield the same deviation from the "midpoint" of zero inaccuracy (BRE = -2 and BRE = 2). This is not the case with MRE'. No matter how much the actual effort is underestimated, MRE' cannot exceed 1, whereas overestimations lead to BRE-values with no lower limits, i.e., an asymmetric MRE' distribution is likely. Symmetry may simplify the calculation of PIs when applying parametric distributions, see Section 2.3.1. However, both BRE and MRE' are potential measures to be used in our approach, and equally suited for the empirical distribution variant, see Section 2.3.1. There are advantages and disadvantages with both measures and the selection of accuracy measure should depend on the interpretation of estimation accuracy, the purpose of the accuracy measurement and statistical properties of the measure.

There are potential problems with estimation accuracy measures. For example, when the software specification is vague, even poor estimates may become accurate when acting as a target (Jørgensen and Sjøberg 2001a), i.e., the estimate is "self-fulfilling". In that case, the estimation accuracy is not a measure of the uncertainty of the estimate, but a measure of the "flexibility" of the software product delivery. See also (Lo and Gao 1997; Hughes, Cunliffe et al. 1998) for discussions on strengths and weaknesses of software development estimation accuracy measures.

2.2 Selection of Projects

The suggested approach requires that we know the estimation accuracy of a set of completed software projects. Ideally, only the projects with expected estimation accuracy similar to the project to be estimated should be selected. Similar expected estimation accuracy may, for example, mean similar size of project, similar type of software development work, same estimator and/or similar use of estimation methods. The extensive work on software risk analysis and management, e.g., (Hall 1998), and software cost factors, e.g., (Jones 1998), may be good starting points for selecting variables that are believed to impact the uncertainty of the estimates. It is also possible to ask experienced project managers to identify earlier projects with similar level of uncertainty, i.e., to base the selection on human judgment. We have been unable to find any software development research study on this topic. Notice that the variables important for the uncertainty of an effort estimate may be different from the variables important for the calculation of the effort estimate.

2.3 Accuracy Distribution

The properties of the accuracy distribution and the size of the set of the selected tasks determine how the PI should be calculated. This paper describes two types of PI calculation methods based on respectively parametric and empirical distributions. We recommend calculations based on the parametric distributions if there are reasons to believe that the estimation accuracy can be approximated by a particular distribution, e.g., a normal distribution. On the other hand, given a "sufficient" number of previous observations, the use of the observed (empirical) estimation accuracy distribution may frequently be a better method. Then, the PIs are calculated from the percentiles of that distribution. This empirical distribution approach simplifies the assumptions and calculations, i.e., it is easier to understand and use. To understand the following sections, it is important to understand the difference between prediction intervals and confidence intervals as described in Section 1. When there are few previous observations and no knowledge about the underlying distributions, none of these approaches can be expected to perform well. Then, the use of expert judgment to develop an estimation accuracy distribution based on the available observations and other expert knowledge, and use this distribution as substitute for the pure empirical distribution, may be a better approach. If there is high inter-correlation between the accuracy values, e.g., the high estimation inaccuracy of a number of projects was caused by the same reasons, it may be important to include expert judgments to adjust the empirical distribution. Dependent observations may, for example, get a lower weight than the independent observations when establishing the estimation uncertainty distribution.

2.3.1 Parametric distribution

The statistical theory on prediction intervals, see for example (Wonnacott and Wonnacott 1990), enables us to calculate the minimum and maximum values for given confidence levels. If we can assume that the accuracy values are normally distributed, the interval end points can be calculated as:

$$\textbf{F3}: \mu \pm t(1-conf, n-1) \cdot \sigma \cdot \sqrt{\frac{1}{n}+1},$$

where *t(1-conf,n-1)* is the two-tailed value of the Student's t-distribution for the confidence level *1-conf* and *n-1* degrees of freedom, μ is the mean value, σ is the standard deviation of the observed BRE and *n* is the number of observations. Note that the "1" term inside the square root describes the spread of the estimation accuracy, while the "1/n" term describes the spread of the *mean* estimation accuracy. When the number of observations (*n*) is high, there will be almost no uncertainty about the mean estimation accuracy, but the spread of the estimation accuracy may nevertheless be large.

Assume that we wish a 90% probability that the actual value is between a minimum and a maximum value for the BRE values of the data described in (Kitchenham, Pfleeger et al. 2002). The number of observations of that data set is 145. This gives t(1-0.9,144) = 1.655. The standard deviation of the data set equals 0.51 and the mean value is -0.10. The prediction interval is, then, $-0.10 \pm 1.655 \cdot 0.51 \cdot \sqrt{(1+1/145)} = -0.10 \pm 0.84 = [-0.94, 0.74]$, i.e., it is 90% probable that the actual BRE of an estimate is between –0.94 and 0.74.

When normality cannot be assumed, then a lower probability limit (Pr) is established through Chebyshev's inequality, which holds for any random variable y:

$$\textbf{F4}: \Pr[\mu - k \cdot \sigma < y < \mu + k \cdot \sigma] \geq 1 - \frac{1}{k^2},$$

where μ is the mean value, σ is the standard deviation of the distribution and *k* is a constant ≥ 0. For example, the probability that the actual value is within the prediction interval $[\mu-2\cdot\sigma, \mu+2\cdot\sigma]$, i.e., ± two standard deviations, is at least 75% ($=1-1/2^2$).

If we can assume that the distribution is symmetric, has only one peak, and that the values decrease as one moves farther from the mode, then the probability increases to:

$$\textbf{F5}: \Pr[\mu - k \cdot \sigma < y < \mu + k \cdot \sigma] \geq 1 - \frac{1}{2.25 \cdot k^2}$$

The approximation means, for example, that the probability that the actual value is within ± two standard deviations is at least $1-1/(2.25 \cdot 2^2) = 89\%$. For data sets where the assumptions are met, adding about two standard deviations of BRE is, therefore, a simple approach to finding a prediction interval that contains the actual value with at probability of at least 89%. Similarly, adding and subtracting one standard deviation lead to a probability of at least 1-1/2.25 = 56%.

