This is the author's version of an IEEE-copyrighted article. The definitive version was published in Proceedings of the 5th International Symposium on Empirical Software Engineering and Measurement (ESEM 2011), Banff, Canada (available at http://ieeexplore.ieee.org).

Handling Estimation Uncertainty with Bootstrapping: Empirical Evaluation in the Context of Hybrid Prediction Methods

Michael Kläs, Adam Trendowicz
Fraunhofer IESE
Kaiserslautern, Germany
{michael.klaes, adam.trendowicz}
@iese.fraunhofer.de

Yasushi Ishigai
Mitsubishi Research Institute
Tokyo, Japan
ishigai@mri.co.jp

Haruka Nakao
Japan Manned Space Systems Corporation
Tsukuba, Japan
haruka@jamss.co.jp

***Abstract*—Reliable predictions are essential for managing software projects with respect to cost and quality. Several studies have shown that hybrid prediction models combining causal models with Monte Carlo simulation are especially successful in addressing the needs and constraints of today's software industry: They deal with limited measurement data and, additionally, make use of expert knowledge. Moreover, instead of providing merely point estimates, they support the handling of estimation uncertainty, e.g., estimating the probability of falling below or exceeding a specific threshold. Although existing methods do well in terms of handling uncertainty of information, we can show that they leave uncertainty coming from imperfect modeling largely unaddressed. One of the consequences is that they probably provide over-confident uncertainty estimates. This paper presents a possible solution by integrating bootstrapping into the existing methods. In order to evaluate whether this solution does not only theoretically improve the estimates but also has a practical impact on the quality of the results, we evaluated the solution in an empirical study using data from more than sixty projects and six estimation models from different domains and application areas. The results indicate that the uncertainty estimates of currently used models are not realistic and can be significantly improved by the proposed solution.**

Keywords-effort estimation; defect prediction; empirical study; Monte Carlo simulation; CoBRA; HyDEEP

I. INTRODUCTION

Successful software development requires effective tools for managing the quality and cost of software. Software prediction is an essential element of key management activities such as planning, monitoring, and controlling of a project's performance in terms of product quality and cost. Ideally, credible predictions should be based on quantitative data. In practice, however, software organizations rather rarely possess sufficient amounts of reliable measurement data to base estimates on. Moreover, much of the organizational knowledge is typically represented by the subjective expertise of human experts.

In this context, using estimation methods that make use of both measurement data and expert judgment seems to be natural. Surprisingly, only a few such *hybrid estimation methods* have been proposed so far.

An essential aspect of software estimation is the explicit addressing of *estimation uncertainty*. Uncertainty is an inherent element of estimation and has many sources. On the one hand, predictions are based on imperfect (e.g., ambiguous and incomplete) information and use imperfect estimation models. On the other hand, unlike in traditional manufacturing, the environment of software development is rarely repeatable and tends to change – often in an unexpected way. An appropriate estimation method should handle uncertainty in that it explicitly considers various sources of uncertainty and supports decision makers in assessing corresponding project risks.

Handling uncertainty is particularly important for hybrid estimation methods where quantitative measurement data are scarce, expert judgments are vague, and estimation models need to combine these two sources of information in a comprehensible way. Existing hybrid estimation methods model uncertainty of information using probability distributions that are synthesized either by means of *Bayesian Theorem* (e.g., [11][26][22][10]) or *Monte Carlo simulation* (e.g., [4][30][19]). Yet, they leave the issue of the imperfect character of estimating models largely unaddressed. In this paper, we focus on two simulation-based hybrid estimation methods, CoBRA [4] and HyDEEP [18], which estimate project effort and defects, respectively. Both methods are based on the same principles – they combine causal models, probability theory, and Monte Carlo simulation (CMMC) – and have shown to be successfully applied in a number of different software development contexts (e.g., [4][25][27][29][18][19]).

Yet, CoBRA and HyDEEP also share a common open issue, which is to explicitly address the estimation uncertainty related to the imperfect character of the estimation model. In order to address this issue, we propose extending these methods by employing *bootstrapping*, a resampling technique [8]. We validate the effect of the extension on the predictive performance of both estimation methods using several real-world industrial cases. We check if employing bootstrapping results in more realistic estimates of estimation uncertainty, i.e., whether the probability distributions obtained as prediction output describe the actual level of uncertainty in the estimates more accurately.

Using the Goal-Question-Metric paradigm [3], we can recapitulate the objective of our research as follows:

Object: Analyze the CoBRA and HyDEEP hybrid prediction methods – the classic (original) methods and their extensions using bootstrapping
Purpose: for the purpose of comparative evaluation
Focus: with respect to predictive performance in terms of accuracy of uncertainty assessments
Viewpoint: from the perspective of software project decision makers
Context: in the context of six prediction models representing different application domains, companies, and cultures.

The remainder of the article is organized as follows: Section II explains the concept of existing hybrid CMMC-based estimation approaches and defines the problem addressed in this paper. Section III presents the extension of the estimation approaches followed by a brief overview of related work in Section IV. Section V provides the empirical evaluation. Finally, Section VI concludes the study and sketches future work directions.

II. Background and Problem Definition

A. Principles of CMMC-based Estimation

In this paper, we focus on two hybrid CMMC-based estimation methods: the CoBRA method for estimating software development effort [4] and the HyDEEP method for estimating software defects [19]. Both methods implement the same basic concept, although historically, the CoBRA method was developed first and then the HyDEEP method adapted its principles for software quality management.

CoBRA and HyDEEP are based on the idea (1) that the actual value of a certain *project characteristic* (P_{Act}) – in this case project effort or introduced/removed defects – can be decomposed into a context-specific *base* value (P_{Base}) and a project-specific *adjustment* coefficient (A).

$$P_{Act} = P_{Base} \times A \quad (1)$$

P_{Base} represents the value of the characteristic in a project that runs under optimal conditions – a so-called nominal project. Adjustment A represents the difference in the observed value of the characteristic in a particular, real project and the value for the nominal project. Adjustment A is measured in terms of relative percentage of P_{Base} (2). For example, in the CoBRA method, P_{Base} represents nominal project effort and the adjustment coefficient A represents the percentage of additional effort needed to overcome the non-optimal characteristics of a real project, such as project-specific time pressure, requirements volatility, etc.

$$P_{Act} = P_{Base} * (1 + A) \quad (2)$$

The adjustment coefficient A is modeled through a simple causal model (Fig. 1). The causal model consists of so-called *influencing factors*, which represent project characteristics that affect P within the considered context. The causal model is developed using expert knowledge (e.g., by involving experienced project managers). Domain experts first decide which factors have the most influence on P and thus should be considered in the model. Next, the experts quantify the impact of each individual factor in the worst case (i.e., when the factor has its worst, yet still realistic, outcome). In order to address the uncertainty of human judgment, each expert quantifies the factor's impact using a triangular distribution. This means that the expert provides three values for each factor: the minimal, maximal, and most likely percentage of the impact.

