A Framework for the Analysis of Software Cost Estimation Accuracy

Stein Grimstad
Simula Research Laboratory and University of Oslo
P.O.Box 134,
1325 Lysaker, Norway
+47 67 82 82 00
steingr@simula.no

Magne Jørgensen
Simula Research Laboratory
P.O.Box 134,
1325 Lysaker, Oslo
+47 67 82 82 00
magnej@simula.no

ABSTRACT

Many software companies track and analyze project performance by measuring cost estimation accuracy. A high estimation error is frequently interpreted as poor estimation skills. This is not necessarily a correct interpretation. High estimation error can also be a result of other factors, such as high estimation complexity and insufficient cost control of the project. Through a real-life example we illustrate how the lack of proper estimation error analysis technique can bias analyses of cost estimation accuracy and lead to wrong conclusions. Further, we examine a selection of cost estimation studies, and show that they frequently do not take the necessary actions to ensure meaningful interpretations of estimation error data. Motivated by these results, we propose a general framework that, we believe, will improve analyses of software cost estimation error.

Keywords

Process improvement, software cost estimation, estimation error analysis.

1. INTRODUCTION

Software development cost estimation is difficult and inaccurate. In spite of the availability of many estimation methods and guidelines, e.g., [1-3], there is still a need for improvement. One means of reducing cost estimation error is through analysis of cost estimation error, e.g., through the identification of estimation processes that lead to more accurate estimates. Cost estimation error is relatively easy to measure. Unfortunately, as shown in this paper, the measured data can be hard to interpret properly. If we do not know what we measure, there may be little to learn from estimation error analysis. By contrast, proper measurement and analyses of estimation error may have the following beneficial results:

- identification of estimation processes resulting in systematic lowering of estimation errors
- improved evaluation and training of people responsible for estimation
- identification of (controllable and not controllable) factors that lead to estimation error, which will enable improved risk management

The goal of this paper is to show that commonly used estimation error analysis processes can lead to wrong conclusions, to document an insufficient focus on removing the effect from non-studied variable in estimation error analyses, and to propose a framework for improved analysis of software cost estimation error.

The paper is organized as follows: Section 2 describes a real-world example where a straight-forward analysis of estimation error led to misleading conclusions and our attempt to improve the analysis. Section 3 examines estimation error analyses in software cost estimation studies. Section 4 presents a framework for analysis of estimation error, and Section 5 summarizes.

2. A REAL-WORLD EXAMPLE

2.1 The Company

Software development company X had estimated the cost of their projects in two ways; some had been estimated by applying a self-developed estimation model and some by expert estimation. The estimation model was based on counting and classifying program elements, e.g., GUI-elements and program components. Each program element was classified according to size (very small, small, medium, large, very large) and technical complexity (very low, low, medium, high, very high). The estimation model suggested the required work-hours for the programming and unit test of each program element based on input of size and complexity. The estimation model had effort values calibrated to different types of development project. Since the classification of size and complexity is expert judgment-based, the estimation model may be characterized as a combination of model and expert judgment. Despite the availability of this estimation model, most projects were estimated by unsupported expert judgment-based estimation. A typical expert estimation-based process was to break the project work into activities and to estimate the effort of each activity, i.e., bottom-up, expert judgment-based estimation.

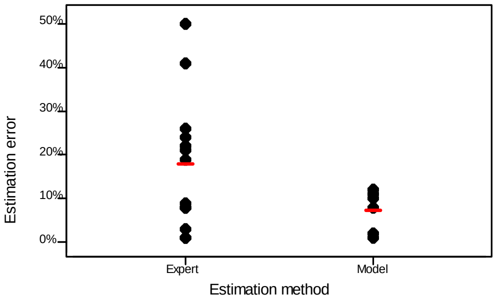
After some time, the company wanted to know whether the estimation performance was better with the model or with the expert estimation. The goal of the estimation error analysis was to compare the performance of the self-developed estimation model with that of expert estimation.