2.3.2 Empirical Distribution

The empirical distribution PI approach is based on the percentiles of the empirical distribution of accuracy values. If we choose the confidence level *conf*, then we use the percentile (1-*conf*) / 2 as the minimum value and the percentile (1+*conf*) / 2 as the maximum value.

For example, to find a 90% PI of the BRE of the 145 tasks described in (Kitchenham, Pfleeger et al. 2002) we use the 5% percentile as the minimum and the 95% percentile as the maximum BRE. This gives the BRE interval of [-0.76, 0.51][3]. A simple method to find the percentiles is to sort the BRE observations and use the rank order. For example, with 100 observations the 5% percentile is the 5th lowest BRE value and the 95% percentile is the 5th highest BRE value.

2.4 Effort PI

[3] This PI is narrower than the one based on a parametric distribution. However, there is no general rule relating these two PI approaches. The empirical distribution-based PI may be narrower, equal to or wider than the distribution-based PI depending on the shape of the accuracy distribution.

A simple reformulation of the BRE formula F1 gives:

$$\textbf{F6}: Act = \begin{cases} \dfrac{Est}{(1-BRE)}, & BRE \le 0 \\ Est \cdot (1+BRE), & BRE > 0 \end{cases}$$

Using the parametric or empirical approach we get a minimum BRE and a maximum BRE for a given confidence level. These values can be put into F6 to find the minimum and maximum actual effort values for a set of former projects, given a new estimate *Est*. The estimate may be based on expert judgment or an estimation model.

Assume, for example, that we want to estimate the 90% confidence PI of a new software project. We select a set of completed projects with estimation properties similar to the new project and find the distribution of estimation accuracy (BRE) of those projects. Based on this distribution, assume we calculate the minimum BRE to be -0.5 and the maximum BRE to be 1.5, and that the estimated effort is 100 work-hours. The BRE values inserted into F6 gives an effort PI of [100 / (1- (-0.5)), 100 * (1+1.5)] = [67, 250] work-hours. This interval is *asymmetric* around the effort estimate 100 work-hours. The reason is that the earlier effort estimates in our example were biased toward underestimation. Asymmetric PIs are frequently more realistic than symmetric PIs and can, for example, reflect that a too low estimate is more likely than a too high estimate. An interesting consequence of PIs including the bias is that we cannot exclude that the estimated effort is *outside* the PI or very close to one of the boundaries. For example, if the earlier effort estimates had been strongly biased toward too low effort estimates, the minimum BRE values may be positive and result in a minimum effort higher than the estimated effort. An effort estimate outside the PI (or close to one of the boundaries) is, of course, not meaningful and should lead to an analysis of whether the strong bias towards too high or too low estimates is expected to be present in the current estimate. If this is the case, the effort PI may be kept and the estimate revised. If not, the "bias factor" (the mean BRE) should be modified. For example, if we believe that the current estimate is totally unbiased, we may set the mean BRE of the similar projects to 0. This relatively simple bias adjustment is a clear advantage compared with the statistically more sophisticated PI approaches. Without an adjustment possibility we have to base the PIs on the assumption that the organization did not learn from previous effort estimations.

3 Evaluation of the Approach

3.1 Evaluation Measures

Evaluating effort PIs is different from evaluating effort estimates. While an estimate can be compared with the actual effort, an effort PI has no corresponding actual value. In the long run, however, an X% confidence level should correspond with X% of the actual effort values inside the effort PIs, i.e., the "hit rate" should be X%. If there is a lack of correspondence between the confidence level and the hit rate, then the effort PIs are inaccurate. In this paper we measure the hit rate as:

$$\textbf{F7:}\ HitRate = \frac{1}{n}\sum_i h_i, \quad h_i = \begin{cases} 1, & \min_i \le Act_i \le \max_i \\ 0, & Act_i > \max_i \vee Act_i < \min_i \end{cases},$$

where $\min_i$ and $\max_i$ are respectively the minimum and maximum values of the prediction interval for the effort estimate of project *i*, Act_i is the actual effort of project *i* and *n* is the number of estimated projects. For example, if 9 out of 10 effort PIs includes the actual effort, the HitRate equals 0.9.

Assume that we have two effort PI calculation methods that are applied on the same set of software projects and that they result in similar correspondence between confidence level and hit rate. In spite of this similarity there may be large differences in how "efficient" the available uncertainty estimation information has been used. Efficient use of the uncertainty information means that the effort PI can be narrower without losing the correspondence between confidence level and hit rate. Consequently, a measure able to compare efficiency, given similar levels of correspondence between confidence level and hit rate, is the median relative width of the effort PIs. We measure the width of an effort PI as:

F8: *PIWidth = (Maximum effort – Minimum effort) / Estimated effort*

We use the median PIWidth instead of the mean PIWidth to avoid a strong impact from a few very wide or very

narrow effort PIs.

3.2 Applicability of the Approach

This evaluation uses the data set described in (Kitchenham, Pfleeger et al. 2002), i.e., our example data set from Section 2. The variables of that data set describe the estimation method and size (in function points) of each project. To evaluate the validity of the PI approach, we divided the projects into four different estimation clusters, i.e., clusters with projects with similar estimation properties, see Table 2. We categorized a project according to whether the estimate was based on expert judgment, or tools or structured processes. Moreover, a project was "Small" when it had a size less than 260 function points, which was close to the median size of the tasks in that data set; otherwise it was "Large". There are many possible clustering strategies. In this case we decided, based on our research on software maintenance tasks (Jørgensen 1995) and experience as project leaders, that estimation method and size were important clustering variables. In other cases, more sophisticated statistical analyses, e.g., cluster analysis methods, may be used.

