The Influence of Selection Bias on Effort Overruns in Software Development Projects

Magne Jørgensen

Simula Research Laboratory & Institute of Informatics, University of Oslo

Norway

magnej@simula.no

Abstract

Context: *A potentially important, but neglected, reason for effort overruns in software projects is related to selection bias. Selection bias–induced effort overruns occur when proposals are more likely to be accepted and lead to actual projects when based on effort estimates that are too low rather than on realistic estimates or estimates that are too high. The effect of this bias may be particularly important in bidding rounds, but is potentially relevant in all situations where there is effort or cost-based selection between alternatives.* **Objective:** *To better understand the relevance and management of selection bias effects in software development contexts.* **Method:** *First, we present a statistical model illustrating the relation between selection bias in bidding and other contexts and effort overruns. Then, we examine this relation in an experiment with software professionals who estimated and completed a set of development tasks and examine relevant field study evidence. Finally, we use a selection bias scenario to assess awareness of the effect of selection bias among software providers.* **Results**: *The results from the statistical model and the experiment demonstrated that selection bias is capable of explaining much of the effort overruns. The field evidence was also consistent with a substantial effect of selection bias on effort overruns, although there are alternative explanations for the findings. We found a low awareness of selection bias among the software providers.* **Conclusion:** *Selection bias is likely to be an important source of effort overruns and should be addressed to reduce problems related to over-optimistic effort estimates.*

Keywords: Effort estimation, effort overrun, selection bias, winner's curse

1. Introduction

Software providers have, in general, a bad reputation in relation to the provision of accurate estimates of the required effort and cost of completing software projects. A particular problem is that the effort estimates of software projects have a strong tendency toward being too low [1, 2]. The consequences of effort estimates that are too low include delayed deliveries, financial losses, low quality of the deliverables, dissatisfied customers, and frustrated developers. There have been numerous studies trying to understand the causes of biased and inaccurate estimates and to propose better estimation processes to improve the situation (see, for example, [3, 4]). In this article, we examine a possible cause of effort overrun that has not been much discussed in software development contexts and requires improvement actions that are to some extent different from those proposed in earlier estimation research. We denote this cause “selection bias.” Selection bias occurs in software development contexts when the clients’ processes for the selection of providers or between investment alternatives lead to an over-representation of proposals based on effort estimates that are too low. When there is a selection bias, we may have a situation where the total set of proposals are based on unbiased or pessimistic effort estimates, while the selected proposals are based on estimates biased toward under-estimation of the effort. Clearly, only the estimates leading to actual projects will be evaluated with respect to accuracy and bias. This means that the previously reported strong tendency towards effort overruns in software projects potentially may have been caused by how project proposals are selected rather than, for example, by poor estimation processes or a disposition towards over-optimism among software professionals. To what extent this is the case in software development contexts and how to deal with this type of selection bias effect are the main topics of this paper.

The selection bias effect may be illustrated by considering a client who selects between project proposals from several software providers. Price is typically one of the criteria used for this selection. Providers will vary in how much effort they think it will take to complete the project, how over-optimistic their effort estimates are, and how they price the project. It is likely that providers who under-estimate the required effort, on average, will submit proposals with a lower price than those who over-estimate the required effort. A client who emphasizes low price will, consequently, tend to select among the providers who under-estimated the effort. To illustrate the same selection bias effect from a provider's perspective, consider a situation where he/she estimates that a project will require about 1,000 work hours. The provider knows that his/her estimated effort may be inaccurate but has no information available to suggest that the project is more likely to require more than it is to require less than 1,000 work hours to complete the work. Assume that the provider learns that several other competent providers have estimated the same project and that all of them have estimated it to take much more effort than 1,000 work hours. Should the provider update the estimate of expected use of effort based on this knowledge? Perhaps not surprisingly, the correct answer may be yes, especially when the uncertainty in the actual use of effort is high, and there is no reason to believe that the provider has substantial advantages in terms of efficiency compared with the other providers. As can be seen from this illustration, there is a difference in the expected use of effort before and after receiving the knowledge about the other estimates. Now, assume that the provider does *not* know about the estimates of the other providers, but knows that there are several other providers estimating the same project and that the client will select among the providers who submit the lowest estimates. This is very much the same situation as the previous one. Unless the provider updates the estimate based on the information about the selection process, despite not knowing the other estimates, it will experience being selected much more frequently when its estimate is over-optimistic. If the selection bias is strong, the provider may experience that he/she is mainly selected when under-estimating the effort to an extent that makes financial losses and/or substantial reduction of quality of deliveries unavoidable. This negative effect of selection bias associated with over-optimistic estimates is frequently discussed under the heading of "the winner's curse" (see, for example, [5, 6]). The illustrations tells us that at least two things: i) A tendency toward over-optimistic effort estimates of executed projects does not require over-optimistic software providers, ii) We cannot expect to solve all challenges related to effort overruns through improved estimation processes. In addition, we need to address the project proposal selection and evaluation processes.

This paper focuses on selection between providers for the same project, such as in bidding rounds, where a low price is an essential selection criterion. However, the selection bias effect is not restricted to selection between providers. The same selection bias effect is, amongst others, present in situations where there is only one provider and a range of investment alternatives. This includes:

- Selection of functionality to be included in the next release of a software system
- Decisions on whether a new project should be started or not
- Selection between different alternatives regarding technology, architecture, development tools, and off-the-shelf-software to be used in a project

In the above cases, a decision in favor of including a specified functionality, starting a new project, using a particular technology, etc., are all more likely when the cost connected with its implementation has been over-optimistically estimated. Notice that this, and other argumentation in this paper, is based on the underlying assumption that lower prices and bids, *on average*, are connected with lower effort estimates. We believe that this is a reasonable assumption in most software development contexts. Notice also, that the selection bias effect does not distinguish between different sources of inaccurate estimates, e.g., whether under-estimating the effort or cost is due to over-optimistic judgment or "strategic misrepresentation" [7]. The use of a low effort or price as a selection criterion may, consequently, not only increase the likelihood of selecting a proposal where the selected provider has been over-optimistic, but also of selecting a provider who has strategically misrepresented, e.g., been dishonest about, what is the most realistic effort or cost.

The remaining part of this paper is organized as follows: In Section 2, we model the statistical mechanisms leading to the selection bias effect on effort overruns. We use this model to assess to what degree selection bias is able to explain the observed tendency towards too low effort estimates and as an input to the discussion of how selection bias effects may be managed. Section 3 describes the results of an empirical study with software professionals. This study aims at demonstrating the size of the selection bias–induced effort overrun in a field setting. Section 4 examines to what extent published empirical evidence is consistent with a substantial effect of the selection bias on effort overruns and to what extent software professionals are aware of this bias. Section 5 discusses possible ways of managing and avoiding selection bias–induced effort overruns. Section 6 presents the conclusion.