2.2 The Straight-Forward Analysis

To compare the performance of the model and the expert judgment-based effort estimates, the company collected data from 19 completed projects. The company finds that the mean estimation error was 18% for the expert judgment-based estimates, yet only 7% for the model-based ones. The difference was statistically significant (t-test of difference in mean values gave p=0.04). The estimation error data are displayed in Figure 1, which clearly suggests that that more use of model-based estimation would lead to lower estimation error.

Figure 1: Estimation Error Data

(means are indicated by lines)



This straight-forward analysis of real-world estimation error data tells a convincing story about the benefits of using the estimation model. Unfortunately, as we discovered as part of a larger study of the organization's estimation performance, this convincing story is probably flawed.

2.3 Our Attempt to Improve the Analysis

The analysis in Section 2.2 is what could have happened if a typical data collection and analysis process had been followed. This section describes how we conducted the data collection and the analysis and how this changes the conclusion.

Our attempt to improve the estimation error analysis started with an examination of the implications of the goal of the analysis. The general *goal* of the analysis was to get more knowledge about the estimation ability (performance) of two different categories of estimation method: the estimation model and the expert estimation. Clarifying the goal, we found that the estimation accuracy of the smallest projects, here defined as projects with less than 100 work-hours, was not essential for the organization and they should, consequently, be excluded from the analysis. Similarly, very large projects were excluded from the analysis for two reasons: (i) they were considered to be substantially different from the other projects with respect to estimation complexity, and (ii) there were too few to be analyzed by statistical means. The factor to be analyzed was, consequently, the relative estimation performance of two estimation methods for medium-size (between 100 and 5000 work-hours) projects.

From previous experience we knew that there might be systematic method use biases with respect to when formal estimation models and expert estimation are used. In particular, we anticipated that:

• Early estimates based on limited information would usually be based on expert estimation, because experts are better able to cope with incomplete data and non-standard data formats than the model.

• The calibration of the formal estimation model required data from similar projects. This means that expert estimation may have been the only possible option for projects when the company had few similar projects.

In addition, we decided to collect data related to the following factors that could potentially bias the comparison of the estimation error of the model and the expert judgment:

• Difference between estimated and actual functionality.

• Unusual reasons for estimation error.

• Differences in the project's priorities on cost control.

Other differences, e.g., difference in the level of estimation skill between users of estimation models and expert estimation, were assumed to be small or unlikely to be more systematically associated more with one estimation method than the other. We assumed, based on information about the experience of the estimators, that the estimation model was applied properly. As can be seen, the identification of factors of interest did require knowledge about the studied process and the organization. Notice also that the estimation model we compare with expert estimation is a rather simple two-parameter (size and technical complexity) model. More sophisticated formal estimation models may, of course, perform better and this is no general evaluation of models vs. experts.

The second author of this paper was fortunate in that he was able to influence the amount and quality of the estimation related data that were logged. Estimation-related information was collected and logged by the estimators as soon as a project estimate was completed and immediately after the project was completed.

We instructed the estimators to interpret "effort estimate" as "most likely effort" when logging the estimation data. In addition, we required that the estimation process should be described briefly for each estimate. This description of the estimation process enabled us to check whether the estimate was a most likely effort estimate and to check the actual use of the estimation methods. We used MRE (=|actual effort – estimated effort|/actual effort) and RE (=(actual effort – estimated effort)/actual effort) as measures of estimation error.

We decided to use a combination of exclusion, adjustment and grouping to isolate the estimation performance factor:

1) Exclude projects with very unusual reasons for estimation errors or where the dominant reason for high estimation accuracy or error was clearly not related to the performance of the estimation method.

2) Adjust the actual effort for large differences, i.e., more than 10%, between estimated and actual functionality.

3) Compare estimation error collected in similar estimation situations, e.g., similar with respect to:

• Time of estimation: a) early estimate used as input to bidding, and, b) estimate used as input for planning.

• Cost control priority of the project: a) priority on cost control, b) priority on time control or quality.

• Estimation complexity: a) the estimator had experience from a similar projects, b) the estimator did not have experience from a similar project.

Estimation data about 56 projects were collected. As many as 21 of these projects were either not started or never completed. This may have lead to some serious measurement selection bias if we aimed at measurement of the organization's estimation skill. We assessed, however, that the reasons for non-completions were not related to use of estimation method and that this therefore would not bias our comparison of the two estimation methods.