Estimation Method	Size	Cluster Id	N	Mean BRE	Std BRE	Mean MRE
Estimates based on expert judgment.	Small	1	56	-0.13	0.44	0.27
	Large	2	49	-0.07	0.63	0.25
Estimates based on tools or structured estimation processes.	Small	3	16	-0.21	0.32	0.30
	Large	4	24	0.00	0.47	0.21

Table 2 Estimation clusters

The projects in estimation clusters 1, 2 and 3 are biased toward too high (!) effort estimates, i.e., an over-pessimism. If we believe that this bias will not continue, we can adjust the mean BRE values to reflect a non-biased situation. In our evaluation, however, the bias is low and we assume that future projects will follow the same patterns as the previous projects, including the bias. The mean BRE is included to exemplify how the combination of MRE and BRE can provide information. Interestingly, we see from Table 2 that although the estimation accuracy measured as the mean MRE is similar for all types of projects, the reasons for estimation errors may be different. For example, small projects seem more prone to biases toward too high effort estimates, while large projects seem to have a larger random error component. An overestimation of effort of small tasks was also found by Gray et al. (1999).

To calculate the effort PI of project *j,* we selected the projects with the same estimation method and size category as the projects in the same cluster. This is the learning set of projects. Our evaluation is based on cross-validation (Bradley and Gong 1983), i.e., project *j* is not part of its own learning set. An improved evaluation approach would be to use the same sequence as in the organization, e.g., that the 10^{th} project used the 9 previous projects as learning data set. However, we had no information about the actual sequence of the projects and could therefore not use this approach.

Three types of effort PIs were calculated for the confidence levels 90% and 60%:[4]

PARAM 1: The PIs are based on the assumption that the BRE values are normally distributed. In our case, three out of the four clusters were too peaked to be normal. This means that we expected PARAM 1 to give too wide prediction intervals for most predictions and confidence levels for these clusters.

PARAM 2: The PIs are based on the parametric approach and that the BRE distribution is symmetric, has only one peak, and decreases as one moves farther from the mode. If these assumptions are met, Chebyshev's inequality shows that BRE_min > (mean BRE – k * std BRE) and that BRE_max < (mean BRE + k * std BRE), where k = 2 for the 89% confidence level and k = 1 for the 56% confidence level. The assumptions are reasonably well met in our case.

EMP: The PIs are based on the empirical distribution approach. In our case there seems to be sufficient historical data to apply the EMP approach.

[4] These confidence levels were chosen to exemplify respectively high (90%) and low (60%) confidence.

The results are displayed in Table 3.

PI Approach	Theoretical Confidence Level	HitRate	Median PIWidth	Informal Assessment of the PIs
PARAM 1	60%	81%	0.59	Inaccurate PIs.
	90%	94%	1.09	Accurate PIs.
PARAM 2	>56%	85%	0.68	OK lower bound (85% > 56%).
	>89%	94%	1.27	OK lower bound (94% > 89%).
EMP	60%	58%	0.30	Very accurate PIs.
	90%	85%	0.82	Accurate PIs.

Table 3 Evaluation results

Table 3 indicates that the empirical distribution approach (EMP), i.e., the simplest approach, resulted in the most accurate PIs, and there were no large differences in mean PIWidth for similar hit rates. This result is, of course, not necessarily the case for other project data sets. It is important to know *when* the different types of PIs can be expected to be accurate. The following analysis of why the PARAM 1 approach gave too wide PIs exemplifies how simple analyses of the BRE distribution can be used to understand when the PI calculations can be expected to give accurate PIs.

PARAM 1 gave much too wide PIs for the 60% confidence level. An analysis of the BRE distribution of the four estimation clusters suggests why. As explained earlier, three of these four clusters had a BRE distribution much more "peaked" than the corresponding normal curve. The difference between estimation cluster 2 (kurtosis[5] 21) and 4 (kurtosis 0.3) illustrates the difference between a "peaked" (cluster 2) and a close to normal distribution (cluster 4). PARAM 1 assumes that the BRE values follow a normal distribution. This difference means that we should expect too wide PIs of estimation cluster 2 and more appropriate PIs of estimation cluster 4. An analysis of the HitRate of the estimates of these two clusters shows that this was indeed the case, see Table 4.

Cluster	Theoretical Confidence Level	HitRate (Approach = PARAM 1)	Comments
2	60%	88%	Much too wide PIs.
	90%	96%	Too wide PIs.
4	60%	67%	OK PIs.
	90%	88%	OK PIs.

Table 4 HitRates of cluster 2 and 4 for PARAM 1

Similar analyses based on a visual inspection of the BRE distribution of different estimation clusters may be conducted to understand when and why PARAM 2 and EMP give (in)accurate PIs.

In the following, we compare only the EMP-variant of our approach with alternative approaches, i.e., regression and human judgment-based approaches. This decision was based on the good results of the EMP-approach in the study described in this section, and on the principle of selecting the simplest method if there exists no solid evidence that more complex, e.g., parametric, approaches are better.

3.3 A Comparison of Our Approach with Regression and Human Judgment Approaches

To understand *when* to use our BRE-based approach, it may be useful to compare it with regression-based and human judgment-based PI approaches. We start this comparison empirically with an experiment comparing the three approaches in Section 3.3.1. Then, we analyze the differences between the approaches more analytically in Sections 3.3.2 and 3.3.3.

3.3.1 Experiment

The experiment compares the empirical distribution-based version (EMP) of our approach with a regression-based PI approach and human judgment-based effort PIs. The 5 tasks to be used as background information (Task 1-5) and the 10 tasks to be estimated (Task 6-15) are described in Appendix 1. For comparison purposes we only use

[5] A positive kurtosis indicates a distribution more peaked than the corresponding normal curve.

the performance on the effort PIs on Task 11-15, leaving Task 1-10 as training data.

An analysis on statistical significance of performance differences between the PI approaches requires a random sample of task data from a population. Our set of tasks is, on the other hand, better described as a "convenience sample", i.e., a set of tasks we found useful to evaluate properties of the approaches. This is clearly different from a random selection of tasks from a well-defined larger population of tasks (and estimation situations/estimators), and we do therefore not include statistical testing of hypotheses in this study. The main purpose of our experiment is to aid in the understanding of *when* we can expect the different approaches to perform well, not to demonstrate that one approach is superior on the particular data set chosen.