2. An Illustrative Selection Bias Model

Selection bias may be seen as a statistical phenomenon with practical consequences. To better understand and illustrate the main mechanisms of selection bias and its capability to explain a tendency towards observing effort overrun, we present an illustrative statistical model of the phenomenon. The model is based on the following assumptions:

- The estimated effort (*est*) equals the actual effort (*act*) plus an estimation error (*err*), i.e., *est* = *act* + *err*. Different providers may have different actual effort, different estimated effort, and different estimation error values for the same project.
- The variables *act* and *err* are independent. This assumption is not likely to be true in contexts with large differences in project sizes, but may sometimes be acceptable in situations where many companies bid for the same projects or for projects of similar sizes. For larger differences in project sizes, we may use a log-normal model, i.e., we may use ln(*est*) and ln(*act*) instead of *est* and *act* (see, for example, the log-normal models in [8]). Then, the error term becomes multiplicative, and the assumption is that there is independence between the relative estimation error (*est/act*) and the actual effort (*act*).
- The variables *act* and *est* are normally distributed, have mean values of μ_{act} and μ_{est}, and standard deviations of σ_{act} and σ_{est}, respectively. Models based on log-normal distributions may, as noted earlier, be more reasonable in situations with large differences in project sizes. The normal distribution assumption may, however, be acceptable in other software development situations with less variance in project sizes.
- The total set of estimates is unbiased, i.e., the mean estimation error is 0. With a mean estimation error of 0, the mean values of the estimated and the actual efforts are the same, i.e., $\mu_{act} = \mu_{est} = \mu$. This assumption implies that our illustrative model focuses on observed estimation bias in a situation where the total set of all estimates is neither biased toward too low nor too high values, i.e., when there are no underlying estimation biases among the providers. When using log-normal distributions, an unbiased situation implies that that the geometric rather than the arithmetic means should be the same for the actual and the estimated effort.
- The estimation accuracy ($\rho_{est,act}$) is measured as the correlation between the estimated and the actual effort.

With additional assumptions, the above model can be extended to include bidding situations, e.g., by including the, somewhat simplistic, assumption that a bid is the estimated cost with a profit margin added. The model may also be extended to include situations where the means of the estimated and the actual effort are not the same, i.e., situations with underlying estimation biases in the total set of estimates. For our purpose, i.e., to illustrate the selection bias mechanisms leading to an observed bias toward effort overrun, these additions are not essential and would not lead to different results.

To find the expected actual effort for a given estimated effort, we may regress *est* (the independent variable) on *act* (the dependent variable), i.e., we examine the regression model:

(1) $act = \alpha + \beta\ est.$

Regressing *est* on *act* gives us a linear model that minimizes the least square of the difference between the actual effort and the *act*-value predicted by the model.

We know that (see, for example, [9, Section 13]:

(2) $\beta = \rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}$ and $\alpha = \mu_{act} - \beta\mu_{est} = (1-\beta)\mu = \left(1 - \rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}\right)\mu.$

Inserting the equations of (2) into (1) shows that the expected actual effort for a given estimated effort is:

(3) $act\ = (1 - \rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}})\mu + \rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}est.$

Assume, for example, a set of providers estimating the same project in a development context where the correlation between the estimated and the actual effort ($\rho_{est,act}$) is 0.8, the standard deviations of the actual and the estimated effort are the same (which implies that $\frac{\sigma_{act}}{\sigma_{est}} = 1$), and the mean estimated effort of the providers (μ) is 1,000 work hours. As before, we assume that the providers are on average unbiased, i.e., that the mean actual effort equals the mean estimated effort. Furthermore, assume that the estimated effort (*est*) by one of the providers is 500 work hours, i.e., the provider estimates the effort to be much lower than the average estimate of the providers. The expected actual effort of that provider, given the above information, is then (1-0.8*1)*1,000 + 0.8*1*500 = 680 work hours, i.e., there is an expected effort overrun of 680 - 500 = 180 work hours.

Notice that the expression found in (3) is the same as that which would have been proposed if we had used regression-toward-the-mean [10] or true score-based theories [11]. More on application of regression-toward-the-mean theories in software effort estimation contexts can be found in [12].

By using (3), the difference between the expected actual and the estimated effort can be expressed as:

(4) $act - est = \left(1 - \rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}\right)\mu + \rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}est - est = \left(1 - \rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}\right)(\mu - est).$

Assume that the selected proposal is based on the estimate *est* = $w \cdot \mu$, where *w* may be interpreted as a measure of the client's focus on low price or effort. If, for example, *w* = 0.75, a provider has selected a proposal which has been based on an effort estimate 75% of the mean estimate. Clearly, the *w*-value cannot be derived directly from a provider's formulated selection strategy, e.g., that he/she places 60% weight on the price and 40% weight on the quality. Indirectly, however, a client's selection strategy, together with other factors such as number of proposals to chose between, imply a certain *w*-value for a particular project. The lower the *w*-value, the more likely it will be to experience a selection bias-induced effort overrun.

If *est* is replaced with $w \cdot \mu$, the expected degree of effort overrun connected with the selected proposal is:

(5) $act - est = \left(1 - \rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}\right)(\mu - w \cdot \mu)) = \mu\left(1 - \rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}\right)(1 - w).$

A measure of the relative estimation bias (*rel*) may be expressed as:

(6) $rel = \frac{act - est}{\mu} = \frac{\mu\left(1 - \rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}\right)(1-w)}{\mu} = \left(1 - \rho_{est,act}\frac{\sigma_{act}}{\sigma_{est}}\right)(1 - w).$

Table 1 displays the *rel* for selected levels of estimation accuracy ($\rho_{est,act}$) and the price focus (w) for the situation where the ratio $\frac{\sigma_{act}}{\sigma_{est}} = 1$, i.e., where the actual effort varies as much as the estimated effort.

Table 1: Expected relative error

$\frac{\sigma_{act}}{\sigma_{est}} = 1$	$\rho_{est,act}$=1.0	$\rho_{est,act}$ =0.9	$\rho_{est,act}$ =0.8	$\rho_{est,act}$=0.7	ρ_{estact}=0.6
w=1.0	0	0	0	0	0
w=0.75	0	0.03	0.05	0.08	0.10
w=0.5	0	0.05	0.10	0.15	0.20
w=0.25	0	0.08	0.15	0.23	0.30

Assume, for example, a situation where $\rho_{est,act}$ = 0.7 (medium estimation accuracy) and w = 0.5 (strong price focus). These values give *rel* = 0.15, which suggest an expected observed effort overrun of 15% due to selection bias in situations where there is no underlying estimation bias, i.e., when evaluating the total set of effort estimates. Table 1 illustrates that the degree of observed effort overruns increases substantially with increased bias toward selecting proposals with lower than average estimates (decreased w) and with decreased estimation accuracy (decreased $\rho_{est,act}$).