Among the 35 completed projects, 19 had a size larger than 100 work-hours. Among those 19 projects, 13 were based on expert judgment and six on the model. (These 19 projects are the same as those displayed in Figure 1).

In spite of our instruction of interpreting 'estimate' as 'most likely effort', we suspected, from the analysis of the estimation process descriptions, that there were variation in what the estimators meant by 'estimate'. Our analysis is based on the assumption that this will not bias the comparison. For future measurements we would recommend that the organization trained the estimators in consistent use of estimation terminology.

As shown in Section 2.2 a straight-forward analysis of the difference between the two estimation methods suggested that the estimation model led to significantly[1] more accurate estimates compared with that of the expert estimation (MRE of 0.07 vs MRE of 0.18). The following analysis shows that the result from the first analysis indeed is questionable and that other conclusions are better supported.

In the improved analysis we started with an exclusion of the outliers. We found only one obvious outlier. This project had a very high flexibility in product delivery and was removed the analysis. The estimate of that project was better described as a 'budget' and the goal was to a large extent to deliver as much functionality as possible within the budgeted resource usage. Eighteen projects remained. This is a small number for estimation error analysis, but not an unusual number of observations available for software companies to analyze.

Next, we adjusted for differences between estimated and actual functionality. We found two projects with more than a 10% difference between estimated and actual functionality. We assumed that an X% increase in functionality was connected with an X% increase in effort and adjusted for this increase in the actual effort. This assumption of a linear relationship between functionality is only an approximation of the true relationship. As shown in [4], a statistically significant non-linear relationship between effort and size is not strongly supported by existing empirical data.

Then, we grouped and analyzed the differences with regard to time of estimation, project priority, and experience with similar projects. Results from the grouping are displayed in Table 1 and Table 2.

Table 1. Grouped Analysis, Estimation Model

Factor	Value	#	Mean MRE	mean RE
Time of estimation	Early Estimate	1	0.08	-0.02
	Planning Phase	5	0.07	0.08
Project priority	Cost	4	0.09	0.03
	Time or Quality	2	0.05	-0.02
Experience from similar projects	Yes	6	0.07	0.00
	No	0	-	-

Table 2. Grouped Analysis, Expert Estimates

Factor	Value	#	Mean MRE	mean RE
Time of estimation	Early Estimate	6	0.15	0.05
	Planning Phase	6	0.14	0.10
Project priority	Cost	1	0.08	0.08
	Time or Quality	11	0.15	0.07
Experience from similar projects	Yes	7	0.10	0.01
	No	5	0.20	0.16

Then, we identified differences in frequency of use of the model and the expert judgment-based estimation methods, i.e., we compared the values in the # columns in Table 1 and Table 2, and found that expert estimates had a larger proportion of their estimates in the early phase, when the focus was on time or quality, and, when the estimators had no previous experience from similar projects. All these situations are connected with higher estimation complexity and higher estimation error, as can be seen from the mean MRE and RE-values in the tables. We, consequently, started to suspect that the higher estimation error of expert estimates could be explained by a more frequent use in higher estimation complexity situations and not by lower estimation performance of the method.

This alternative explanation would be supported if the difference in estimation error disappeared if we compared projects with similar estimation complexity characteristics only. Optimally, we should have compared for each combination of factor values, e.g., for the combination of "early estimate", "cost priority" and "experience from similar projects". The low number of data points did not enable such analyses. Instead, we had to rely on a mix of individual analysis of each factor and informal assessment of the contribution of the other factors. Table 1 and Table 2 display data for the purpose of the comparison. Notice that several of the mean values are based on quite few observations and should be interpreted especially carefully.

Table 1 and Table 2 suggest, as we understand the data, that there is at least one estimation complexity factor that may have biased the original, straight-forward, estimation error analysis: Experience from similar projects. Experience from similar projects is, not surprisingly, strongly connected with decrease of estimation error compared to the situation where the estimator has no previous experience from similar projects (MRE of 0.09 vs 0.20). Interestingly, the estimation model was never used when the estimator lacked relevant experience.