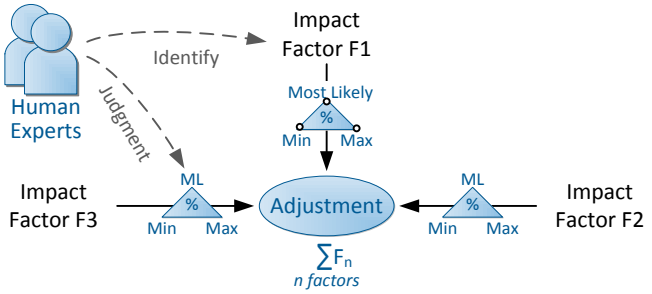

Figure 1. A simple causal model is used to determine adjustment

In order to determine the probability distribution describing the overall adjustment for a concrete project, Monte Carlo simulation is applied. It is used to combine the probability distributions provided by the experts for the different influencing factors in the model and adjusts their impact based on the given characteristics (i.e., factor levels) of the estimated project. More details on the algorithms are published in [4] and [20].

In order to predict the actual value of the project characteristic P_{Act}, the project-specific adjustment as well as the context-specific base value have to be known (2). This later value is computed using information about n already completed – historical – projects from the same context. For each historical project, the project-specific *Adjustment* A_i is computed using the quantified casual model and the factors' actual levels in the project. Based on the actual value of the project characteristic $P_{Act}(i)$ of the i-th already completed project, the base value can be computed (3).

$$P_{Base_i} = P_{Act_i}/(1 + A_i) \quad (3)$$

For a perfect causal model, we would assume that after we "extract" the impact of all factors influencing the characteristic of interest through the adjustment coefficient, the $P_{Base}(i)$ value should be constant across all n historical projects. However, in practice, due to its imperfection, the causal model does not account for all influences on the considered project phenomenon. In consequence, the P_{Base} values computed for multiple historical projects typically vary to some extent. Therefore, for the purpose of prediction, the median is taken as a robust estimator for the "real" base value (4).

$$\hat{P}_{Base} = Median(P_{Base_i}), i = 1, \dots, n \quad (4)$$

B. Deficit of CMMC-based Estimation

Taking the median as a point estimator over the multiple base values computed across the already completed projects has a drawback. By taking a point value, we lose information about the estimation uncertainty caused by the imperfection

of the estimation model. This may result in a probability distribution for the predicted project characteristic that is too narrow and, consequently, in overconfidence in the provided estimate.

III. Improving CMMC-based Predictions

In this section, we explain the approach used to adjust the investigated methods for CMMC-based estimations in order to address their deficits regarding uncertainty estimates.

A. Solution Idea

In order to address the uncertainty caused by model imperfection within CMMC-based estimations, we propose employing bootstrapping for determining the *Base* value upon which project-specific estimates are founded. Specifically, we apply Bootstrap sampling upon the *Base* values computed for multiple already completed projects and use the bootstrapped distribution as the *Base* when estimating the new project. Since the shape of the distribution of *Base* is unknown, we use a non-parametric Bootstrap method. Fig. 2 illustrates the current approach and the proposed improvement.

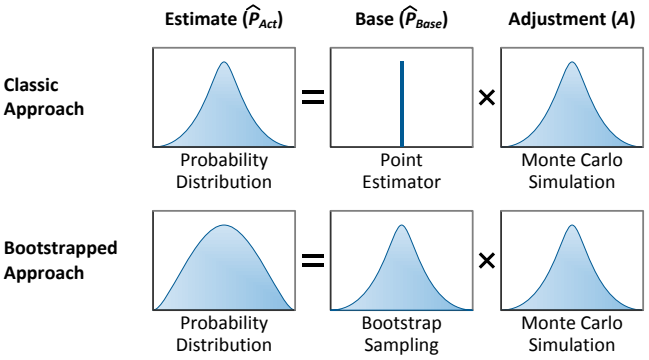

Figure 2. Extending hybrid estimation through Bootstrapping.

B. Foundations of Bootstrapping

In general, Bootstrapping belongs to the resampling methods and can be used for obtaining probability distributions for an estimator of a population parameter (such as mean, median, variance, etc.). In our study, we employ the original non-parametric Bootstrap method [8], which unlike parametric methods, does not rely on assumptions about the distribution of the estimator. Instead, non-parametric Bootstrap is based on the empirical distribution of data and relies on the assumption that the sample's distribution is a good estimate of the population's distribution.

Bootstrap treats the sample (i.e., the observed data set) of size n as the population from which it randomly draws, with replacement, a number of new samples of size n. In consequence, each "resample" could include some of the original data points more than once while it might not include others at all. Because the resamples will usually differ from the original sample, the estimator $\hat{\gamma}_i$ of the distribution parameter γ calculated based on the i-th resample will also vary for $i = 1,\ldots,n$. The relative frequency of the $\hat{\gamma}_i$ values across n bootstrapped samples is then an estimate for the distribution of $\hat{\gamma}$, which can be used to make inferences about the parameter γ.

C. Integrating Bootstrapping into CMMC-based Prediction

We integrate bootstrapping into the CMMC-based estimation methods in that instead of taking the median over the original sample of *Base* values computed across multiple already completed projects, we take the probability distribution for the median. For this purpose, we adapt the bootstrap procedure that is proposed in [9]:

1. Construct an empirical probability distribution from the set of P_{Base} values computed for n already completed projects by placing a probability of $1/n$ at each value P_{Base1}, P_{Base2}, …, P_{Basen} of the sample. Each sample's element has thus the same probability of being drawn.
2. Take a random sample of size n with replacement from the empirical distribution of P_{Base}.
3. Compute the median $Med\,(P_{Base})$ for the random sample.
4. Resample m times and compute the median $Med_i(P_{Base})$ for each i-th sample ($i = 1, \ldots, m$),
5. Construct a relative frequency histogram from the m $Med_i(P_{Base})$ values by placing a probability of $1/m$ at each value $Med_1(P_{Base})$, $Med_2(P_{Base})$, …, $Med_m(P_{Base})$. The distribution obtained is the bootstrapped estimate of the distribution of $Med(P_{Base})$. This distribution can now be used to make inferences about the parameter P_{Base} for the purpose of estimating P_{Act} (see Fig. 2 and (2)).

IV. Related Work

A. Handling Uncertainty in Hybrid Estimation Methods

In their discussion about estimation approaches [12] Jørgensen and Boehm arrived at the conclusion that "Making a one-size-fits-all decision on using models versus experts in all situations doesn't appear to be a good idea." The most important practical consequence of this finding is that combining data analysis and expert judgment in hybrid prediction methods would increase the applicability of systematic software estimation in industrial contexts, where measurement data are sparse and human judgment expensive. One key aspect to consider when combining these two estimation strategies is the handling of associated uncertainties. In the context of software estimation, several authors discuss detailed sources and causes of estimation uncertainty [15][13]. These may be generalized into two sources of uncertainty: 1. the imperfect character of estimation model (or method in case of non-model-based estimation approaches) and 2. the imperfect character of the information upon which estimates are based. In the context of hybrid software estimation, typical approaches for handling uncertainty include the use of Monte Carlo simulation [4][19][30] and the Bayes' Theorem [10].