An examination of a selection of software development data sets suggests that the included correlation, price focus, and standard deviation assumption of Table 1 may reflect some, although not all, field situations. For example, the software development data sets in [13-16] give correlations between actual and estimated effort in the range of 0.62 to 0.95. Studies on bidding rounds sometimes report a large difference in bids for the same project. For example, [17] reported that the lowest bid was only 30% of the mean bid. There is, as mentioned earlier, a difference between bids and effort estimates, but we think it is reasonable to assume that a large variance in bids, to some extent, reflects a large variance in effort estimates. The assumption of similar variance of estimated and actual effort ($\frac{\sigma_{act}}{\sigma_{est}}$ close to 1.0) is supported by the data set in [14] but not by, for example, that in [13], which represents a ratio of about 1.2. We will illustrate the model with the use of a standard deviation ratio of 1.2 later in this section.

Notice that the above data sets are from in-house development, where the selection bias is likely to be low, due to less competitive environment. The reason for this is that it may be misleading to use standard deviation and accuracy data from situations already exposed to selection bias for the purpose of illustrating selection biases in our model.

If we assume a "shrinkage" of the variance of the estimates compared with that of the actual efforts, i.e., if we observe $\frac{\sigma_{act}}{\sigma_{est}} > 1$, we will, in accordance with (6), observe a reduction of the expected bias toward over-optimistic effort estimates. Shrinkage of the variance of the estimates would, for example, be present if the software providers knew about the selection bias effect and tended to provide estimates close to the typical mean effort of similar projects to compensate for it. Table 2 illustrates the expected relative error for the shrinkage level $\frac{\sigma_{act}}{\sigma_{est}} = 1.2$, i.e., when the estimates' standard deviation shrinks by about 20% compared with that of the actual efforts, as found in the data set reported by [13].

Table 2: Expected relative error with estimation shrinkage

$\frac{\sigma_{act}}{\sigma_{est}} = 1.2$	$\rho_{est,act}$=1.0	$\rho_{est,act}$ =0.9	$\rho_{est,act}$ =0.8	$\rho_{est,act}$=0.7	$\rho_{est,act}$=0.6
w=1.0	0	0	0	0	0
w=0.75	-0.06	-0.03	0	0.03	0.06
w=0.5	-0.13	-0.06	0	0.06	0.13
w=0.25	-0.19	-0.09	0	0.09	0.19

As can be seen from Table 2, a situation where the estimates' standard deviation shrinks by 20% would over-compensate for selection bias in situations with high estimation accuracy and under-compensate in situations with low estimation accuracy. The optimal level of shrinkage for this situation, in terms of achieving unbiased estimates, is when $\rho_{est,act} \frac{\sigma_{act}}{\sigma_{est}} = 1$. This is the case for the estimation shrinkage level of 20% when the estimation accuracy corresponds to $\rho_{est,act} = 0.8$. This level of shrinkage would be obtained if we used the expected actual effort from equation (3) instead of the unadjusted estimated effort. Unfortunately, an adjustment based on (3) may in practice be difficult, since it requires the knowledge about the mean estimated effort of the other providers.

Assume a situation where the actual effort is about the same for all providers, but a high estimation uncertainty leads the providers to produce quite different estimates. Then, σ_{act} will be much lower than σ_{est} and $\frac{\sigma_{act}}{\sigma_{est}}$ will be close to 0, i.e., we will observe the opposite of estimation variance shrinkage. From (6), we can see that the expected relative error would then be close to (1 – *w*), i.e., the effect of the selection bias would in this case be very strong. Selecting, for example, an estimate 75% of the average estimate (*w* = 0.75), which may be considered as a rather moderate price focus, would, for example, lead to a selection bias–induced effort overrun of 25%. Consequently, our model illustrates that not only the estimation accuracy ($\rho_{est,act}$) and the client's price focus *w* determine the effect of the selection bias on the observed effort overrun. The more similar the expected actual efforts of the different providers, the more likely it is that a client will experience selection bias-induced effort overruns.

The main results from our illustrative selection bias model analysis may be summarized as:

- Situations with selection bias–induced estimation over-optimism include: i) the effort estimates are inaccurate, ii) the selected project proposals are based on effort estimates lower than the average estimate, *and* iii) there is insufficient estimation variance shrinkage to compensate for the estimation inaccuracy.
- Situations with little or no selection bias–induced estimation over-optimism include: i) the effort estimates are inaccurate, ii) the selected project proposals are based on effort estimates close to the average estimates, iii) the providers manage to adjust their estimates (corresponds to estimation variance shrinkage) to reflect the accuracy of the estimates, *or* iv) the selection of the provider is mainly based on competence or quality and not on price.

The above analysis and results would not be much different if we assumed that the estimated and the actual efforts were log-normally distributed, i.e., that ln(*act*) and ln(*est*) were normally distributed. Using the same assumptions for ln(*act*) and ln(*est*) as we had for *act* and *est* earlier, which implies for example, a multiplicative rather than additive error term and the same geometric rather than the same arithmetic mean of *act* and *est*, (using (6)) we have the following equation for the log-transformed variables:

$$(7)\ \ln(act) - \ln(est) = \ln\left(\frac{act}{est}\right) = \left(1 - \rho_{\ln(est),\ln(act)} \frac{\sigma_{\ln(act)}}{\sigma_{\ln(est)}}\right)(\mu_{ln} - \ln(est)),$$

where $\rho_{\ln(est),\ln(act)}$ is the correlation between ln(*est*) and ln(*act*), $\sigma_{\ln(act)}$ is the standard deviation of ln(*act*), $\sigma_{\ln(est)}$ is the standard deviation of ln(*est*), and μ_{ln} is the mean of ln(*est*) and ln(*act*), i.e., μ_{ln} = ln(*est*) = ln(*act*).

Back-transforming (7) to the raw scores yields that the relative estimation error ($\frac{act}{est}$) can be expressed as:

$$(8)\ \frac{act}{est} = e^{\left(1-\rho_{\ln(est),\ln(act)} \frac{\sigma_{\ln(act)}}{\sigma_{\ln(est)}}\right)(\mu_{ln} - \ln(est))}.$$

The conditions where we can expect to observe effort estimates that are too low, i.e., $\left(\frac{act}{est} > e^0 = 1\right)$, are similar to those identified earlier. If a log-transformed effort estimate does not equal the mean of the log-transformed effort estimates, we will observe a bias toward effort overrun when:

- there is an imperfect correlation between the log-transformed estimated and actual effort ($\rho_{\ln(est),\ln(act)} < 1$) and no estimation shrinkage ($\frac{\sigma_{\ln(act)}}{\sigma_{\ln(est)}} = 1$), *and*
- the estimation shrinkage is not sufficient to compensate for the correlation between the log-transformed estimated and the log-transformed actual effort, i.e., when ($\frac{\sigma_{\ln(est)}}{\sigma_{\ln(act)}} < \rho_{\ln(est),\ln(act)}$).