We compared projects where the estimator had experience from similar projects and found that the mean MRE and RE-values were similar for the model and the expert judgment-based estimates (mean MRE of 0.07 vs 0.10 and mean RE of 0.00 vs 0.01) for those projects. This suggests that the reason for the improved performance of the estimation model was that it was not used in situations with high estimation complexity! The "convincing" story from the straight-forward analysis is therefore not at all convincing when applying a proper error analysis process. A proper conclusion from the estimation error analysis, in our opinion, is that there is no evidence that support a difference in performance between the two estimation methods.

[1] Notice that statistical significance analysis of difference in mean values requires random allocation of treatment. It is therefore not a proper analysis for this and most other real-world software industry data set analyses.

An additional benefit of our analysis is that it provides information about the infrequent use of the estimation model certain situations, e.g., when no experience from similar projects is available. This information suggests that even if the model-based estimates were more accurate, they may not be able to replace the expert judgment-based estimates in all situations.

3. COST ESTIMATION STUDIES

In a previous review of estimation terminology [5], we showed that consistent and well-defined terminology is rarely applied in software cost estimation studies. As discussed in Section 2, inconsistent use of terminology is one factor that can lead to flawed conclusions in estimation error analysis unless the studied factor is isolated. In this section we investigate software cost estimation research papers to determine whether isolation strategies are commonly used in analysis of estimation error.

Selection of reviewed material:

Cost estimation studies were selected from the BESTweb library[2], which is an online library of estimation papers. Papers were selected by reading all the abstracts in BESTweb, and then the full versions of the papers that appeared to meet the following seven inclusion criteria. The paper:

- deals with estimates of software development effort, schedule, budget or cost or with project success/failure/performance,
- analyzes empirical collected estimates from real projects,
- analyzes estimates made before completion of the project,
- analyzes estimation accuracy,
- analyzes cross-company estimation performance,
- is the most recent paper by the main author that met the above criteria, and,
- was reviewed in the terminology review paper [5].

Eight studies met these criteria (see Table 3). The inclusion criteria are based on a combination of practical concerns and what we believe are characteristics of situations where estimation error analyses are most complex.

The small selection of reviewed papers is a threat to the generality of the results. It is outside the scope of this paper to perform a comprehensive review of the estimation literature. Our goal is to give an indication of the state-of-practice for isolation strategies applied in estimation error analysis. We believe the selection is sufficient for this purpose. We are mainly reviewing studies published in journals. It is, for that reason, possible that our selection is biased in the direction of describing a too positive state-of-the-practice in estimation error analysis.

Assessment and data extraction:
We extracted the following information from the studies

- I1: Factor(s) investigated by estimation error analysis.
- I2: Isolation strategies used to isolate the effect of the investigated factor(s).

Possible estimation error analysis factors (I1) were 'estimation ability', 'estimation complexity' and 'measurement process'. Possible isolation strategies (I2) were 'randomization', 'exclusion', 'grouping', 'adjustments' and 'none'. These factors are described in more detail in Section 4. Multiple factors are possible for each study.

For each study the information were extracted by both authors independently of each other. Disagreements where discussed until we reached agreement.

Results:

Table 3 Results

Study	Factor(s) investigated	Isolation strategies
Barki et. al. [6]	Estimation complexity	None
Bergeron and St-Arnaud [7]	Estimation ability Estimation complexity	Exclusion, grouping,
Heemstra and Kusters [8]	Estimation ability	Grouping.
Jørgensen [9]	Estimation ability	Randomization, exclusion, grouping
Lederer and Prasad [10]	Estimation complexity	None
Moløkken and Jørgensen [11]	Estimation ability Estimation complexity	None
Ropponen and Lyytinen [12]	Estimation complexity	None
Subramanian and Breslawski [13]	Estimation ability	None

Table 3 shows the results of the review. Five out of eight studies used no isolation strategy when analyzing estimation error. Among the studies that did use an isolation strategy, grouping is most popular (three studies) followed by exclusion (two studies) and randomization (one study). None of the studies investigated "measurement process", and none of the studies used "adjustments" as an isolation strategy. All the studies that attempt to isolate the investigated factors investigate factors of type "estimation ability".