HJ: Human judgment-based effort PIs

To provide human judgment-based effort PIs, we paid 13 experienced software professionals to participate. All the participants had at least one year experience as software developers. The average experience as software professional was 4.6 year. Six of the participants had substantial experience as project leaders. The tasks to be estimated were completed in another organization. The participants had therefore no knowledge about the applications or the developers performing the task, other than that described in the experience database (Appendix 1). This means that the relevant estimation experience was not high, and that high estimation accuracy could not be expected. This is, however, not an important limitation of this study since we are mainly interested in the software professional's ability to describe the uncertainty of their own effort estimates based on available information and feedback on previous uncertainty estimates. Less experience and limited information should lead to wider effort PIs.

The estimation process was as follows:

1. The software professionals were informed about how to interpret a 90% confidence effort PI and given 10-15 minutes on 11 training tasks on estimating minimum, most likely and maximum values for different quantities, e.g., the number of inhabitants in Spain.
2. Information about Task 1 - 5 from the experience database was presented together with information about the task to be estimated (Task 6). The participants estimated minimum, most likely and maximum effort on Task 6 based on the information about Task 1 - 5 and their own judgment about how the task characteristics of Task 6 would impact the effort. All the information about Task 6 described in the experience database (Appendix 1), except the actual effort, was presented to the participants.
3. The participants got feedback on the Task 6 estimate, i.e., they were informed about the actual effort used on the task.
4. The participants estimated the minimum, most likely and the maximum effort of Task 7 based on the information about Task 1 - 6, the feedback on the Task 6 estimates, and the information about Task 7.
5. Steps 3 and 4 were repeated for the next estimation tasks, until the estimate of Task 15 was completed.

When 7 out of 10 tasks had been estimated, i.e., when Task 12 was completed, the participants were instructed to check how many of the previous effort prediction intervals (Task 6 – 12) that had included the actual effort. The participants were reminded that, on average, 9 out of 10 actual values should be inside their 90% prediction interval. For Task 10 and 12 the participants were instructed to describe how they had derived the estimated, the minimum, and the maximum effort. The participants used on average 43 minutes on the software estimation tasks, i.e., on average 4-5 minutes on each estimation task. The human judgment-based estimates, minimum and maximum effort values are included in Appendix 2A.

As expected, the more experienced software professionals had the most accurate estimates and the highest hit rates. For example, while the participants with project leader experience had a median MRE of 0.5 and a hit rate of 0.7, the participants without project leader experience had a median MRE of 0.7 and a hit rate of 0.6.

EMP: Empirical distribution of BRE-based effort PIs

For each individual software professional (n=13) we:

1. Calculated a 90% confidence effort PI of Task 10 based on the BRE-values from the software professional on Task 6 – Task 9, applying the approach described in Section 2.3.2.
2. Estimated Task 11 based on the BRE-values of Task 6 – Task 10, Task 12 based on BRE-values of Task 6 – Task 11, etc., until the estimate of Task 15 was completed.

The minimum and maximum effort values from this approach are displayed in Appendix 2B.

REG: Regression-based effort PIs

The estimation process was as follows:

1. We calculated a 90% confidence, effort PI of Task 10 based on a log-linear regression model with variables Size (= New + Changed + Deleted Lines of Code, as suggested in (Jørgensen 1995)) and Effort based on Task 1 - Task 9. We decided to use the log-linear model of the relation between size and effort, i.e., ln(Effort) = a +

b·ln(Size), based on previous experience with regression models on a very similar data set from the same organization, i.e., (Jørgensen 1995). The regression model (based on the 10 first tasks) is not expected to be very accurate, e.g., the adjusted R^2-value was as low as 0.35. However, both parameters *a* and *b*, were significant (p=0.1 and p=0.02) and the model performed better than the naïve (default) estimation method of taking the average of earlier actual effort values, i.e., assuming that the b-parameter is zero. This indicates, we believe, that the model is meaningful, although not very accurate. Just as important, the accuracy of the effort estimates of the software professionals was even less accurate, i.e., we compare prediction intervals based on human judgment and regression model where both the types of estimates are rather inaccurate. There may be better regression models (and better expert estimators). Our comparison should therefore not be interpreted as more than an illustration of important differences between the EMP and REG PI approaches.

2. Then, we applied the regression model to estimate Task 11 based on the information about Task 1 – Task 10, estimate Task 12 based on Task 1 – Task 11, etc., until the estimate of Task 15 was completed. We applied two variants of regression model based prediction intervals:
    a. PURE-REG: The prediction intervals provided by the regression model.
    b. EMP-REG: Prediction intervals based on the previous estimation accuracy (BRE-distribution) of the regression model.

The estimates, minimum and maximum effort values from these approaches are displayed in Appendix 2C. While the HJ and EMP approaches provide a set of effort PIs for each person (in total 13 sets of effort PIs), only one set of effort PIs are calculated applying the REG approaches. The REG results are, consequently, more uncertain and less detailed than the other results. For example, the hit rates of REG can only take the values 0%, 20%, 40%, 60%, 80%, and 100%. As an illustration of differences between the approaches, the REG results are, nevertheless, meaningful.

Table 5 displays the resulting average hit rates and PIWidths for a 90% confidence level for the HJ, EMP and REG approaches. To compare the HJ and EMP approach for the same hit rate, we added EMP results for the confidence level resulting in a hit rate of 68%, i.e., the confidence level resulting in the same hit rate as the HJ approach. We use the median, not the mean, PIWidth and MRE to avoid a strong impact from a few inaccurate estimates and very wide PIs.