Notice that the effect of selection bias on effort overrun does not necessarily imply that clients act irrationally when selecting among the lower-priced proposals, i.e., among the proposals most likely to have under-estimated the effort. However, as described in [18, p. 30], the dilemma is: "*Choosing apparently better alternatives will, on average, produce higher returns. Thus, it is sensible to choose alternatives that are estimated to be relatively good. However, such a procedure is not surprise neutral. In the absence of behavioral adjustments, higher expected benefits will be associated with greater expected disappointments.*"

We will use the insight from the illustrative model in the following sections when examining empirical evidence (Sections 3–4) and when discussing how to avoid and/or manage the selection bias effect on effort overruns (Section 5).

3. An Empirical Study on Selection Bias

The study presented in this section empirically examines how much different degrees of lowest price focus affect the observed tendency toward observing under-estimation of effort in a field context.

3.1 Study Design

Twenty software professional software developers were recruited from different Norwegian companies. All participants had competence in the relevant technology, i.e., in UML, Struts, JSP, Java, the Eclipse IDE, and MySQL. The developers completed the same five development tasks on an existing system, but they did not know about each other and were asked to treat this as ordinary work. They were paid close to ordinary fees for their development work. The BESTWeb system (see www.simula.no/BESTweb), a system of about 3,000 lines of Java code,[1] was subject to the development work.

The main steps of the estimation and development work relevant to the analyses in this paper were as follows:

1. The developers received a task specification, starting with Task 1.

2. They estimated the most likely effort needed to complete the specified task.

3. They performed the task, which included designing, programming, testing, and documenting the task.

4. Upon completion of the task, it was submitted for acceptance testing. The system was tested using a predefined system acceptance test plan. If the test failed, the developer was told the problem and asked to fix it and to submit the solution again. When the test was passed, we recorded the effort used to

[1]Results on the effect of lessons-learned sessions on estimation accuracy from this study have previously been published in 19. Jørgensen, M. and T.M. Gruschke, *The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty Assessments.* IEEE Transactions on Software Engineering, 2009. **35**(3): p. 368-383. We found no substantial difference in the estimation accuracy between the developers exposed to lessons-learned sessions and the others and have joined the two groups for the selection-bias analysis purpose of this paper.

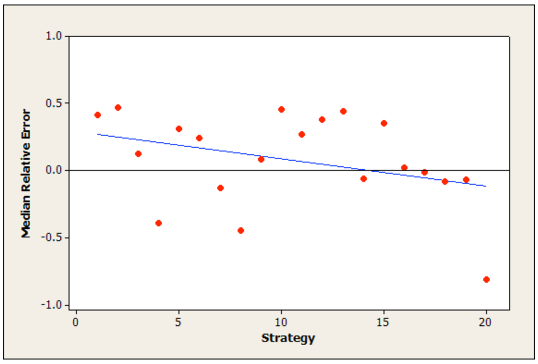
complete the task. Then, the developer received the next task, i.e., repeated the process in Step 1 with the new task, until all five tasks had been completed.

3.2 Results

The developers' total effort on the five tasks varied from 20 to 60 work hours, with a median of about 40 work hours. This relatively small difference in total work effort is likely to be a consequence of how we selected the developers, i.e., only senior, highly skilled developers were selected. The median value of the estimation bias, defined as the relative estimation error (RE) (RE = (actual effort–estimated effort)/actual effort), was 0.14, i.e., the median estimate was 14% too low. This suggests a weak tendency toward over-optimistic effort estimates in the total set of effort estimates. We used the median RE values as a measure of the average estimation bias because the mean relative estimation error was heavily influenced by a few very high estimation errors.

For the purpose of studying the effect of selection bias, we analyzed what would have been observed as the RE if we had selected only the developer with the *i*-th lowest estimate for each of the five tasks. This is meant to reflect different degrees of low estimated effort (low price) focus of a client when selecting a provider. The median observed RE for each *i* is displayed in Figure 1. The figure includes a regression line through the data points to illustrate the general trend toward higher effort overrun as a consequence of the increased focus on low estimates.

Figure 1: Effect of selection bias on observed effort overrun



As can be seen in Figure 1, there is a strong connection between the selection strategy (the focus on low estimated effort) and the expected effort overrun. Although the median effort overrun of the total set of the estimates was only 14%, we would have observed an estimation overrun of more than 40% if we had used a strategy where we selected the developers with the lowest or the second lowest estimates. As can be seen from Figure 1, we would typically observe unbiased or even pessimistic estimates if we had selected the developers with substantially higher than average estimates, e.g., those with ranks 16–20.

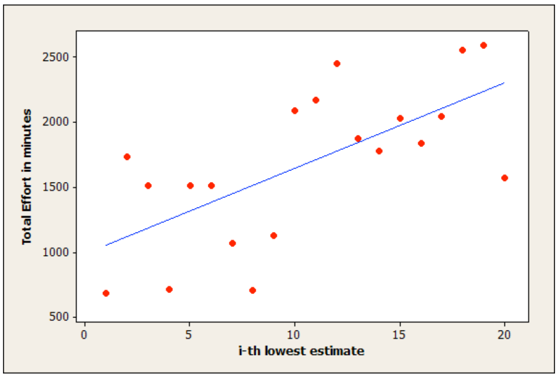
To examine the relevance of the model we presented in Section 2, assume that we selected the lowest estimate, i.e., use the strategy where i = 1, for each of the five tasks. For the data set as a whole, this would give w = 0.35, i.e., the selected estimate would, on average, be 35% of the median estimate. According to equation (6), applying the correlation between the estimated and the actual effort (which was 0.3) and the standard deviation values for the actual and the estimated effort (which were 263 and 124, respectively) of our data sets, we get:

$$rel = \left(1 - \rho_{est,\ act}\frac{\sigma_{act}}{\sigma_{est}}\right)(1-w) = \left(1 - 0.3\frac{263}{124}\right)(1-w) = 0.36 - 0.36w = 0.23.$$

The calculated *rel* value of 0.23 may be interpreted as the expected contribution of the selection bias to the observed estimation overrun, i.e., we would expect an effort overrun of 23% in situations with no underlying estimation bias due to the selection of the lowest estimate. The observed bias combines the underlying (RE = 14%) and the selection bias-induced estimation bias (*rel* = 23%). This means than the observed estimation bias of about 40% (see Table 1), when selecting the developer with the lowest estimate (w = 0.35), is what we would expect and supports the validity of our model.