The results indicate that studies often ignore the potential impact of non-studied factors, e.g., how systematic differences in estimation complexity or differences in the measurement process can disturb an analysis of the estimation ability. Many factors may impact the estimation error, and systematic biases are common. Expert estimation may, for example, frequently be preferred to formal models when data from similar projects is missing. Therefore, as demonstrated in Section 2, the lack of

[2] Available at www.simula.no/BESTweb

proper isolation strategies in the estimation error analysis may be a threat to the validity of the results.

This review does not investigate to which degree it was required to use an isolation strategy to analyze the estimation error factors. It may, for example, be the case that for some analyses, the impact of non-studied factors is ignorable. We believe, however, that in most cases there is a need for isolation of the investigated factor when performing estimation error analysis. An important reason for this belief is that we do not have a deep understanding of when factors do and do not impact the estimation error.

There are large variations in the ways the studies applies isolation strategies. While Bergeron and St-Arnaud [7] excludes projects with less than 150 days (actual effort), Jørgensen [9] limits his analysis to studies of projects with more than 10 hours (estimated effort), and duration of less than approximately eight calendar months. Similarly, Bergeron and St-Arnaud [7] group projects by time of estimation (project phase), Heemstra and Kusters [8] by size, while Jørgensen [9] groups by more variables: Type of task, project complexity, "know-how" skills of the estimators, estimation skills of estimators, whether or not the estimator estimated own work, type of payment, project importance for the customer and organizational role of the estimator.

It is hard to determine whether the studied factor is properly isolated in the three studies that use isolation strategies. It is clear that size of projects and time of estimation impact estimation error, but so does many other factors not addressed in these studies. We believe that this review suggest that there is a potential for improved estimation error analysis also in studies that do use isolation strategies in the estimation error analysis.

In this review, we evaluate our own work and may easily be biased. Ideally, this part of the review should have been performed by researchers independent of our research group. This was not done for practical purposes. We recommend the readers to examine the papers and make their own opinions.

4. A FRAMEWORK FOR ANALYSIS OF ESTIMATION ERROR

The proposed estimation error analysis framework has the following steps:

1. Decide on the factor to be analyzed

2. Define terminology and measures

3. Decide the strategy for isolation

4. Measure estimation error and collect other information necessary for isolation of the factor to be analyzed

5. Analyze and interpret the measured estimation error

Several of the steps are similar to steps included in other measurement frameworks, e.g., the GQM (Goal-Question-Metrics) framework, and textbooks on the improvement of software development processes. Our contribution to the improved analysis of estimation error is the tailoring of the general steps in other measurement frameworks to error analysis in the context of software estimation. A search of the literature, has not uncovered a framework for the analysis of software estimation error measurement or an applicable framework for the analysis of estimation error from another domain. We found examples of particular causal models of estimation error, e.g., [14, 15], but these were not, in our opinion, practical frameworks for error analysis in the context of the software industry. We believe that our framework will be particularly useful when the data set is small, e.g., less than 30 observations, and study design involving random allocation to treatment is not possible. These situations, we believe, are typical for analyses of software estimation error.

The proposed framework for the analysis of estimation error requires the measurement of a set of projects and is not meant for the analysis of reasons for estimation error of a single project. For that type of analyses, other types of analysis framework should be applied, e.g., root-cause analysis , post-mortem review [16], or "measured mile" analysis . Even in those cases, however, several parts of our framework may be useful, e.g., as checklists for possible estimation error factors.

The following description of our framework for the analysis of estimation error is not extensive for all steps, e.g., we have not described fully how to ensure proper data collection and statistical analysis. These topics are well covered in standard textbooks on the improvement software development processes and, therefore, are not covered here. A good estimation error analysis is supported by our framework, but the framework does not replace good measurement and analysis skill and experience.