Monte Carlo simulation is typically used for combining multiple expert judgments delivered in the form of simple probability distributions. In order to account for their uncertainty, human experts are usually asked to provide three values: minimal, maximal, and most likely. These are then interpreted as parameters of triangular or Beta-PERT

probability distribution and processed using Monte Carlo simulation to deliver final estimates in the form of probability distributions. The deficit of Monte Carlo simulation is that it focuses on the uncertainty of estimation inputs while leaving unaddressed the uncertainty associated with the estimation model or method.

Bayes' Theorem is used in the context of software estimation to infer about probability of certain events based upon initial knowledge (beliefs) and actual observations. In the approaches based upon Bayes' Theorem, human experts define the structure of the estimation model and the prior joint probability distribution of the predicted characteristic. These probability distributions are then updated to posterior distributions by means of Bayesian inference using marginal distributions of project's actual measurement data. Several studies use this approach in the context of regression models for updating regression coefficients [5][21]. Yet, the most prominent application of Bayes' Theorem for prediction purposes is represented by Bayesian Belief Networks (BBNs), where the estimation model has the form of a structural causal model [11][26][22][24][31][10]. The use of Bayes' Theorem in the context of software estimation is limited by several aspects. One aspect is that it focuses on addressing uncertainty of estimation inputs. Moreover, unlike Monte Carlo simulation, approaches based on Bayes' Theorem do not explicitly handle the aspect of multiple, potentially inconsistent, uncertain estimation inputs (e.g., judgments of multiple human experts). Another aspect, specific for BBNs, is that in practice, they are limited to discrete data. Currently, implementing BBNs for continuous or mixed data requires applying sophisticated theories.

Summarizing, existing hybrid estimation approaches focus on one aspect of estimation uncertainty, namely the uncertainty of estimation inputs, while leaving the uncertainty caused by the estimation model or method largely unaddressed.

B. Bootstrapping

To the best of our knowledge bootstrapping has not be employed so far in the context of hybrid software estimation. It has been employed in the context of data-driven software estimation based upon sparse information for approximating prediction confidence. For example, Angelis and Stamelos [1] used bootstrapping for determining prediction intervals in the context of analogy-based estimation. They compared prediction intervals based on bootstrapping and regression for the prediction of most likely project effort. The authors reported that both approaches were able to provide unbiased prediction intervals, where about 95 percent of the actual effort values were included in the 95 percent confidence prediction intervals. However, Jørgensen [13] questioned these results by claiming the model-based effort prediction intervals presented by Angelis and Stamelos were unrealistically wide. According to Jørgensen *"much of the interval width may be a result of inaccurate models of most likely effort and lack of integration of important uncertainty information i.e., most of the uncertainty is 'model uncertainty' (poor integration of knowledge) and not 'project uncertainty' (inherent uncertainty).*" Angelis and Mittas [23] addressed the comment of Jørgensen and continued in [1] their work on constructing prediction intervals using bootstrapping in the context of an effort prediction method that combines least-square regression and estimation by analogy.

V. Empirical Study

This section describes the empirical study we conducted to address the stated research objective. First, we concretize our objective and discuss how to measure whether a given probability distribution describes the actual level of uncertainty in a realistic way. Based on this, we define our hypothesis and the corresponding study design. Then, we discuss the context, the population, and our sample, e.g., the prediction models and projects we used in this study. Finally, we present the study results and test our hypothesis, discuss threats to the study's validity, and interpret the results in the context of existing work.

A. Study Goals

In our study, we focused on the following three research questions (RQ1 to RQ3), which we attempted to answer:

RQ1. Are the uncertainty estimates provided by the investigated classic CMMC-based methods accurate (i.e., realistic)?

RQ2. Are the uncertainty estimates provided by the methods when extended by bootstrapping more accurate (i.e., realistic)?

RQ3. Do specific characteristics of the investigated prediction models have an impact on the accuracy (i.e., realism) of the provided uncertainty estimates?

B. Evaluation Measures

In this section, we present, critically discuss, and finally select the measures used in our study to characterize the realism/accuracy of uncertainty estimates. Beforehand, we briefly introduce accuracy measures for point estimates, since we apply them to characterize the estimation models and projects used in our study.

For the evaluation of *point estimates* on ratio scales, a set of most commonly used standard measures [6] can be identified in the software engineering literature, even if they are partially being criticized [16]. This is not the case for the assessment of uncertainty estimates, which are typically provided in one of two forms, *prediction intervals* or *probability distributions*. Moreover, since current work focuses mainly on the evaluation of prediction intervals, we generalize some existing measures to make better use of the information provided by probability distributions.

Point estimates (PEs) provide one value that should represent the "most likely" or "expected" actual outcome of the estimated characteristic. The accuracy of point estimates is typically measured by the *magnitude of relative error* (MRE) where MRE = $|actual\ value - estimated\ value| / actual\ value$. The corresponding aggregation statistic most commonly used to describe the accuracy of the applied prediction model or approach is the *mean magnitude of relative error* (MMRE), which is the average over all MRE

values for a given set of point estimates [6]. Consequently, a smaller MMRE values means an estimation model or method provides more accurate point estimates. A second aggregation statistic commonly used for point estimates is *prediction at level 25* Pred(.25), which delivers the relative number of estimates with an MRE less than or equal to 0.25 [6].

Prediction intervals (PIs) are used to characterize the uncertainty in estimates. They comprise a minimum and maximum value and should be associated with a *confidence level* (CL) for which they are provided [2]. An example of a PI is "the outcome of the estimated characteristic is between *min* and *max* with a probability of 80%," where *min* and *max* are the lower and upper bounds of the PI and *80%* is the confidence level (Fig. 3, upper left diagram). PIs should not be confused with the concept of confidence intervals, which are provided for an aggregation statistic of a population and not for a specific estimate.