While the above empirical results demonstrate that there may be a substantial increase in estimation overrun when selecting the developer with the lowest, or among the lowest, estimates, it does not necessarily imply that we should avoid selecting the developer with the lowest effort estimate. In fact, as illustrated in Figure 2, the strategy of selecting the developer with the lowest effort estimate for each task would have been a good strategy for keeping the total effort as low as possible. However, the degree to which a low effort estimate is a good indicator of competence is context dependent. Sometimes, a low estimate is an indicator of a lack of knowledge about the complexity of the project (see, for example, [17]).

Figure 2: Effect of selection bias on total effort



4. Field Evidence on Selection Bias Effects

The model-based argumentation in Section 2 and our results from the study in Section 3 document the potential effect of selection bias on effort overruns in software development. This section examines relevant evidence from various sources, with the aim of better understanding how likely it is that selection bias is not only capable of, but actually explains some, or perhaps most, of the bias toward effort overrun observed in software development contexts. Sections 4.1 and 4.2 compare effort and cost overruns in situations with an expected low degree of selection bias, i.e., in-house development and experimental contexts, with situations with a higher degree of expected selection bias, i.e., competitive bidding rounds. The motivation for this comparison is that if a selection bias explains parts of the observed tendency toward effort overrun, we would expect this tendency to decrease in situations with less or no selection bias. If we observe no such tendency, this would strongly weaken our belief in the importance of the selection bias effect to explain effort overruns. Section 4.3 examines the degree of awareness among software providers. The motivation for this examination is that a low degree of awareness makes a presence of a selection bias effect more likely and that there are no adjustments of the estimates among the providers to compensate for the effect.

4.1 In-house Development

One situation where we would expect a relatively low degree of selection bias is in-house development and maintenance of software. While there is likely to be some selection bias even in this environment, e.g., related to cost-benefit optimization of competing investment alternatives, there is likely to be less selection bias than in, for example, bidding rounds with higher effort uncertainty, many providers, and a stronger focus on low price as a selection criterion. Notice that we do not claim that selection bias is the only factor that could explain a lower degree of effort overrun of in-house software development. Potential alternative explanations include less "wishful thinking" and reduced organizational pressure to give low estimates for in-house work, compared with other software development environments. The evidence presented should, therefore, be evaluated, together with other evidence, before concluding on the strength of evidence in favor or disfavor of an important role of selection bias in explaining effort overruns.

In a previous study [20], we completed a systematic search to identify all studies, not only in software contexts, reporting on effort estimation bias. Forty-two studies were identified. Five of these studies explicitly stated that the data were from in-house software projects or tasks. We examined the original data sets of these five studies. We define, as in the previous section, the effort estimation bias as the median RE, where a positive value indicates a tendency toward effort overruns and a negative value indicates a tendency toward effort underruns. Table 3 gives an overview of the estimation bias of the identified studies and suggests that in-house development projects, in general, have no systematic bias towards over-optimistic effort estimates. While the number of studies may not be impressive, the number of tasks and projects included in the studies (1,950) is quite high.

Table 3: Studies reporting in-house development effort estimation bias

Study	**Observations**	**Median effort**	**Bias**
[16]	1,294 software maintenance tasks	2 work hours	0% (neutral)
[14]	506 enhancement tasks and projects	3.8 work days	-15% (underrun)
[15]	21 projects	1,489 work hours	+6% (overrun)
[21]	115 software tasks and	69 work hours	0% (neutral)

	projects		
[2]	14 projects	1,221 work hours	+7% (underrun)

As can be seen, the estimation biases of the tasks and projects in Table 3 are in strong contrast to those of most software project survey results. Such surveys typically report average effort overruns in the range of 20–30% (see, for example, [22, 23]).

4.2 Emphasis on Low Price

As suggested by our model in Section 2, the degree of effort overrun strongly increases with higher price competitiveness, i.e., when the client's priority is on low price and he/she can select among many providers. The survey presented in [2] contains, in addition to the data on the 14 in-house projects displayed in Table 3, data on 28 other projects, many of which involved competitive bidding rounds. Although these projects were of similar size and delivered a similar amount and type of functionality compared to the in-house projects, they had a substantially higher effort overrun (median overrun of 21%). Even more illustrative may be the finding that the projects with the most price-focused clients, which in this case were the public clients, had a much higher effort overrun than the other projects. The median overrun of projects with public clients, when not in-house projects, was as high as 67%! Again, reasons other than selection bias may potentially explain the increase in effort overrun in situations with high price competitiveness, e.g., strategic bidding to obtain a public reference client or "strategic misrepresentation" of the effort estimates. Nevertheless, the finding is consistent with a substantial effect of selection bias on effort overrun.

Another situation with no selection bias, where we would expect much less bias toward effort overruns, is the experimental situation where *all* effort estimates are evaluated, i.e., the situation described in our study in Section 3, where the median effort overrun was only 14%. In total, we found three such software development effort estimation studies. Only one of these studies [24], reported effort estimates that were too low, whereas the other two [25, 26] reported neutral estimates or effort estimates that were too high. A similar lack of a systematic tendency toward effort overruns have been found in effort estimation experiments from other domains. In [20], we found that among the twenty relevant experiments, where all effort estimated were evaluated, eight reported underestimation, one reported unbiased estimates and eleven reported overestimation of effort.

We were unable to identify previous field studies on the effect of selection bias on software project effort or cost overrun. We were, however, able to identified three studies that included surveys on software and other types of engineering projects where comparable projects had been exposed to different degrees of selection bias. Two of them report mainly from other project contexts than software projects, but may nevertheless, we believe, provide relevant evidence:

- The survey reported in [27] included experience from 700 large IT outsourcing projects. It found a strong and significant connection between the "level of competitiveness", defined as whether there was a bidding process or not, and project failure, defined as whether a contract was extended/expanded (indicating success) or not. This connection was present even when adjusting for what they call contract misalignment, meaning a lack of fit between the type of project and the type of contract, and the experience of the vendor. Strongly under-estimating the effort is a frequently reported reason for poor project plans and lack of client satisfaction. Consequently, this result may indirectly suggest the presence of an increased selection bias effect in competitive bidding situations.
- The survey reported in [28] compared the cost overruns of 1,093 projects of various types selected based on either the lowest price given sufficient quality of the proposal or on the "average bid." The "average bid"-format has been developed to avoid the negative consequences of selection bias (or more specifically, avoid the "winner's curse") and prescribes a selection processes emphasizing that the selected bid should be close to the average bid. As can be seen in our model in Section 2, we would expect that the selection bias effect is removed when the selected estimate equals the average estimate, i.e., when $w = 1$. Consistent with a strong selection-bias effect, the study found a strong decrease in cost overruns with the "average bid" and stated (page 6): "*This decrease is quite remarkable, as it is nearly as large as the average cost overrun in the sample.*" In other words, the finding is consistent with a situation where the selection bias explains most of the observed cost overrun in typical low-price focused bidding rounds.
- The study of 76,188 public construction projects in [29] found that contracts awarded based on bidding rounds were much more prone to amendments, which led to cost increases and overruns,

than those awarded based on “negotiation,” i.e., without any competitive bidding rounds. While more than 70% of the bidding round contracts had amendments, this was the case for less than 10% of the negotiation-based contracts. The frequency of very large amendments, e.g., larger than 1,000,000 Euro, was also substantially higher for bidding-based contracts. An explanation consistent with this observation is that the projects selected based on competitive bidding rounds were more likely to be under-estimated. Unless actions are taken, such projects will face financial losses. One action to avoid such losses is for the project to be less flexible about what should be considered inside the project specification and what should be considered an amendment. A higher number of amendments may, consequently, support a substantial effect of selection bias on effort overrun and project problems. The study refers to several other field studies with similar results.

4.3 Awareness of the Effects of Selection Bias

If software providers were aware of and able to adjust for the selection bias effect on their estimates and bids, the effects and/or the negative consequence of this bias on effort overrun may disappear. Such awareness and adjustments have been observed in other domains with bidding formats similar to that in software development, i.e., in so-called sealed bid bidding rounds or auctions. In [30], for example, it is reported that (page 123): “*When objects have uncertain value in sealed bid auctions, the standard competitive effect is offset by a nonaggressive effect as buyers respond to the risk of overestimating the object’s unknown value*.” The author explained the above observations, i.e., less aggressive instead of more aggressive bids with increased competition, as behavior to avoid the selection bias effect, which eventually would lead to financial losses.

As far as we have been able to detect, there is a lack of discussion of selection bias effects in software engineering research and textbooks, which suggests a low awareness of this phenomenon. Results from surveys of reasons for estimation inaccuracy illustrate this. These surveys typically request providers and/or clients to list what they think are the main reasons for estimation errors and biases. We were not able to identify any survey that listed selection bias or related phenomena as responsible for estimation errors (for an overview on such surveys, see [4]).

However, it could be that the actual estimation and/or bidding practice, nevertheless, reflected behavior that adjusted for selection-bias effects. Our search for selection-bias awareness in the software industry showed, for example, that the US Air Force (*Aviation Week & Space Technology*, September 2004, page 23) informed their bidders that "... *if their price was much lower than the service's estimate* [The US Air Force's own estimate of the cost], *it would be seen as a risk to the program, not a benefit.*" This instruction to potential bidders suggests awareness of problems related to selection of low-price bidders, i.e., the motivation is similar to the introduction of the "average bid" format described in Section 4.2. In spite of this observation from US Air Force, our general impression based on search in the research and practice-based software literature, is that the level of awareness of selection bias effects on effort overrun among software professionals is low.

To examine the level of awareness in a more controlled setting, we conducted a study with 53 experienced software professionals from different Norwegian software providers. The study was completed in the context of software cost-estimation seminars. All the participants included in our study had previous experience in project management, effort estimation, and bidding. Although we clearly cannot generalize from this small set of participants to the whole population of software professionals, the results may provide an indication of how well software professionals understand the effect of the client's price focus in selecting among providers on biases in effort estimates and bids. All the software professionals were given the same estimation and bidding scenario. The scenario consisted of a description of a company that participated in a bidding round where:

- The estimation uncertainty was high.
- There were many bidders, perhaps as many as ten, and they had no knowledge of the skill, estimates, or price offered by the other bidders.
- The client either emphasized a low price (Type A client) or the quality of the proposal (Type B client). A Type A client would place a 70% weighting on the price and 30% on the quality of the proposal. A Type B client would place a 30% weighting on the price and 70% weighting on the quality of the proposal. To avoid unintended assumed implications of this difference between the client types, we emphasized in the scenario that the two types of clients would only differ in how they selected among the proposals, not in their subsequent behavior.

- The company was *not* willing to bid strategically, e.g., accept a loss in a competitive situation to secure a reference client, and would only provide a bid it considered likely to return a profit.

The above data were selected to resemble realistic settings. The high estimation uncertainty and the variability in bids have been documented (see for example, [31]). We examined the bidding information from the public (Norwegian) database www.doffin.no to assess the number of bidders and typical weights on price and quality typical in (governmental) software development projects. The last 100 projects in the database were examined. According to the database, 8–10 bidders would be a high, but not an extremely high, number of bidders participating a bidding round. The average number of bidders was four. The same database documented that the weight placed on low price as a selection criterion, as stated in the call for bids documents, varied between 10 and 80%,[2] with an average of about 40%. The findings from the database suggest that the scenario we presented is something that most providers may experience in bidding rounds.

We asked the participants to state how the type of client (the price-focused Type A and quality-focused Type B client) in the above scenario affected: i) the expected actual effort *in the event that their bid was selected* as the winning bid, and ii) what they would propose as a bid that made it likely that they would make a financial profit on the project. The software professionals were asked to select the two statements they thought were correct from those described below:

Statements related to the expected use of effort

1. If your company wins the bidding round, the expected use of effort is the **same** regardless of whether the client is Type A or B.
2. If your company wins the bidding round, the expected use of effort is **lower** when the client is Type A rather than Type B.
3. If your company wins the bidding round, the expected use of effort is **higher** when the client is Type A rather than Type B.

Statements related to the bid

[2] It is mandatory for all governmental clients in Norway to use such weights to state their prioritizations when selecting between the providers' proposals.

1. Your company's bid (price to client) should be the **same** regardless of whether the client is Type A or B.
2. Your company's bid (price to client) should be **lower** when the client is Type A rather than Type B.
3. Your company's bid (price to client) should be **higher** when the client is Type A rather than Type B.

As we have shown in Section 2, the expected effort of the winning bid is likely to be higher when the bid is selected mainly on low price, rather than on high quality, i.e., the *w*-value will be lower with price-focused (Type A) compared with quality-focused (Type B) clients. Given that the expected use of effort of the selected provider is higher when the selection is based on low price, a company could choose to compensate for this by bidding less aggressively, i.e., by submitting a higher bid, to avoid financial losses for Type A clients. In other words, we argue that an awareness of selection-bias effects should lead to the selection of Alternative 3 for both questions.