4.1 Decide on Factor to be Analyzed

The first step in every estimation error measurement and process of analysis should be to get a precise understanding of the factor we want to know more about, i.e., the goal of the study. Ironically, an important step towards a better analysis of the factor that we want to study is to identify those factors that have an effect on the estimation error that we do not want to study. In particular, it is important to identify those factors that we do not want to study (often referred to as “nuisance” factors) that may bias the estimation error analysis. To support this process of factor identification we provide a comprehensive list of factors that can have a major impact on the measured estimation error. The top level categories of factors are these:

1) Estimation ability factors

2) Estimation complexity factors

3) Measurement process factors

4.1.1 Estimation Ability Factors

Important estimation ability factors include the following:

• Expert judgment skills. The ability to estimate the development effort of a software project applying judgment-based estimation methods.

• Accuracy of an estimation model. The ability of a formal estimation model to estimate the development effort accurately. Kitchenham and Linkman summarize different types of sources for estimation model error in [17]: measurement error, model error, assumption error, and, scope error.

• Skills in selection of estimation method. The ability to select the most suitable estimation method.

• Skills in the use of a formal estimation model. The ability to apply formal effort estimation models properly. It may sometimes be useful to separate this ability from the accuracy of an estimation model if the estimation method cannot be assumed to be applied as prescribed.

4.1.2 Estimation Complexity Factors

Some projects are more difficult to estimate than others and the reason for higher estimation error may be a higher estimation complexity. Factors related to estimation complexity include:

• Project management (cost control) ability. The project manager's ability to control the cost, i.e., to manage the project to the budget, is frequently a prerequisite for accurate effort estimates.

• Project member skill. It is usually easier to estimate the effort of skilled developers.

• Client and sub-contractor performance. The performance of a software project depends on the skills of the clients and sub-contractors.

• Completeness and certainty of information. If the input to an estimation process or formal estimation model (measurement error of input variable) is poor, we cannot expect accurate effort estimates.

• Inherent project execution complexity. Innovative projects, e.g., utilizing "leading edge" technology, and projects developing complex functionality, are inherently more difficult to estimate than repeating or simple projects. Another example of inherent project complexity is size (large projects are more difficult to estimate).

• Project priorities. Projects with a strong focus on time-to-market, for example, typically have less accurate estimates than those with a focus on cost control.

• Flexibility in product and process execution. If the project has a flexible scope, a simplification of the product can compensate for initially poor estimates and thus reduce estimation complexity and risk.

The above factors is related to the so-called "cost factors" listed in, e.g., [18, 19]. The main difference between cost and estimation complexity factors is the difference between factors with an impact on cost and with an impact on how difficult it is to estimate. In spite of this difference, it may be useful to examine lists of cost factors to identify sub-factors to the categories above. Factors with an impact on cost, frequently also have an impact on estimation complexity.

4.1.3 Measurement Process Factors

This category covers factors that affect the measured estimation error related to the quality of the measurement process itself:

• Inconsistent use of terminology: When there is a lack of clear definitions of terms and there exist differences in interpretations of important estimation terminology, variance in estimation error cannot automatically be attributed to variance in estimation ability or estimation complexity.

• Logging problems: Lack of proper logging routines for the actual use of effort may result in there being differences in activities included in the measured actual effort, or may affect whether overtime is recorded or not.

• Difference between planned and actual output/process: Software projects may experience increases or reductions in functionality. Similarly, the project may not conduct all planned quality assurance activities or deliver the planned quality. Differences in estimation error may be caused by these differences between planned and actual output/process and not, for example, estimation ability.

• Measurement selection bias: Three particularly important selection biases are: 1) Exclusion of cancelled projects. This typically leads to too positive a view of the estimation ability. 2) Exclusion of projects that have been estimated, but never been started. This may easily lead to too negative a view of the estimation ability, e.g., it is more likely to win a bidding round with an over-optimistic estimate compared with a realistic estimate. 3) Inclusion of only the projects that confirm the desired output of the analysis, i.e., a "confirmation bias". We discuss the effect of measurement selection biases in [20].

Our categorization of estimation error factors should serve as a starting point for software organizations' discussions on which factors to study, and as a checklist for the need to understand, and control the effect of, the nuisance factors.