PI Approach	Confidence Level	Mean HitRate	Median PIWidth	Median MRE
Human judgment (HJ)	90%	68%	1.3	0.50 (Uses HJ estimates)
Empirical BRE distr. on HJ estimates (EMP_90%)	90%	82%	2.8	0.50 (Uses HJ estimates)
Empirical BRE distr. on HJ estimates (EMP_73%)	73%	68%	1.7	0.50 (Uses HJ estimates)
Pure regression (PURE-REG)	90%	80%	6.0	0.31 (Uses regression model estimates)
Empirical BRE distr. on regression model estimates (EMP-REG)	90%	80%	1.6	0.31 (Uses regression model estimates)

Table 5 Hit rates, PIWidth and MRE

There are several interesting observations possible from Table 5, e.g.:

- In spite of the training and the feedback, the average hit rate of the HJ approach (68%) was much too low to reflect the stated 90% confidence. We discuss this finding (over-confidence) in Section 3.3.2.
- The PIs of the HJ approach (median PIWidth=1.3) was narrower, although not much, than those of the EMP_90% approach (median PIWidth=1.7) for the same hit rate (68%), indicating more efficient use of the uncertainty information. We discuss this finding in Section 3.3.2.
- The average hit rates of EMP_90% (82%), PURE-REG (80%) and EMP-REG (80%) were similar and somewhat too low to reflect a 90% confidence. The median interval width was much higher for PURE-REG than for the EMP_90% and EMP-REG approaches (6.0 versus 2.8 and 1.6), indicating that these two EMP approaches made more efficient use of the uncertainty information. This difference between the EMP approaches and PURE-REG was surprising in light of the higher estimation accuracy of the regression model (median MRE=0.31) than the accuracy of HJ (median MRE=0.50). A PIWidth of 6.0 means, for example, that the effort PI of an estimate of 100 work-hours is [17; 600]! We discuss this difference in median PIWidth in Section 3.3.3.

- The median PIWidth of EMP-REG was better than that of the EMP_90% (1.6 versus 2.8 for 90% confidence), i.e., the combination of a formal model to find the estimate and an empirical distribution to find the minimum and maximum had the best performance. The regression model is only evaluated on five data points, and, as discussed earlier, the regression model results are therefore more uncertain than the other results.

3.3.2 Human Judgment

The overly narrow effort PIs (over-confidence) found in our experiment was no large surprise. In fact, the expectation that we would find an over-confidence was a major motivation for our BRE-based effort PI approach. In (Jørgensen, Teigen et al. 2002) we report even higher levels of over-confidence in industrial and student software development projects than those observed in the current experiment. In that paper we report an average hit rate of the effort PIs as low as 35% (no confidence level stated) for the industrial projects, and 62% (90% confidence) of the student projects. The students had just before the project planning started been taught that human judgment-based effort PIs were typically much too narrow. This may explain that the students performed better than the software professionals. We concluded in (Jørgensen, Teigen et al. 2002) that a major reason for the poor performance was the almost total lack of feedback on effort PIs. Other reasons for overly narrow human judgment-based effort PIs seem to be:

- Interpretation difficulties. For example, what does 90% confident mean when historical accuracy data is missing?
- Hidden agendas. In particular, the desire to be evaluated as a skilled software developer or hide lack of skill may be important agendas that lead to overly narrow effort PIs (Jørgensen, Teigen et al. 2002).
- Project managers may favor narrow intervals and high confidence, because narrow intervals make their project planning and execution easier.
- The effort estimate may become an "anchor" (Jørgensen and Sjøberg 2001b) for the minimum and maximum effort.

The relatively small difference in prediction interval width (PIWidth of 1.3 versus 1.7) for the same hit rate (68%) may indicate similarity in estimation approach between the human judgment and our EMP approach. An analysis of the described strategies of Task 10 and 12 (the participants were asked to describe their estimation strategies only on these two tasks) did not support this hypothesis. Although it was evident that the effort PIs got wider when large estimation errors were made, i.e., a behavior in accordance with our approach, only a few of the participants described an effort PI strategy similar to our approach. In fact, the most obvious result from our analysis of the described strategies was that the participants had large difficulties describing their strategies and that the strategies to a large extent were intuition-based. More (and better) studies on the PI strategies applied by software professionals are needed. Unless we achieve a better understanding of these human judgment strategies, we will be unable to improve them or know when to replace them with formal PI approaches.

Formal approaches and human judgment-based effort PIs seem to have complementary advantages and disadvantages. The experiment indicates that software professionals use the available uncertainty information more efficiently, i.e., have lower median PIWidth for the same hit rate. On the other hand, they provide effort PIs with a poor correspondence between confidence level and hit rate. Efficient use of uncertainty information is particularly important when the number of observations to learn from is low. Then, the application of formal model-based PI approaches may result in meaninglessly wide intervals (as it probably did with the regression-based approach in our experiment, see discussion in Section 3.3.3). Qualitative uncertainty knowledge, e.g., that the project to be estimated is lead by X who is a skilled project leader with good control of the project work, may provide useful information enabling a decrease in interval width without a poorer correspondence with confidence level. There is, however, little benefit in efficient use of uncertainty information if one cannot trust that there is a reasonable accurate correspondence between stated and actual confidence. There is for this reason no obvious "best choice" between human judgment and our approach regarding effort PIs. Instead, we may have to choose between efficient use of uncertainty information combined with over-confidence, and correspondence between confidence level and hit rate combined with inefficient use of uncertainty information.

3.3.3 Regression

To understand the performance of the regression-based effort PIs, it is necessary to understand some of the underlying assumptions. For this reason, we start with a brief description of regression-based effort PIs.

A (linear) regression model can be described as $Y_i = a + bX_i + \varepsilon_i$, where Y_i is the observed value of the

dependent variable, X_i is the observed value of independent variable, ε_i is the error component of observation *i*, and *a* and *b* are parameters derived from an algorithm minimizing the sum of the squared deviation between actual and estimated Y-values. Regression-based models are based on an assumption that, for every value of X, there exists a probability distribution of ε and consequently a probability distribution of Y. This probability distribution is used to calculate prediction intervals for the Y values. The validity of the use of the probability distribution of ε (the error component) is based on several assumptions (Pindyck 1997), such as:

1. The mean value of the error component equals 0, i.e., the error is unbiased.
2. The error component has constant variance for all observations.
3. The error components are statistically independent.
4. The error component is normally distributed.

