

**Title page:**

**How to Avoid Impact  
from  
Irrelevant and Misleading Information  
when  
Estimating Software Development Effort**

Magne Jørgensen & Stein Grimstad  
Simula Research Laboratory  
[magnej@simula.no](mailto:magnej@simula.no); [steingr@simula.no](mailto:steingr@simula.no)

# How to Avoid Impact from Irrelevant and Misleading Information When Estimating Software Development Effort

**Abstract:** *Software development effort estimates are reported to be highly inaccurate and systematically over-optimistic. We provide empirical evidence that suggests that this problem is caused, to some extent, by the influence of irrelevant and misleading information, e.g., information about the client's budget, present in the estimation material. The only really effective means of eliminating this influence is to avoid exposure to this type of information. Other means, such as more use of formal effort estimation models, improved analysis of requirement specifications, and better selection of estimators, had a positive effect but did not eliminate the influence. We propose process elements that are designed to avoid irrelevant and misleading information and illustrate how this process may lead to more realism in effort estimation.*

**Keywords:** Effort estimation, expert judgment, irrelevant and misleading information.

## Motivation

A recent survey of effort estimation accuracy suggests that the average for software projects is about 30% [1]<sup>1</sup>. It may be unrealistic to expect highly accurate effort estimates, due to the inherent uncertainty in software development projects. Nevertheless, the strong tendency towards over-optimistic effort estimates and the high level of estimation inconsistency among software professionals [3] suggest that there is great potential for improvement in current effort estimation processes.

---

<sup>1</sup> The situation is, however, not as bad as is frequently claimed by those who cite the Standish Group's early CHAOS reports. We have documented the lack of credibility of those reports [2], which for 1994 reported an average cost overrun of 189% (which implies that, *on average*, a project costs almost three times the estimated cost) of so-called "challenged projects".

We claim that software development effort estimation processes, regardless of whether they are based on formal estimation models or expert judgment, will improve if they better address the effect on the effort estimates of irrelevant and misleading information. This claim is based on results from recently conducted empirical studies on software effort estimation and studies from other domains. Our results suggest, for example, that information about future profitable opportunities leads estimators to indulge in wishful thinking and should therefore be removed from the estimation information. We also show that estimators strongly underestimate the level of their wishful thinking in such situations. Information about future opportunities should, of course, be reintroduced when using the effort estimates as input to produce bids for the software projects.

One condition necessary for irrelevant and misleading information to have an impact on effort estimates is that estimation processes, to some extent, are based on unconscious cognitive processes, i.e., the estimators are not always in full control of the factors that affect their expert estimates of required effort or their judgment-based input to formal effort estimation models. We have previously provided evidence to support the validity of this condition through think-aloud protocol studies, interviews and experiments; see [4] for an overview. The partly unconscious estimation process explains why, for example, estimators are poor at explaining why “100 work-hours” feels more correct than “80 work-hours”. Similar results are reported in many other types of judgments [5], and should not be surprising.

### **The Size of the Impact**

To test for the existence of an effect from irrelevant and misleading information on the effort estimates and to indicate its size, we conducted four estimation studies. The studies used software professionals as subjects. The estimation tasks were completed in an experimental environment and on relatively small estimation tasks, but

there are reasons to believe that the irrelevant and misleading information would be observed to have a similar effect in real-life estimation situations with larger estimation task, as well. The main reason for this is that estimators to some extent are unaware of the factors that affect their effort estimates, regardless of realism of context and size of task [4]. Notice that we do not exclude that there are techniques sometimes applied in real-life software projects, e.g., the use of estimation models, which may reduce the size of the impact. We will discuss such means later in this article.

### **Study Design and Analysis Principles**

In all four studies, independent software professionals established that the information was irrelevant for effort usage. In addition, we explicitly instructed the estimators not to use the information in their estimation work in two of the studies. It is possible to argue that some of the information intended to be irrelevant contains implicit messages and, consequently, that it is rational to use it. It is, for example, possible to argue that a client describing a task as a “minor extension” as opposed to an “extension”, as we do in Study 2, expects a lower quality of the work. Even in cases where this argument is valid, e.g., when there is little other information available, we believe it is important to know about the size of the impact from that kind of information. All specifications used in the studies were detailed, i.e., it was not necessary to rely on messages hidden in irrelevant information to estimate the effort.