4.2 Define Terminology and Measures

Most estimation error surveys and software engineering textbooks do not provide a clear definition of what they mean by an effort estimate . In addition, we, in our role as estimation advisors, have observed an unfortunate mix of interpreting an effort estimate as "most likely effort", "planned effort", "best case effort" and "effort used as input to price-to-win" among software professionals. If the data collection does not ensure consistent use of 'effort estimate", it is extremely difficult to perform meaningful analyses.

Two common cost estimation error measures are:

Magnitude of Relative Error (MRE) = |Actual Effort – Estimated Effort| / Actual Effort

Relative Error (RE) = (Actual Effort – Estimated Effort) / Actual Effort

Both measures have been criticized and are far from perfect measures of cost estimation errors. Frequently, it helps to use more than one measure, more than one 'central value', e.g., display both the mean and the median estimation error, and include plots of all the estimation error data.

4.3 Decide the Strategy for Isolation

The following four strategies are candidates for isolation of the estimation error factor of interest:

• Randomization of treatment is the most powerful factor isolation strategy. For example, if we want to compare the estimation performance of two different estimation methods, this strategy would require that the choice of an estimation method be random. The strength of this isolation strategy is that it removes any biased contribution of the non-studied factors to estimation error. One problem with the strategy is that software companies may, for good reasons, not accept the random choice of an estimation method.

• Grouping (Blocking) of the projects into subsets similar with respect to non-studied factors may be necessary to ensure meaningful comparisons. For example, supposing that we want to study the impact of project cost control on the estimation error, we should compare projects with similar levels of estimation expertise and estimation complexity.

• Adjustment for the contribution of one or more non-studied factors to the estimation error may be necessary, if there is no randomization or if the effect of the non-studied factors is not removed through grouping. For example, if some projects delivered more functionality than initially planned, the actual effort values may be adjusted to reflect only the effort to implement the initially planned functionality. The adjustment strategy is only possible if we, to some extent, are able to assess the impact of the non-studied factors on the estimation error.

• Exclusion of observations from the analysis may be necessary when it is possible for the non-studied factors to have a large but unknown impact on the estimation error, or when an observation is not relevant for the goal of the estimation error analysis. Notice that exclusion of an observation does not mean that the observation is of no value for other analyses or purposes. The excluded observation is merely not relevant for the analysis of the factor to be studied.

Most of the time, a combination of isolation strategies, e.g., of grouping, adjustment and exclusion, is needed. If isolation of the factor of interest is not possible, the measurement and the measurement analysis may not be worthwhile.

4.4 Measure Estimation Error and Collect Other Information Necessary for Isolation of the Factor to Be Studied

The information to be collected depends on the factor to be studied (Step 1), the terminology for and measures of estimation error (Step 2), and the chosen isolation strategies (Step 3). In particular, the isolation strategies may have a large impact on the information required for proper analysis:

Randomization: Guidelines for randomization are included in standard statistics textbooks. As opposed to the other isolation strategies, it is not necessary that we collect much context information, as long as the power of the study is high and the treatment is randomized.

Grouping: Proper use of this strategy requires the collection of information about non-studied grouping factors with a large, potentially biased, impact on the estimation error analysis.

Adjustments: Potentially, there are a large number of non-studied factors that could have a substantial impact on estimation error. If we were to group for all these factors, the number of observations in each group would be very small. To avoid this, we may combine grouping with an adjustment strategy. An adjustment strategy requires the identification of the important non-grouped factors and knowledge about their impact on the use of effort.

Exclusion: Projects should be removed from the measurement-based analysis if the impact of non-studied, potentially biasing factors cannot be removed through randomization, grouping or adjustments. Excluded projects may still be subject to analysis, but then other types of analysis frameworks are needed, e.g., root cause analysis or post-mortem review frameworks.

4.5 Analyze and Interpret the Measured Estimation Error

The estimation error analysis to be conducted depends on the measurement goals, the factor to be studied and the isolation strategy chosen. It does not lie within the scope of our framework to provide detailed guidelines for this. Several analysis techniques are described in standard textbooks on statistics and software process improvement.