Table 4: Responses to the scenario questions

Question	**Alternative 1 (same)**	**Alternative 2 (lower)**	**Alternative 3 (higher)**
Expected effort	53% (28 responses)	28% (15 responses)	19% (10 responses)
Bid	53% (28 responses)	36% (19 responses)	11% (6 responses)

As can be seen, most of the software professionals believed that the type of client had no effect on the expected effort when selected as the provider, or what they should submit as a profit-making bid. Only two participants gave the normatively correct answer (Alternative 3) to both questions. When discussing the results with one of these two participants, after the seminar was over, he said that although being aware of the selection bias effect, he unable to convince the other managers about the connection between the client's price focus and their own tendency toward over-optimistic estimates.

5. Managing Selection Bias

From the model's predictions in Section 2 and the experiment in Section 3, we see that selection bias is able to produce a substantial effort overrun in software development contexts. Furthermore, Section 4 provides evidence that is consistent with a substantial selection-bias effect on effort overrun. Selection bias as an explanation of observed bias in estimates and judgments is far from new. It has been discussed under various headings and domains, such as the "the winner's curse" [32] in bidding contexts, "the optimizer's curse" [33] in cost-benefit evaluations, and "post-decision surprise" [18] in general evaluation situations. At the core of all these curses, surprises and disappointments are the same two elements: an uncertainty in judgment and a non-random selection between alternatives. In this section, we briefly discuss how we may be able to better manage selection bias effects in software development contexts. We divide this discussion into sub-sections with focus on the selection bias management in the context of the client (Section 5.1), the provider (Section 5.2) and the software engineering researchers (Section 5.3). We do not intend to provide a full examination of all potential approaches to manage the selection bias in software project contexts or to discuss the feasibility of them in detail. The main purpose is to show that there are ways to reduce the unfortunate effects of the selection bias. We think several of these options should be the subject of practical implementation in industrial contexts, but find also that there is a need for more studies to determine their feasibility and effects.

5.1 The Clients

Potential means for the client to reduce the negative effects of selection bias include: i) reduction of the estimation uncertainty, ii) reduction of the weight put on low price as a selection criterion, and iii) adjustment for the selection bias effect in planning and budgeting.

The clients may reduce the estimation uncertainty, for example, through better requirement specifications, invitation of providers with previous experience from similar projects, and requests for solutions with a low level of cost uncertainty. A reduction of the weight placed on low effort estimates or low price when selecting providers may, in many situations, not only reduce the likelihood of unrealistic plans and budgets, but also reduce the likelihood of selecting less competent providers, see our study on this in [17]. It is, however, not to be expected that a low effort estimate or price will be totally unimportant when selecting between alternatives. If the client chooses to use a low estimated effort or price as an important selection criterion, he/she needs to be especially careful in the selection of development models and planning to avoid unrealistic plans leading to delivery problems. This may, for example, include a focus on frequent re-planning, an extra contingency buffer to deal with unexpected amendments of the contract, and the use of incremental development methods, such as agile and lean development.

As far as we are aware of, there is no documented experience of using the "average bid"-format in software development bidding contexts. While it is likely that this will lead to less selection bias-induced cost overruns, the experience of the "average bid" reported by [28] has not been positive on all aspects. Selection based on "average bid" seems, for example, to be more exposed to manipulation by bidders, who collaborate to increase the average. An approach to avoid the manipulation experience in "average bid" situations may be that used by the US Air Force (see Section 4.3) or other methods based on the use of independent expertise to estimate the realistic level of cost for the project and select providers with an estimated cost close to that level. An alternative way to reduce the weight placed on low price in the selection of providers, which we have found is implemented by a few client organizations, is to pre-select a small number of providers based on competence alone, e.g., a set of only 2–3 providers, and only invite this smaller set for the price-based bidding round. This is likely to decrease the variance in bids and estimates and, consequently, reduce the effect of selection bias on effort overruns.

While it may be possible to demonstrate the existence of a selection bias–induced effort and cost overrun, it will be much more difficult for a client to calculate how large this will be in a particular context and adjust the estimated and bids for this effect. The main reason for this is that this would require information typically not available, e.g., they will not know the mean and the standard deviation of the actual effort of the competing providers. In spite of this limitation, we believe that our model may be used to support the clients in their identification of situations with a high likelihood of selection biased–induced effort overruns, i.e., to identify situations that require extra attention to avoid budget and delivery problems.

5.2 The Providers

The provider may, similarly to the client, try to reduce the estimation uncertainty to avoid negative consequences of selection bias. This may, for example, be achieved through improved estimation processes, improved use of historical data, improved understanding of the system to be developed, and improved communication with the client. While the providers, typically, have no control of the format of the selection process, e.g., how much weight is placed on low price, they have control over whether they submit a project proposal or not. In some situations, the provider may be able to deduce that it is very likely that he/she will only be selected when he/she has estimated the effort over-optimistically, e.g., when there are many other bidders, and the client is likely to use low price as the main selection criterion. In these situations, the provider may chose not to submit a project proposal to avoid financial losses or to participate in spite of the higher risk of financial losses, e.g., for strategic reasons. Similarly to the client, it will be very difficult for a provider to know how much to adjust his/her estimate for the expected selection bias–induced effort overrun. This would, for example, require that he/she knew the distribution of the other estimates of the same project.

5.3 The Researchers

In a previous study [34], we reported that the very high software cost overruns found in the frequently cited 1994 CHAOS-report by the Standish Group were likely to have been caused by a nonrandom sample and unlikely to represent the state-of-practice of the software industry at that time. The selection process was described in the CHAOS report as (page 13): "*We then called and mailed a number of confidential surveys to a random sample of top IT executives, asking them to share failure stories. During September and October of that year, we collected the majority of the 365 surveys we needed to publish the CHAOS research.*" In spite of the survey's intended focus on failure stories, many software researchers have used the cost overrun numbers reported to describe a poor state-of-estimation-practice in the software industry. This misuse of the CHAOS results is perhaps understandable given that the Standish Group's report presented the data as representing the state-of-estimation-practice in the software industry. However, this misuse also indicates that the awareness of the importance of selection bias among cost estimation researchers is not sufficiently high.

It may be less intuitive, and lower awareness of, that even a random selection of completed projects may be exposed to selection biases and give an incorrect picture of the estimation abilities of software providers. The selection bias effect means that researchers should be very careful when presenting and interpreting the estimation accuracy results from surveys of software projects. The accuracy of the cost and effort estimates of completed projects, should *not*, for example, be presented or interpreted as reflecting the estimation abilities or biases of software providers in situations with expected substantial selection bias effects. Instead, the accuracy results should be presented as representing a combination of the providers' estimation abilities and the provider or project selection process, and a set of other factors, see (Grimstad and Jørgensen 2006) for more on the interpretation of estimation accuracy measures.