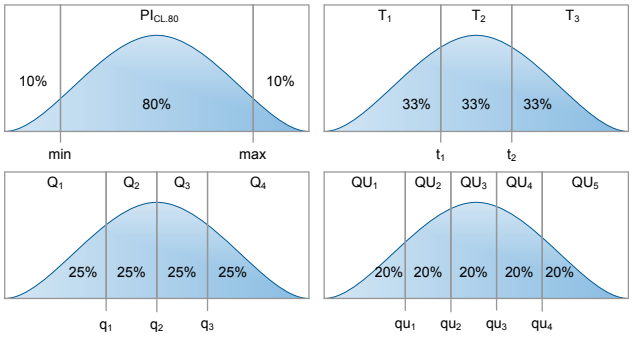

Figure 3. Prediction interval (PI), tertiles (T), quartiles (Q), quintiles (QU)

The most commonly used PI confidence level in software estimation studies is 0.90 (short $CL_{0.9}$). However, in line with Jørgensen [13], we argue for the use of PIs with a lower confidence level. In our study, we determine PIs with $CL_{0.8}$ and $CL_{0.5}$ due to the following reasons: (1) It is difficult for experts and models based on limited data to provide accurate PIs with a high CL, such as $CL_{0.9}$ or $CL_{0.95}$, since this would require considering very rare events in the estimate [14]. (2) PIs with high CL are typically too wide to be of practical use [1]. (3) A $CL_{0.8}$ is sufficient in most practical settings as only one of ten projects will exceed the upper bound of the PI and only one will go below the lower bound. (4) We consider additional PIs with $CL_{0.5}$, which provide an estimate for the "typical" range, which is not exceeded by half of the projects. Moreover, they can be used to calculate the inter-quartile fraction, which will be introduced later in this section.

The hit rate (HitR) is the measure for evaluating PIs most commonly used [23][13][17]). The underlying idea: If PIs with a confidence level of *x%* are provided for several (*n*) estimates, we would assume that an average of *x%* of actual outcomes will fall inside the lower and upper bounds of their PI. We can calculate the *hit rate* by counting the number of actual outcomes hitting their PI and dividing them by *n*. If the PIs are realistic with respect to the uncertainty they describe, we should obtain a hit rate around the chosen confidence level. If the observed hit rate is higher, the PI estimates are too wide; if the hit rate is lower, the PI estimates are too narrow (i.e., overconfident).

When applying the hit rate measure, one should consider that its accuracy depends on the number of estimates used to calculate it. This means that we would need a kind of confidence interval (≠confidence level) and statistical tests to reasonably interpret and judge the significance of the observed hit rate. However, to the best of our knowledge, previous studies using this measure did not perform such an analysis. We suggest applying resampling in order to obtain a 90% confidence interval for the calculated hit rate and performing a binomial test to check whether the difference between the expected and the obtained hit rate is significant (see hypotheses section).

Relative PI width (rWidth) is another measure proposed in [14] to evaluate PIs. It is calculated as the range of the PI (i.e., distance between the lower and upper boundaries) normalized by the estimate for the most likely outcome (i.e., the point estimate). Jørgensen et al. [14] argue that when comparing estimators with the same or a similar hit rate, the estimators that provide a lower relative width should be preferred. However, it should be mentioned that (1) the relative width expresses the precision of the PI but not whether the PI expresses the actual level of uncertainty in the estimates. (2) There is a relationship between precision (i.e., relative width) and hit rate: When the precision of an estimator with a given accuracy (i.e., measurement error) is increased, the hit rate is reduced and vice versa. (3) Since the optimal relative width depends on the estimation situation's intrinsic uncertainty, no clear target value can be provided (in contrast to the hit rate, which should be equal to the PI confidence level). Therefore, we consider the relative width as an appropriate measure for describing the precision of PIs and judging their applicability in practices but not for evaluating the realism/accuracy of uncertainty estimates provided by PIs, which is what we are mainly interested in.

The width-accuracy correlation proposed in [14] considers the correlation between the accuracy of the point estimate (e.g., measure by MRE) and the rWidth of the PI. The assumption is that estimates with a high accuracy should be also more precise (i.e., are provided with a narrower PI).

The adjusted hit rate score is a measure proposed by Mittas and Angelis [23] to compare the quality of two PIs by considering (1) their overlapping, (2) their range, and (3) whether they include the actual outcome. Based on a complex definition considering different cases, the measure tries to provide a one-number indicator for comparing the quality of prediction intervals. Due to the integration of concepts related to hit rate as well as PI precision, the estimator is difficult to interpret with respect to the realism/accuracy of the uncertainty estimates and is not considered in our study.

Probability distributions (**PDs**) describe for each possible outcome the estimated probability of this outcome. This means PDs are more powerful with respect to the extent of information they provide. Based on a given PD, the corresponding *min* and *max* values of a PI can be determined for any CL value. Moreover, the probability (*p*) that an outcome is inside a given interval can be calculated. In

return, this means that the measures that are applicable for PIs can also be applied for PDs. However, they do not use the full information provided by a PD and consequently only evaluate selected aspects of it.

The Inter-quartile fraction (IQF) is a measure proposed by Connolly and Dean [7] to assess estimates provided as probability distributions. It is similar to the hit rate measure for PIs in the way that it considers the portion of actual outcomes falling within a certain interval. In the case of the IQF, this interval is the *inter-quartile range* of the estimated probability distribution. Since the inter-quartile range is defined as the area between the first quartile (q_1) and the third quantile (q_3), it comprises 50% of the probability distribution and we would expect that 50% of all actual outcomes fall within this interval. An IQF<0.5 therefore indicates a too narrow body of the distribution, where an IQF>0.5 indicates a too wide body. Please note that based on its definition, the IQF is equivalent to the hit rate measure for PIs with $CL_{0.5}$.

Quantile-based fractions (T_i, Q_i, QU_i) generalize the inter-quartile fraction measure. They consider not whether the actual outcome is inside or outside the inter-quartile interval but how often the actual outcomes fall within each of the intervals defined by two neighboring quantiles. For instance, the quartiles q_1 to q_3 divide the PD into four parts (Q_1 to Q_4) each covering 25% of the probability (see Fig. 3, lower left diagram). The underlying idea is similar to the one of the hit rate: If PDs are provided for several estimates, we would assume that ¼ of the actual outcomes fall inside Q_1 of their PD. Naturally, the same counts for Q_2 to Q_4. The advantage over the hit-rate measure is that not only over or under-confidence can be identified, but also other biases. For instance, such a bias may be that an estimation method tends to be too optimistic and strongly underestimate the actual outcome (i.e., too many outcomes fall within Q_4) or slightly underestimate it (i.e., too many outcomes fall within Q_2).

We propose this generalization in order to get more information about the difference between estimated and actual distribution of outcomes. Since a high number of quantiles would reduce the number of actual outcomes expected for a certain quantile-based interval, we limit our investigation to *tertiles* (T), *quartiles* (Q), and *quintiles* (QU), see Fig. 3. We further suggest performing a chi-square goodness-of-fit test to check whether the distribution of the actual outcomes significantly differs from the expected equal distribution over the quantile-based intervals.

C. Hypotheses

Based on the stated research questions and measures for the concepts of interest, we formulate in this section quantitative hypotheses that we will check in our study.

RQ1: Are the uncertainty estimates provided by the investigated classic CMMC-based methods accurate (i.e., realistic)?

H_1: The investigated classic CMMC-based methods are overconfident, i.e., they deliver unrealistic narrow probability distributions for the estimated outcome. We accept H_1 if at least one of the two sub-hypotheses $H_{1.1}$, $H_{1.2}$ is accepted.