The assumption that the error components have a constant variance is typically violated when estimating software development effort, because large software projects have larger deviation between the actual and the estimated effort than smaller projects. To better meet that assumption, we usually transform the variables using the logarithm, as we did in the experiment (3.3.1). A more detailed description of transformed variables, probability distribution assumptions, and extension to multiple variable regression models can be found in (Pindyck 1997).

Given that a regression model has been developed and that the underlying assumptions are met, the prediction interval of a new observation (Y_0) can be calculated from the observed independent variable (X_0) as:

F8: $$Y_0 = (a + b \cdot X_0) \pm t_{(1-conf,n-1)} \cdot s \cdot \sqrt{\frac{1}{n} + \frac{(X_0 - \overline{X})^2}{\sum x^2} + 1}$$, where

n is the number of observations,

$t_{(1\text{-conf, n-1})}$ is the two sided t-value with *n*-1 degrees of freedom,

$$s = \sqrt{\frac{\sum \varepsilon^2}{n-2}},$$

$\overline{X}$ is the mean value of *n* observed X values,

$x = X - \overline{X}$.

As can be seen from F8, there are similarities between our approach and the regression approach. For example, both use the distribution of previous estimation error to predict future error. The main differences are that the regression approach requires a formal model of the most likely effort, applies a different error measure, and is based on more assumptions. However, the increased sophistication means also that if the regression model explains the underlying phenomenon well and there is a sufficiently high number of observations to be confident about the model validity, we should expect more accurate effort PIs from the regression model than from our approach. On the other hand, when the regression model is not a good model of the underlying relationships, there are outliers not excluded from the model building data set, the variables impacting the uncertainty of the estimate are not part of the model, or the assumptions are strongly violated, then the simplicity and robustness of our approach may lead to better effort PIs. Two examples:

1) The regression-based approach includes *all* earlier deviations between estimated and actual effort in the calculation. One single inaccurate estimate may therefore strongly impact the width of the effort PI of a new project. The empirical distribution variant (EMP) of our approach, however, provides effort PIs that are not impacted by extreme values if the extreme values are "outside" the selected percentile, i.e., it may in some situations be more robust against outliers.
2) The regression model-based PIs are very sensitive to low *n* combined with inaccurate models of most likely effort, i.e., situations typical for software project effort estimation. This sensitivity to the number of observations of the regression model can lead to more appropriate effort PIs, but the lack of data can also lead to interval widths with no basis in reality. For example, when estimating the effort PI of Task 14, the regression model applied in the experiment (3.3.1) suggested a minimum effort value implying a productivity (Size/Effort) more than five times as high as the highest productivity observed on previous tasks!

4 Conclusions and Further Work

The development of models for calculating software development effort prediction intervals (PIs) seems a

neglected research area; we have found only one study (Angelis and Stamelos 2000) that suggests and/or evaluates effort PI approaches on software development projects. The contribution of this paper is the implementation and evaluation of an effort PI calculation approach that is relatively simple and can be based on any types of effort estimates, e.g., estimates based on expert judgment, regression models or analogy-based estimation models. Based on two empirical studies we provide insight into when to use our approach, regression-based approaches or software professionals' judgment. An important, but difficult, tradeoff seems to be between accuracy and efficiency. While software professionals make more efficient use of uncertainty information, they are also strongly biased towards overconfidence. Consequently, supporting and training the software professionals to reduce the overconfidence bias may be the optimal solution. One option is to train software professionals in the use of the EMP-variant of our approach, i.e., to train them in the use of previous estimation accuracy to predict effort PIs of new projects. Whether this is possible, is our main topic for further studies.

Acknowledgement

This paper has been stimulated and improved by discussions with Prof. Barbara Kitchenham, University of Keele and Dr. Scient. Magne Aldrin at the Norwegian Computing Center. The research is funded by The Research Council of Norway through the industry-project PROFIT (PROcess improvement For the IT industry).

References

Albrecht, A. J. and J. E. Gaffney 1983. Software function, source lines of code, and development effort prediction. *IEEE Transactions on Software Engineering* 9(6): 639-648.

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

Angelis, L. and I. Stamelos 2000. A simulation tool for efficient analogy based cost estimation. *Empirical software engineering* 5: 35-68.

Armstrong, J. S. 1985. *Long-Range Forecasting*. New York, John Wiley.

Armstrong, J. S. 2001a. The forecasting dictionary. *Principles of forecasting: A handbook for researchers and practitioners*. Ed. J. S. Armstrong. Boston, Kluwer Academic Publishers**:** 761-824.

Armstrong, J. S. 2001b. Selecting methods. *Principles of forecasting: A handbook for researchers and practitioners*. Ed. J. S. Armstrong, Kluwer Academic Publishers**:** 365-386.

Boehm, B. 1981. *Software Engineering Economics*. Englewood Cliffs, NJ, Prentice-Hall.

Bongaarts, J. and R. A. Bulatao 2000. *Beyond Six Billion: Forecasting the World's Population*, National Academy Press.

Box, G. E. P. and G. M. Jenkins 1976. *Time Series Analysis: Forecasting and Control*. San Francisco, Holden-Day.

Bradley, E. and G. Gong 1983. A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistican* 37(1): 36-48.

Briand, L. C., K. El Emam and F. Bomarius 1998. COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment. *Int. Conf. on Software Engineering*, Kyoto, Japan: 390-399.

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

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

Conte, S. D., H. E. Dunsmore and V. Y. Shen 1986. *Software engineering metrics and models*, Benjamin/Cummings Publishing Company Inc.

Efron, B. and R. J. Tibshirani 1993. *An introduction to the bootstrap*, Chapman & Hall/CRC.

Gardner, J., E. S 1988. A simple method of computing prediction intervals for time-series forecasts. *Management Science* 34: 541-546.

Gray, A. R., S. G. MacDonell and M. J. Shepperd 1999. Factors systematically associated with errors in subjective estimates of software development effort: the stability of expert judgment. *Proceedings Sixth International Software Metrics Symposium*: 216-227.

Hall, E. M. 1998. *Managing risk: methods for software systems development*. Reading, Massachusetts, Addison Wesley.