6. Conclusions

A tendency toward selecting project proposals with low effort and low cost estimates results in the set of evaluated effort and cost estimates being more over-optimistic than the total set of estimates. The strength of this selection bias effect depends on the estimation uncertainty, the weight placed on low effort (or low price when the price is derived from the effort or cost estimates) in the selection of providers, and the extent to which the providers compensate for the selection-bias effect. We found that the available evidence is consistent with a selection bias explaining much of the frequently reported tendency toward too low effort estimates in software projects. The main consequence of this observation is, we believe, the need to emphasize other means, in addition to better estimation methods, to reduce the project problems related to effort and cost estimates that are too low. Such means may, in particular, include the use of alternative, less price-based provider selection formats.

References:

1. Yang, D., et al. *A Survey on Software Cost Estimation in the Chinese Software Industry*. in *ACM/IEEE International Sympsoium on Empirical Software Engineering and Measurement*. 2008. Kaiserslautern, GERMANY: Assoc Computing Machinery.
2. Moløkken-Østvold, K., et al. *A survey on software estimation in the Norwegian industry*. in *10th International Symposium on Software Metrics*. 2004. Chicago, IL: IEEE Computer Society.
3. van Genuchten, M., *Why is software late? An empirical study of reasons for delay in software development.* IEEE Transactions on Software Engineering, 1991. **17**(6): p. 582-590.
4. Jørgensen, M. and K. Moløkken-Østvold, *Reasons for software effort estimation error: impact of respondent role, information collection approach, and data analysis method.* IEEE Transactions on Software Engineering, 2004. **30**(12): p. 993-1007.

5. Thaler, R.H., *Anomalies: The Winner's Curse.* Journal of Economic Perspectives, 1988. **2**(1): p. 191-202.
6. Bazerman, M.H. and W.F. Samuelson, *I won the auction but don't want the prize.* The Journal of Conflict Resolution, 1983. **27**(4): p. 618-634.
7. Flyvbjerg, B., M.K. Skamris Holm, and S.L. Buhl, *What causes cost overrun in transport infrastructure projects?* Transport Review, 2004. **24**(1): p. 3-18.
8. Chen, M. and J. Dyer, *Inevitable disappointment in projects selected on the basis of forecasts.* Society of Petroleum Engineers Journal, 2009. **14**(2): p. 216-221.
9. Wonnacott, T.H.W. and R.J. Wonnacott, *Introductory Statistics*1990: John Wiley & Sons.
10. Campbell, D.T. and D.A. Kenny, *A primer on regression artifacts*1999, New York: The Guilford Press.
11. Cronbach, L.J., Gleser, G. C, Nanda, H., Rajaratnam, N., *The dependability of behavioral measurements: Theory of generalizability for scores and profiles*1972, New York: John Wiley & Sons.
12. Jørgensen, M., U. Indahl, and D.I.K. Sjøberg, *Software effort estimation by analogy and "regression toward the mean".* Journal of Systems and Software, 2003. **68**(3): p. 253-262.
13. Kitchenham, B., et al., *An empirical study of maintenance and development estimation accuracy.* Journal of Systems and Software, 2002. **64**(1): p. 57-77.
14. Hill, J., L.C. Thomas, and D.E. Allen, *Experts' estimates of task durations in software development projects.* International Journal of Project Management, 2000. **18**(1): p. 13-21.
15. Jørgensen, M. *An empirical evaluation of the MkII FPA estimation model*. in *Norwegian Informatics Conference*. 1997. Voss, Norway: Tapir, Oslo.
16. Hatton, L., *How Accurately Do Engineers Predict Software Maintenance Tasks?* IEEE Computer, 2007. **40**(2): p. 64-69.
17. Jørgensen, M. and G. Carelius, *An empirical study of software project bidding.* IEEE Transactions of Software Engineering, 2004. **30**(12): p. 953-969.
18. Harrison, J.R. and J.G. March, *Decision making and postdecision surprise.* Administrative Quarterly Journal, 1984. **29**: p. 26-42.
19. Jørgensen, M. and T.M. Gruschke, *The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty Assessments.* IEEE Transactions on Software Engineering, 2009. **35**(3): p. 368-383.
20. Halkjelsvik, T. and M. Jørgensen, *From origami to software development: A review of studies on judgment-based predictions of performance time.* Psychological Bulletin, 2012. **138**(2): p. 238-271.
21. van Oorschot, K.E., J.W.M. Bertrand, and C.G. Rutte, *Field studies into the dynamics of product development tasks.* International Journal of Opertaions & Production Management, 2005. **25**(8): p. 720-739.
22. 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.
23. Sauer, C., A. Gemino, and B.H. Reich, *The impact of size and volatility - On IT project performance.* Communications of the Acm, 2007. **50**(11): p. 79-84.
24. Connolly, T. and D. Dean, *Decomposed versus holistic estimates of effort required for software writing tasks.* Management Science, 1997. **43**(7): p. 1029-1045.
25. Ohlsson, M.C. and C. Wohlin. *An empirical study of effort estimation during project execution*. in *International Software Metrics Symposium*. 1999. Boca Raton, FL, USA.: IEEE Comput. Soc, Los Alamitos, CA, USA.
26. Ohlsson, N., C. Wohlin, and B. Regnell, *A project effort estimation study.* Information and Software Technology, 1998. **40**(14): p. 831-839.
27. Bapna, R., et al. *Analyzing IT outsourcing contract outcomes: The role of intermediaries*. in *11 th International Conference on Web Information System Engineering*. 2010. St Louis, USA.
28. Bucciol, A., O. Chillemi, and G. Plalzzi, *Cost overrun and auction format in public works*, in *Discussion Paper No. 129, University of Padua* 2011.
29. Chong, E., C. Staropoli, and A. Yvrande-Billon, *Auction versus negotiation in public procurement: Looking for empirical evidence*, in *Working paper/HAL*2010.
30. Meyer, D.J., *Competition and bidding behavior: Some evidence from the rice market.* Economic Inquuiry, 1988. **26**(1): p. 123-132.

31. Anda, B.C.D., D.I.K. Sjøberg, and A. Mockus, *Variability and reproducibility in software engineering: A study of four companies that developed the same system.* IEEE Transactions of Software Engineering, 2009. **35**(3): p. 409-429.
32. Kern, T., Willcocks, and E. van Heck, *The winner's curse in IT outsourcing: strategies for avoiding relational trauma.* California Management Review, 2002. **44**(2): p. 47-69.
33. Jones, C., *Applied software measurement*1991, New York: McGraw-Hill.
34. Jørgensen, M. and K. Moløkken-Østvold, *How large are software cost overruns? A review of the 1994 CHAOS report.* Information and Software Technology, 2006. **48**(4): p. 297-301.