$H_{1.1}$: The prediction intervals with $CL_{0.8}$ determined using the probability distributions provided by the investigated classic CMMC-based approaches are too narrow.

$$H_{1.1}: \text{HitR}(PI_{\text{Classic,CL0.8}}) < 0.8,\ H_0: \text{HitR}(PI_{\text{Classic,CL0.8}}) \geq 0.8$$

$H_{1.2}$: The prediction intervals with $CL_{0.5}$ determined using the probability distributions provided by the investigated classic CMMC-based approaches are too narrow.

$$H_{1.2}: \text{HitR}(PI_{\text{Classic,CL0.5}}) < 0.5,\ H_0: \text{HitR}(PI_{\text{Classic,CL0.5}}) \geq 0.5$$

RQ2: Are the uncertainty estimates provided by the methods when extended by bootstrapping more accurate (i.e., realistic)?

H_2: The bootstrapped CMMC-based approaches are less overconfident, i.e., they deliver more realistic probability distributions for the estimated outcome than the classic approaches. We accept H_2 if for our sample…

$$\text{HitR}(PI_{\text{Classic,CL0.8}}) < \text{HitR}(PI_{\text{Boot,CL0.8}}) \leq 0.8 \quad \text{and}$$
$$\text{HitR}(PI_{\text{Classic,CL0.5}}) < \text{HitR}(PI_{\text{Boot,CL0.5}}) \leq 0.5 \quad \text{and}$$

at least one of the two sub-hypotheses $H_{2.1}$, $H_{2.2}$ is accepted.

$H_{2.1}$: The hit rate measured for prediction intervals with $CL_{0.8}$ determined based on the bootstrapped approaches differs from the one based on the classic approaches.

$$H_{2.1}: \mu(\text{Hits for } PI_{\text{Classic,CL0.8}}) \neq \mu(\text{Hits for } PI_{\text{Boot,CL0.8}})$$
$$H_0: \mu(\text{Hits for } PI_{\text{Classic,CL0.8}}) = \mu(\text{Hits for } PI_{\text{Boot,CL0.8}})$$

$H_{2.2}$: The hit rate measured for prediction intervals with $CL_{0.5}$ determined based on the bootstrapped approaches differ from the one based on the classic approaches.

$$H_{2.2}: \mu(\text{Hits for } PI_{\text{Classic,CL0.5}}) \neq \mu(\text{Hits for } PI_{\text{Boot,CL0.5}})$$
$$H_0: \mu(\text{Hits for } PI_{\text{Classic,CL0.5}}) = \mu(\text{Hits for } PI_{\text{Boot,CL0.5}})$$

RQ3: Do specific characteristics of the investigated prediction models have an impact on the accuracy (i.e., realism) of the provided uncertainty estimates?

Since RQ3 is explorative, we provide no hypotheses.

D. Context and Sample

The population we want to make a statement about are the predictions performed by hybrid CMMC-based prediction models. Each prediction is determined by two facts, the project for which the prediction is performed and the prediction model applied.

Our sample of hybrid prediction models and project data is a convenience sample in the way that it is based on available data. However, to the best of our knowledge, this is a general issue in any software estimation study published. Our sample comprises 61 projects and 6 corresponding prediction models in total. The data is extracted from previous studies where CMMC-based models were developed in different companies. The models provide estimates for different properties and are spread over a broad range of domains (for details, see Table I). The data have been annonymized due to confidentiality reasons. Based on the authors experience with CMMC-based models, the model characteristics such as the number of factors and historical project data included as well as their point

estimation accuracy are representative of the population of CMMC-based models.

TABLE I. STUDY INPUT: EXISTING HYBRID PREDICITON MODELS

Context		**Model Characteristics**			**Accuracy**	
Domain	***Estimated Quantity***	***No. Projects***	***Direct Factors***	***Indirect Factors***	***MMRE***	***Pred (.25)***
Web Dev.	New dev. effort	12	9	0	0.19	0.67
MIS	New dev. effort	16	15	6	0.13	0.88
Medical systems	Maintenance effort	10	4	2	0.27	0.50
Telecom-unication	Defect content	8	5	0	0.30	0.75
MIS	Maintenance effort	10	14	2	0.09	0.90
Space	Quality assur. effort	5	4	0	0.18	0.80

E. Study Design and Excecution

This section briefly describes how we designed and executed our comparative study.

In a first step, we developed and integrated the option to apply bootstrapping in the available tool support for CMMC-based models and validated our implementation of the algorithm described in Section III.C with exemplary data.

In a second step, we amassed all required input data to conduct the study. This included the information about the existing models used in our study (Table I) as well as the characteristics and actual values of the projects that were part of the historical data basis of these models.

In a third step, we estimated for each project the probability distribution for its expected outcome applying the classic CMMC-based methods without bootstrapping. In order to do this, we used the jackknife approach (also known as leave-one-out analysis) to create for each project a prediction model based on other projects from the considered context and applied this model for the estimation. To illustrate this: In order to estimate a project *i* from the web domain, we used the remaining 11 projects from this context and, based on the existing data, built a model, which we applied to project *i* to obtain the probability distribution $PD_{Classic,i}$ for its expected outcome. This means that we finally obtained a set of 61 probability distributions ($PD_{Classic,i}$ with $i=1...61$) that represent the estimated outcomes for the 61 projects using the classic CMMC-based approaches.

In a fourth step, we repeated what we did in the third step, but this time using the approaches that integrate bootstrapping. This means we obtained another set of 61 probability distributions ($PD_{Boot,i}$ with $i=1...61$).

In a fifth step, we calculated based on the measure definitions in Section V.B the data required for our analysis in the next section using the obtained probability distributions and the actual project outcome values.

F. Study Results

This section first provides a descriptive overview of the project-specific results. Next, the results for the classic and bootstrapped approaches including the corresponding test results for hypotheses H_1 and H_2 are presented and discussed. Finally, the results for different subsets of the data are presented and discussed.

Descriptive statistics: The box-and-whisker plots in Fig. 4 summarize the project-specific "row" results used to calculate the aggregation statistics and test our hypotheses.

The distribution of the *MRE* values is asymmetric (as we expected based on observations in other published studies) and has some outliers caused by point estimates strongly overestimating the actual effort. We present the MRE values independent of the applied approaches, since bootstrapping mainly affects the observed probability distribution but not the point estimate. The *relative width* of the PIs for the lower confidence level ($CL_{0.5}$) tends to be smaller than for the higher one ($CL_{0.8}$). Moreover, the PIs generated by the bootstrapped approaches tend to be broader than the ones of the classic approaches. Most relative width series (apart from $PI_{Classic,CL0.8}$) have a number of higher values, which are indicated in the diagram as outliers and therefore do not appear to be normally distributed.