Hughes, R. T., A. Cunliffe and F. Young-Marto 1998. Evaluating software development effort model building techniques for application in a real-time telecommunication environment. *A-IEE Proceedings Software* 145(1): 29-33.

Jones, C. T. 1998. *Estimating software costs*, McGraw-Hill.

Jørgensen, M. 1995. An empirical study of software maintenance tasks. *Journal of Software Maintenance* 7: 27-48.

Jørgensen, M. and D. I. K. Sjøberg 2001a. Impact of effort estimates on software project work. *Information and*

Software Technology 43: 939-948.
Jørgensen, M. and D. I. K. Sjøberg 2001b. Software Process Improvement and Human Judgement Heuristics. *Scandinavian Journal of Information Systems* 13: 99-122.
Jørgensen, M., K. H. Teigen and K. Moløkken 2002. Better sure than safe? Overconfidence in judgment based software development effort prediction intervals. *Submitted to Journal of Systems and Software*.
Kahnemann, D., P. Slovic and A. Tversky 1982. *Judgement under uncertainty: Heuristics and biases*, Cambridge University Press.
Kitchenham, B. and S. Linkman 1997. Estimates, Uncertainty, and Risk. *IEEE Software* 14(3): 69-74.
Kitchenham, B., S. L. Pfleeger, B. McColl and S. Eagan 2002. A case study of maintenance estimation accuracy. *Accepted for publication in Journal of Systems and Software*.
Lo, B. W. N. and X. Gao 1997. Assessing software cost estimation models: criteria for accuracy, consistency and regression. *Australian Journal of Information Systems* 5(1): 30-44.
Makridakis, S., S. C. Wheelwright and R. J. Hyndman 1998. *Forecasting methods and applications*. New York, John Wiley & Sons.
McConnel, S. 1998. *Software project survival guide*, Microsoft Press.
Miyazaki, Y., A. Takanou, H. Nozaki, N. Nakagawa and K. Okada 1991. Method to estimate parameter values in software prediction models. *Information and Software Technology* 33(3): 239-243.
Miyazaki, Y., M. Terakado and K. Ozaki 1994. Robust regression for developing software estimation models. *Journal of Systems Software* 27(1): 3-16.
Moder, J. J., C. R. Phillips and E. W. Davis 1995. *Project management with CPM, PERT and precedence diagramming*. Wisconsin, U.S.A, Blitz Publishing Company.
Myrtveit, I. and E. Stensrud 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.
Padberg, F. 1999. A probabilistic model for software projects. *7th European Software Engineering Conference*, Toulose, Springer: 109-126.
Pascual, L., J. Romo and E. Ruiz 2001. Effects of parameter estimation on prediction densities: a bootstrap approach. *International Journal of Forecasting* 17: 83-103.
Pindyck, R. S. 1997. *Econometric models and economic forecasts*, McGrawHill.
Pindyck, R. S. and D. L. Rubinfeld 1997. *Econometric models and economic forecasts*, McGraw-Hill.
Ramsey, M. H. 1998. Sampling as a source of measurement uncertainty: techniques for quantifications and comparison with analytical sources. *Journal of Analytical Atomic Spectrometry* 13: 97-104.
Stamelos, I. and L. Agnelis 2001. Managing uncertainty in portfolio cost estimation. *Information and Software Technology* 43: 759-768.
Stamelos, I., L. Angelis and E. Sakellaris 2001. BRACE: BootstRap based Analogy Cost Estimation. *ESCOM 2001*, Londaon: 17-23.
Symons, C. 1991. *Software sizing and estimating: MkII Function Point Analysis*, J. Wiley and Sons.
Taylor, W. J. and D. J. Bunn 1999. Investigating improvements in the accuracy of prediction intervals for combinations of forecasts: A simulation study. *International Journal of Forecasting* 15: 325-339.
Tversky, A. and D. Kahneman 1974. Judgment under uncertainty: Heuristics and biases. *Science* 185: 1124-1130.
Williams, W. H. and M. L. Goodman 1971. A simple method for the construction of empirical confidence limits for economic forecasts. *Journal of the American Statistical Association* 66: 752-754.
Wonnacott, T. H. and R. J. Wonnacott 1990. *Introductory statistics*, John Wiley & Sons.
Yaniv, I. and D. P. Foster 1997. Precision and accuracy of judgmental estimation. *Journal of behavioral decision making* 10: 21-32.

APPENDIX 1 - The Estimation Tasks

Task	Type of change, Experience of developer	Productivity (LOC/work-hours)	New LOC	Chan-ged LOC	Dele-ted LOC	Actual effort (work-hours)	Comp-lexity	Size of application (LOC)
1	Change of existing functionality in application A. Development of new modules. Completed by a developer with 4 years development experience and 3.5 years experience from development on application A.	13.8	350	200	0	40	Medium	400000
2	Change of existing functionality in application B. Interface changes. Completed by a developer with 22 years development experience and 21 years experience from development on application B.	6.3	0	250	0	40	Medium	12000
3	Change of existing functionality in application B. Development of new modules. Completed by a developer with 22 years development experience and 21 years experience from development on application B. The same developer as in Task 2.	4.7	75	0	0	16	Low	12000
4	Development of new functionality in application C. Development of new modules. Completed by a developer with 9 years development experience and 0.5 years experience from development on application C.	18.8	300	0	0	16	Low	70000
5	Development of new functionality in application D. Development of new modules. Completed by a developer with 25 years development experience and 3.5 years experience from development on application D.	37.5	2100	0	0	56	Medium	165000
6	Change of existing functionality in application E. Development of new modules. Completed by a developer with 8 years development experience and 5 years experience from development on application E.	12.5	400	100	0	40	Low	320000
7	Development of new functionality in application F. Change of control flow in existing modules. Completed by developer with 2.5 years development experience and 2 years experience from development on application F.	2.5	10	0	0	4	Medium	64000
8	Development of new functionality in application A. Development of a new module. Completed by a developer with 3 years development experience and 3 years experience from development on application A.	5.0	40	0	0	8	Medium	400000
9	Development of new functionality in application A. Change of existing modules. Completed by a developer with 3 years development experience and 3 years experience from development on application A. The same developer as in Task 8.	12.5	100	0	0	8	Low	400000
10	Change of existing functionality in application G. Interface changes. Completed by a developer with 6 years development experience and 0.5 years experience from development on application G.	0.4	0	25	0	56	Medium	500000
11	Change of existing functionality in application G. Development of new modules and change of control flow in existing modules. Completed by a developer with 6 years development experience and 0.5 years experience from development on application G. The same developer as in Task 10.	1.2	80	20	40	120	Medium	500000
12	Development of new functionality in application G. Mainly development of new	7.5	300	0	0	40	Medium	500000