5. Summary

A real-world example where common analysis of estimation error lead to a flawed conclusion, together with a review of published estimation error analyses in research studies, suggest that there is a need for better analyses of software cost estimation error. Particularly, there may be a need for clarification of the goal of the estimation error analysis and how the impact of the factors of interest on estimation error should be isolated. Currently, we frequently measure cost estimation error with improper means to understand what we measure. We present a framework that can be, we believe, a useful tool to improve the estimation error analysis. In particular, we believe, the checklist to identify non-studied factors with a potential biasing impact on the measured estimation error, the emphasis on proper estimation terminology, and the support on isolation strategies are useful. The framework does, however, not replace good analysis skill.

Acknowledgement: We thank Professor Barbara Kitchenham for critical and useful comments on an early draft of this paper.

References

1. Boehm, B.W., *Software engineering economics*. 1981, New Jersey: Prentice-Hall. XXVII, 767 s.
2. Putnarn, L.H., D.T. Putnam, and W. Myers, *Adapting project estimation to advancing technologies.* American Programmer, 1996. **9**(6): p. 23-29.
3. Jones, C.T., *Estimating software costs*. 1998, USA: McGraw-Hill. XIX, 724 s. : ill.
4. Dolado, J.J., *On the problem of the software cost function.* Information and Software Technology, 2001. **43**(1): p. 61-72.
5. Grimstad, S., M. Jørgensen, and K. Moløkken-Østvold, *Software Estimation Terminology - The Tower of Babel.* Accepted for publication in Information and Software Technology, 2005.
6. Barki, H., S. Rivard, and J. Talbot, *An integrative contingency model of software project risk management.* Journal of Management Information Systems, 2001. **17**(4): p. 37-69.
7. Bergeron, F. and J.Y. St-Arnaud, *Estimation of information systems development efforts: a pilot study.* Information and Management, 1992. **22**(4): p. 239-254.
8. Heemstra, F.J. and R.J. Kusters, *Function point analysis: Evaluation of a software cost estimation model.* European Journal of Information Systems, 1991. **1**(4): p. 223-237.
9. Jørgensen, M., *Realism in assessment of effort estimation uncertainty: It matters how you ask.* IEEE Transactions on Software Engineering, 2004. **30**(4): p. 209-217.
10. Lederer, A.L. and J. Prasad, *Causes of inaccurate software development cost estimates.* Journal of Systems and Software, 1995. **31**(2): p. 125-134.
11. Moløkken, K. and M. Jørgensen. *A review of software surveys on software effort estimation*. in *International*

Symposium on Empirical Software Engineering. 2003. Rome, Italy: Simula Res. Lab. Lysaker Norway.

12. Ropponen, J. and K. Lyytinen, *Can software risk management improve system development: an exploratory study.* European Journal of Information Systems, 1997. **6**(1): p. 41-50.
13. Subramanian, G.H. and S. Breslawski, *An empirical analysis of software effort estimate alterations.* Journal of Systems and Software, 1995. **31**(2): p. 135-141.
14. Lederer, A.L. and J. Prasad, *A causal model for software cost estimating error.* IEEE Transactions on Software Engineering, 1998. **24**(2): p. 137-148.
15. Abdel-Hamid, T.K. and S.E. Madnick. *A model of software project management dynamics*. in *COMPSAC 82*. 1982. Chicago, IL, USA: IEEE, New York, USA.
16. Birk, A., T. Dingsøyr, and T. Stalhane, *Postmortem: Never leave a project without it.* IEEE Software, 2002. **19**(3): p. 43-45.
17. Kitchenham, B. and S. Linkman, *Estimates, uncertainty, and risk.* IEEE Software, 1997. **14**(3): p. 69-74.
18. Symons, C.R., *Software sizing and estimating - Mk II FPA*. 1991, Chichester, U.K.: John Wiley&Sons.
19. Boehm, B.W., *Software engineering economics.* IEEE Transactions on Software Engineering, 1984. **10**(1): p. 4-21.
20. Jørgensen, M. and S. Grimstad. *Over-optimism in Software Development Projects: "The winner's curse"*. in *CONIELECOMP*. 2005. Puebla, Mexico.