Prob(x>actual) represents the estimated probability for an outcome greater than the one actually measured for the project. It is the basis for calculating the *hit rate* (HitR) and the *quantile-based fractions* (T_i, Q_i, QU_i). For instance, we can determine the hit rate for $PI_{CL0.8}$ by counting all projects with a *Prob(x>actual)* value between 10 and 90 percent and dividing the number by the total number of projects.

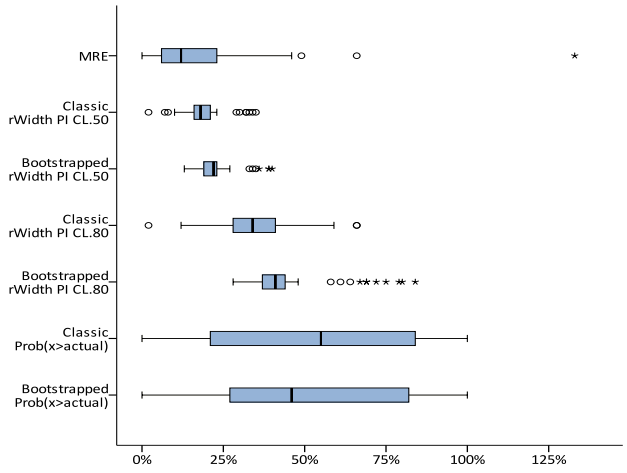

Figure 4. Results for classic and bootstrapped approaches (n=61)

RQ1 and RQ2: *Hit rate (HitR)*: Table II presents the hit rates for the classic and bootstrapped approaches. The actually observed hit rates in our sample are in all cases lower than the expected ones (i.e., the respective PI confidence levels), with the gap being higher for the classic approaches than for the bootstrapped ones. However, one should note that the hit rates measured for our sample are only estimates for the population's actual hit rate values. Therefore, we use resampling to provide a 90% confidence interval for the population's hit rate. The confidence intervals indicate that the hit rate for the classic approaches are too low.

This is the author's version of an IEEE-copyrighted article. The definitive version was published in Proceedings of the 5th International Symposium on Empirical Software Engineering and Measurement (ESEM 2011), Banff, Canada (available at http://ieeexplore.ieee.org).

TABLE II. COMPARING HIT RATES AND RELATIVE PI WIDTHS

Approach	*Confidence level (CL)*	*Hit rate*	*CI_{90} for HitRate*	*Hit rate differs* [a]	*Median rWidth*
Classic	0.50	0.41	[0.31;0.52]	p=0.200	0.18
Bootstrapped	0.50	0.46	[0.36;0.57]	p=0.609	0.22
Classic	0.80	0.64	[0.54;0.74]	**p=0.003**	0.34
Bootstrapped	0.80	**0.74** [b]	[0.66;0.84]	p=0.146	0.41

a. Binominal test with significance level = 0.05 (bold values are significant)
b. Differs significant from the Classic approaches (Wilcoxon test, α = 0.05, p = 0.014)

H_1: We tested $H_{1.1}$ and $H_{1.2}$ using a binomial test with α = 0.05. Based on the result we can reject the null hypothesis for $H_{1.1}$ and conclude that the hit rate for the population is significantly (p=0.003) too low. This means that the uncertainty estimates provided by the classic approaches are overconfident and therefore not realistic.

H_2: We tested $H_{2.1}$ and $H_{2.2}$ using the non-parametric Wilcoxon signed rank test. Based on the results we can reject the null hypothesis for $H_{2.1}$, meaning that the bootstrapped approaches significantly (p=0.014) improve the hit rate and therefore the realism of the uncertainty estimates.

Discussion of relative PI width results: The median rWidth increases when applying the bootstrapped approaches (Table II). The reason is that bootstrapping cannot improve the accuracy of the point estimates provided by the estimation method but improves the accuracy of the uncertainty estimates by adjusting the width of the confidence intervals. When comparing the observed rWidth results with the results reported in [14] for expert-based estimation, they are in a comparable region. When comparing them with the rWidth results provided by purely data-based PIs as presented in [1][23], they are much lower, i.e., they are more practical.

Discussion of width-accuracy correlation results: Based on our sample we could not observe any correlation between the accuracy of the point estimates measured by MRE and the relative width of the respective PIs. Therefore, we should not assume that estimates provided with a narrow prediction interval also provide more accurate point estimates. This observation is in line with the findings by Jørgensen et al. [14] for human-based cost estimation.

Discussion of quantile-based fractions results: Fig. 5 shows the results obtained for the quantile-based fractions measure. For the *tertiles-based fractions*, we observe that the bootstrapped approaches are very close to the expected distribution of the outcomes, whereas the classic approaches seem to have too many outcomes falling within T_3, which may be an indicator that the classic methods tend to underestimate the project outcomes. Looking at the *quartile-based fractions*, it seems that applying the classic approaches result in too many outcomes in Q_1 and Q_4, which may indicate that the classic methods favor overconfident estimates. For the bootstrapped approaches the picture is not that clear. Nevertheless, the high number of outcomes in Q_4 and the low number of outcomes in Q_3 may indicate that the bootstrapped approaches underestimate the probability of "unusually high" project outcomes. The quintile-based fractions underline this picture. However, it should be mentioned that we performed a chi-square test for each set of quantile-based fractions and none of them showed a statistically significant derivation from an equal distribution (Table III). Therefore, we should be very careful not to over-interpret the results, even if the observed overconfidence of the classic approaches is confirmed by our previous hit rate based results.

TABLE III. TESTS RESULTS FOR QUANTILE-BASED FRACTIONS

Approach	*Tertile-based fractions*	*Quantile-based fractions*	*Quintile-based fractions*
Classic	p = 0.297	p = 0.069	p = 0.418
Bootstrapped	p = 0.923	p = 0.320	p = 0.574

a. Chi-Square test for equal distribution with significance level = 0.05 (bold values are significant)

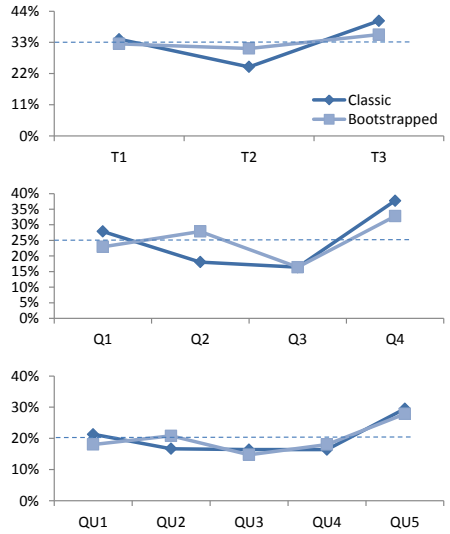

Figure 5. Quantile-based fractions, i.e., percentage of actual outcomes falling within a specific tertile (T), quartile (Q), and quintile (QU) interval

RQ3: In order to analyze whether certain characteristics of the estimation model may have an impact on the accuracy of the provided uncertainty estimates, we created subsets of the investigated projects. Each subset was defined based on one characteristic of the estimation model and analyzed with respect to the obtained hit rate and median rWidth (Table IV). Please note that the subsets intentionally overlap in part in order to obtain a reasonable number of projects in each subset.