	modules. Completed by a developer with 8 years development experience and 4 months experience from development on application G.							
13	Development of new functionality in application A. Development of a new module and change of control flow in existing modules. Completed by a developer with 4 years development experience and 0.5 years experience from development on application A.	1.9	15	0	0	8	Low	400000
14	Development of new functionality in application A. Development of new modules. Completed by a developer with 4 years development experience and 0.5 years experience from development on application A. The same developer as in Task 13.	21.9	525	0	0	24	Low	400000
15	Change of existing modules in application A. Change of control flow in existing modules. Completed by a developer with 4 years development experience and 0.5 years experience from development on application A. The same developer as in Task 13 and 14.	10.0	200	0	200	40	Medium	400000

APPENDIX 2 – The Estimates

A) Human Judgment (90% confidence)

Task	Person	1	2	3	4	5	6	7	8	9	10	11	12	13
6	**Estimate**	30	40	42	20	24	41	40	32	40	40	40	20	40
	Min	20	20	25	10	20	35	16	16	25	25	38	17	25
	Max	40	60	100	50	40	45	56	40	55	60	43	30	100
7	**Estimate**	15	8	16	4	5	5	30	8	8	20	2	10	5
	Min	10	4	4	2	4	3	20	2	6	15	2	5	3
	Max	20	16	24	8	6	7	35	16	12	35	3	15	10
8	**Estimate**	7	8	8	3	11	6	8	8	6	8	4	7	9
	Min	5	6	2	2	8	5	4	4	4	4	4	5	6
	Max	10	12	20	5	14	8	16	16	8	16	5	15	15
9	**Estimate**	13	8	10	16	11	5	16	12	10	16	16	14	11
	Min	10	6	4	10	10	4	8	8	8	8	15	10	8
	Max	15	10	25	24	15	7	24	16	12	25	18	20	15
10	**Estimate**	5	10	4	6	50	30	6	8	10	16	12	7	7
	Min	3	6	2	2	30	25	2	6	8	8	11	4	5
	Max	15	12	12	12	60	40	8	16	16	25	14	10	11
11	**Estimate**	30	72	40	160	120	67	80	250	80	50	24	30	100
	Min	10	16	8	60	100	55	40	100	20	20	22	15	50
	Max	50	92	550	300	130	90	120	300	180	100	26	50	200
12	**Estimate**	30	200	200	200	100	162	135	150	300	50	60	60	60
	Min	20	40	40	80	80	25	80	50	260	20	55	40	30
	Max	40	300	400	400	120	200	160	500	380	75	80	80	150
13	**Estimate**	10	8	4	8	2	5	4	16	8	5	7	15	6
	Min	5	4	2	2	1	3	2	4	4	3	4	5	5
	Max	40	16	12	16	3	9	8	60	20	20	16	50	10
14	**Estimate**	30	20	60	60	100	44	35	40	50	60	40	60	80
	Min	10	16	20	20	80	30	24	16	16	30	30	30	60
	Max	50	40	100	100	120	55	40	80	100	140	56	100	100
15	**Estimate**	70	40	40	100	50	150	32	12	30	50	60	50	20
	Min	20	16	20	20	30	50	24	8	12	10	40	30	15
	Max	120	56	60	200	60	190	40	24	80	130	100	75	60

B) Empirical BRE-distribution (90% confidence)

(This approach uses the estimates provided by the software professionals, see Part A)

Task	Person	1	2	3	4	5	6	7	8	9	10	11	12	13
11	**Min**	6,1	27,1	7,0	31,6	91,1	39,5	20,0	75,3	27,9	26,3	8,1	7,2	28,4
	Max	71,7	97,9	89,6	217,6	165,0	74,1	318,4	420,0	121,6	154,0	32,6	60,6	129,5
12	**Min**	5,5	81,5	38,5	50,0	81,3	89,7	44,0	62,5	129,3	19,1	12,7	12,5	23,4
	Max	61,5	240,0	360,0	293,3	137,5	173,3	418,5	302,5	420,0	130,0	72,0	114,0	76,5
13	**Min**	2,3	4,4	1,2	2,7	1,8	2,8	2,2	10,8	4,8	2,0	1,5	3,6	4,1
	Max	17,1	17,0	16,2	17,0	2,8	6,8	14,2	34,4	17,8	10,6	10,6	26,7	8,3
14	**Min**	8,2	12,6	20,8	23,2	62,5	24,9	18,0	32,8	33,8	26,0	9,2	16,0	60,7
	Max	47,4	37,6	220,2	115,2	137,5	53,8	112,4	82,9	95,5	114,6	57,6	111,6	108,8
15	**Min**	21,5	26,0	14,7	40,3	33,0	86,3	17,2	10,2	20,6	23,0	15,3	14,5	15,4
	Max	106,4	68,8	143,2	236,0	109,3	250,5	95,7	24,7	61,8	118,0	97,2	116,3	29,3

C) Regression (90% confidence)

Task	Estimate	PURE-REG Min	PURE-REG Max	EMP-REG Min	EMP-REG Max
11	20.5	4.7	89.1	3.0	41.5
12	31.8	5.5	184.9	4.1	63.4
13	10.5	1.8	62.8	1.3	18.0
14	40.9	8.3	200.3	5.2	72.9
15	30.9	6.9	138.4	4.4	56.8