TABLE IV. HIT RATES AND RELATIVE PI WIDTHS FOR SUBSETS

Approach	*Criteria for model selection*	*Project number*	*Confidence level (CL)*	*Hit rate*	*Median rWidth*
Classic	direct factors < 7	23	0.80	**0.48** [a]	0.32
Classic	direct factors ≥ 7	38	0.80	**0.74** [a]	0.34
Bootstr.	direct factors < 7	23	0.80	0.61	0.58
Bootstr.	direct factors ≥ 7	38	0.80	0.82	0.40
Classic	MMRE < 0.20	43	0.80	**0.72** [b]	0.35
Classic	MMRE ≥ 0.15	35	0.80	**0.49** [b]	0.30
Bootstr.	MMRE < 0.20	43	0.80	**0.84** [c]	0.41
Bootstr.	MMRE ≥ 0.15	35	0.80	**0.63** [c]	0.40
Classic	projects ≤ 10	33	0.80	0.55	0.32
Classic	projects ≥ 10	48	0.80	0.69	0.34
Bootstr.	projects ≤ 10	33	0.80	0.67	0.41
Bootstr.	projects ≥ 10	48	0.80	0.77	0.41

a. Hit rate of compared subsets differs significantly (Mann–Whitney U, α = 0.05, p=0.043)
b. Hit rate of compared subsets differs significantly (Mann–Whitney U, α = 0.05, p=0.035)
c. Hit rate of compared subsets differs significantly (Mann–Whitney U, α = 0.05, p=0.037)

Discussion hit rate results of the subset: The hit rates for the subset of models that contain a higher number of influencing factors are higher. This result is statistically significant for the classic approaches but can also be observed in its general tendency for the bootstrapped approaches. A possible explanation may be that the experts can better judge the uncertainty in their estimates when they can separate it during modeling in a larger set of independent influencing factors. The hit rate results for the subset of models with higher accuracy (i.e., lower MMRE value) are higher than for the subset of models providing less accurate point estimates. This result is statistically significant for both kinds of approaches, the classic and the bootstrapped ones. A possible explanation may be that the models with lower overall accuracy are typically also the models that comprise one or more projects with high divergence between their actual and their estimated outcome. These divergences typically are also an indicator for inaccurate uncertainty estimates and consequently a lower hit rate result. The observed hit rates for the set of models that comprise a higher number of projects are higher for both kinds of approaches. However, the level of increase is lower than for the other two separation criteria and we could not show any statistical significance of the observed increase.

Discussion relative PI width results of the subset: The median PI width varies for the different subsets between 0.30 and 0.58, with most results being between 0.32 and 0.41. There seems to be no general relation between the observed hit rate and the median relative PI width for the different data subsets. Interestingly, the bootstrapped approach provides the broadest PIs for the subset with the lowest hit rate.

G. Threats to Validity

The results of any empirical study have to be discussed with respect to their validity. In this section, threats to validity are presented that were identified during and after the conduction of this study and which are considered to be relevant. In order not to overlook any major threat, we used a checklist of typical threats [28]:

Statistical Conclusion Validity: With respect to the validity of inferences about the correlation (covariation) between study treatments and observed effects we identified as the major threat the *low statistical power*. Since we could analyze the impact of bootstrapping only on 6 project data sets with 61 data points, not all of our observations could be underlined with statistically significant test results.

Internal Validity: With respect to the validity of inferences regarding whether observed covariations reflect causal relationships between treatments and effects, we identified the threat of s*election*. Specific characteristics of projects used in the validation study may confound the effects observed. Our main selection criteria were the availability of the required model and project data and confidence in the data source. All data were extracted from high quality primary studies conducted in companies.

Construct Validity: With respect to the validity of inferences regarding the higher-order constructs investigated in the study, we see the threat of *mono-operation bias*. The measures used to quantify the accuracy of estimated uncertainty might not be adequate or account for all aspects of the uncertainty construct. We tried to address this issue by discussing the hit rate results for two confidence levels in the context of further measures proposed in the literature for evaluating uncertainty estimates.

External Validity: With respect to the validity of inferences regarding the generalizability of cause-effect relationships observed in the study, we identified two major threats. (1) There may be an *interaction of the causal relationship with the outcomes*, caused by the fact that the validation was performed on data from specific projects and the observed effect might not recur if validation were to be performed on other project data. We tried to reduce this risk by covering with our study models for different domains, focusing on different project characteristics and with differences in accuracy and complexity. (2) There may also be an *interaction of the causal relationship with the setting*. The observed effect might be caused by the particular realization of the bootstrapping. We applied the original non-parametric algorithm; however, in the literature, many enhanced approaches can be found, e.g., iterative approaches with bias correction. Moreover, the observed effects may be caused by the particular estimator (median) that is used to compute the base value within the classic hybrid estimation methods. In practice, other strategies are possible (mean, regression, etc.). Finally, we validated the impact of bootstrapping on prediction outcomes in the context of two CMMC-based estimation methods, which means they follow a specific approach for implementation hybrid estimation. These results might not recur when validated in the context of other hybrid estimation approaches.

VI. Conclusions And Future Work

Several studies have shown that estimation methods that combine measurement data and expert judgment are applicable in practice and provide accurate point estimates for project characteristics that are relevant for project decision makers such as effort and defects.

However, as illustrated in this article, the ability of hybrid prediction methods to provide realistic uncertainty estimates may suffer due to the fact that they do not appropriately consider the uncertainty introduced by imperfect models. Based on two hybrid CMMC-based prediction methods and estimation models developed in six different environments, we could show that this issue is not a theoretical one but could be observed in real-world hybrid models built with the considered prediction methods. In particular, the prediction intervals provided by existing models based on a confidence level of 0.8 are too narrow with statistical significance. This means the models are overconfident in their uncertainty estimates.

Therefore, we proposed as a possible solution the integration of bootstrapping in the considered prediction methods. The empirical results show that when extended by bootstrapping, the methods provide more realistic uncertainty estimates. The hit rate for prediction intervals with a confidence level of 0.8 could be improved with statistical significance from 0.64 for the classic approaches to 0.74 for the bootstrapped ones. At the same time, the

median relative prediction interval width increased from 0.34 to 0.41 but is still low when compared to prediction interval widths reported for purely data-based models. Further analyses indicate that the model's point estimation accuracy and the number of modeled influencing factors may have a positive impact on the realism of the uncertainty estimates provided by the model.

We consider the study results as a good starting point for further investigations on the impact of bootstrapping on the performance of hybrid estimation methods. In our opinion, the results raise two major research questions that should be addressed by further studies: Can the observed results be generalized to other hybrid estimation methods such as methods based on Bayesian Belief Networks? Are there any causalities underlying the observed correlation between specific properties of CMMC-based prediction models and the accuracy of their uncertainty estimates?

ACKNOWLEDGMENT

We would like to thank especially all study providers and the many domain experts who enabled us to analyze this set of hybrid prediction models. We would also like to thank Sonnhild Namingha from Fraunhofer IESE for the initial review of the paper. This work has been partially funded by the BMBF project Quamoco (01 IS 08 023 C).

REFERENCES

[1] L. Angelis and I. Stamelos, "A simulation tool for efficient analogy based cost estimation," *Empirical Software Engineering*, vol. 5, no. 1, pp. 35-68, 2000.

[2] J. S. Armstrong, "The forcasting dictionary," in J. S. Armstrong (Ed.), Principles of Forecasting: A Handbook for Researchers and Practitioners, Kluwer, pp. 761-824, 2001.

[3] V.R. Basili, G. Caldiera, H.D. Rombach, "Goal Question Metric Paradigm," in Encyclopedia of Software Engineering, 2nd Edition, J.J. Marciniak., vol. 1, John Wiley & Sons, 2002.

[4] L.C. Briand, K. El Emam, F. Bomarius, "COBRA: a hybrid method for software cost estimation, benchmarking, and risk assessment," in Proc. of 20th Int. Conf. on Software Engineering, pp. 390-399, 1998.

[5] S. Chulani, B.W. Boehm, B. Steece, "Bayesian analysis of empirical software engineering cost models," *IEEE Transactions on Software Engineering*, vol. 25, no. 4, pp. 573-583, 1999.

[6] S.D., Conte, H.E. Dunsmore, V. Shen, *Software engineering metrics and models*, The Benjamin-Cummings Publishing Company, 1986.

[7] T. Connolly and D. Dean, "Decompose versus holistic estimates of effort required for software writing tasks," *Management Science*, vol. 43, no. 7, pp. 1029-1045, 1997.

[8] B. Efron, "Bootstrap methods: Another look at the Jackknife". *The Annals of Statistics*, vol. 7, no. 1, pp 1–26, 1979.

[9] B. Efron and R.J. Tibshirani, *An Introduction to the Bootstrap*, Chapman and Hall, 1993.

[10] N. Fenton, et al., "On the effectiveness of early life cycle defect prediction with Bayesian Nets." *Empirical Software Engineering*, vol. 13, no. 5, pp. 499-537, 2008.

[11] N. Fenton, W. Marsh, M. Neil, P. Cates, S. Forey, M. Tailor, "Making resource decisions for software projects," Proc. of the 26th Int. Conf. on Software Engineering, pp. 397- 406, 2004.

[12] M. Jørgensen, B. Boehm, S. Rifkin, "Software development effort estimation: Formal models or expert judgment?," *IEEE Software*, vol. 26, no. 2, pp. 14-19, 2009.

[13] M. Jørgensen, "Evidence-based guidelines for assessment of software development cost uncertainty,", *IEEE Transactions on Software Engineering*, vol. 31, no. 11, pp. 942-954, 2005.

[14] M. Jørgensen, K. H. Teigen, K. Moløkken, "Better Sure than safe? Overconfidence in judgement based software development effort prediction intervals," *Journal of Systems and Software*, vol. 70, nos. 1-2, pp. 79-93, 2004.

[15] B. Kitchenham and S. Linkman, "Estimates, uncertainty, and risk," *IEEE Software*, vol. 14, no. 3, pp. 69-74, 1997.

[16] B. Kitchenham, L. Pickard, S. MacDonell, M. Shepperd, "What Accuracy Statistics Really Measure," *IEEE Software*, vol. 148, no. 3, pp. 81-85, 2001.

[17] D. Kahneman, P. Slovic, A. Tversky, *Judgement under uncertainty: heuristics and biases*, Cambridge University Press, 1982.

[18] M. Kläs, F. Elberzhager, J. Münch, K. Hartjes, O. Graevemeyer, "Transparent combination of expert and measurement data for defect prediction: an industrial case study," in Proc. of the 32nd Int. Conf. on Software Engineering, pp. 119-128, 2010.

[19] M. Kläs, H. Nakao, F. Elberzhager, J. Münch, "Support planning and controlling of early quality assurance by combining expert judgment and defect data – A case study," *Empirical Software Engineering*, vol. 15, no. 4, pp. 423-454, 2010.

[20] M. Kläs, A. Trendowicz, A. Wickenkamp, J. Münch, N. Kikuchi, Y. Ishigai, "The use of simulation techniques for hybrid software cost estimation and risk analysis," *Advances in computers*, vol. 74, pp. 115-174, 2008.

[21] C. van Koten, A.R. Gray, "Bayesian statistical effort prediction models for datacentred 4GL software development," *Information and Software Technology*, vol. 48, pp. 1056–1067, 2006.

[22] E. Mendes, "A comparison of techniques for web effort estimation," in Proc. of 1st Int. Symp. on Empirical Software Engineering and Measurement, 2007, pp. 334-343, 2007.

[23] N. Mittas and L. Angelis, "Bootstrap prediction intervals for a Semi-parametric software cost estimation model," in Proc. of the 35th Euromicro Conf. on Software Engineering and Advanced Applications, pp. 293-299, 2009.

[24] J. Moses, J. Clifford, "Learning how to improve effort estimation in small software development companies," in Proc. of the 24th Int. Computer Software and Applications Conf., pp. 522-527, 2000.

[25] H. Nakao, A. Trendowicz, J. Münch, "Estimating effort of an independent verification and validation in the context of mission-critical software systems – A case study," in Proc. of the 20th Int. Conf. on Software and Knowledge Engineering, pp. 167-172, 2008.

[26] P.C. Pendharkar, G.H. Subramanian, J.A. Rodger, "A probabilistic model for predicting software development effort," *IEEE Trans. on Software Engineering*, vol. 31, no. 7, pp. 615-624, 2005.

[27] M. Ruhe, R. Jeffery, I. Wieczorek, "Cost estimation for web applications," in Proc. of 25th Int. Conf. on Software Engineering, pp. 285-294, 2003.

[28] W.R. Shadish, T.D. Cook, D.T. Campbell, *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Wadsworth Publishing. 2nd edition , 2001.

[29] A. Trendowicz, J. Heidrich, J. Münch, Y. Ishigai, K. Yokoyama, N. Kikuchi, "Development of a hybrid cost estimation model in an iterative manner," in Proc. of the 28th Int. Conf. on Software Engineering, pp. 331-340, 2006.

[30] P. Umbers, G. Miles, "Resource estimation for web applications," in: Proc. of the 10th Int. Symp. on Software Metrics, pp. 370-381, 2004.

[31] D. Yang, et al., "COCOMO-U: An Extension of COCOMO II for cost estimation with uncertainty," in: Q. Wang, D. Pfahl, D. Raffo, P. Wernick (eds.) Software Proc. Change., Springer, pp. 132-141, 2006.