In all studies there were a few software professionals with much lower effort estimates than the others, e.g., due to different understanding of the task. The mean effort estimates and the standard deviations were strongly affected by these outliers. To avoid potentially biased outlier removal and misleading mean values, we report the more outlier robust median effort estimate. Of similar reasons, we use the non-parametric Kruskal-Wallis test of difference in median values. Notice that the Kruskal-Wallis test of statistical significance only indicates whether group medians are different or not (two-sided test). The strength of our results, we argue, is therefore better than the

strength indicated by the p-value reported by the Kruskal-Wallis test, since the differences are in the expected direction in all of our studies (one-sided test). In addition, an observation of a difference in the same direction as previous results, e.g., in other human judgment domains, strengthens rather than weakens the evidence even when the observed difference is statistically not significant. This consideration is particularly important when the power of the study is relatively low, as is the case in Studies 2, 3 and 4. Contact the authors for more information about the statistical analyses, the data and the study material.

### **Information about Clients' Cost Expectations (Study 1)**

As suggested by evidence presented later in this paper, software professionals often possess information about the desired outcome of their effort estimation work. They may, for example, know the budget, the price-to-win or the cost expected by the client. Study 1 examined the degree to which this knowledge affects the estimates of most likely effort when estimators are instructed not to let it affect the estimation work.

We recruited 165 experienced software professionals and randomly divided them into four groups: Group-HIGH, Group-LOW, Group-VERY\_LOW, and Group-CONTROL. The Group-HIGH, Group-LOW and the Group-VERY\_LOW participants received the information that the client believed that 800, 40, and 4 work-hours, respectively, was a reasonable use of effort for the project. Group-CONTROL participants received no information about the client's expectations. The estimators in Group-HIGH, Group-LOW and Group-VERY\_LOW received the instruction: "*The client does not want you to be affected by his cost expectation in your estimation work, and wants you to estimate the effort that you most likely will need to develop a quality system that satisfies the needs described in the requirement specification.*" All estimators received exactly the same information about the system to be developed.

In spite of the explicit instruction not to use the client's expectation as input to the estimation process, the effort estimates were strongly affected by it. The median effort estimates were as follows:

- Group-VERY\_LOW (n = 43): 60 work-hours
- Group-LOW (n = 43): 100 work-hours
- Group-HIGH (n = 32): 300 work-hours
- Group-CONTROL (n = 47): 160 work-hours

(Kruskal-Wallis test:  $p < 0,001$ . The lower number of Group-HIGH participants is a result of how we distributed the estimation material and is, as far as we can evaluate, not likely to impact the results.)

It is possible that, despite our instructions, the estimators assumed that the client's cost expectation contained some relevant information, about, e.g., the expected quality of the software. We therefore asked the estimators about how much they felt they had been influenced by the client's expectation. Note that this is quite a leading question and that some of the respondents had attended our previous presentations on estimation, when we had warned of the strong impact of clients' expectations on the effort estimates. In spite of those factors, the most frequent response was that the effect on their estimate was in the interval 10-25%. Fewer than 5% of the estimators assessed the effect to be higher than 50%. The typical effect must, as may be derived from the results, have been at least 50% in comparison with the Group-CONTROL participants' estimates. This suggests that even if an estimator is aware that he has been influenced by irrelevant information, he has typically no access to the size of the effect. This is a natural consequence of the unconscious use of information in judgement-based estimation processes.

### **Variation in Wording (Study 2)**

There are many ways of formulating the same requirement in estimation work. To test the effect on estimation of variations in the way a requirement is formulated, we

recruited 65 software professionals and divided them randomly into three groups: Group-LOW, Group-CONTROL and Group-HIGH. All participants received the same programming task specification. The only difference was the variation in words used to describe some of the requirements. The Group-LOW participants' specification included more words that are typically associated with small and simple tasks, the Group-CONTROL participants' included more neutral words to describe the task, while the Group-HIGH participants' specification included more words that are typically associated with large and complex tasks. The irrelevance to use of development effort of the variance in wording was evaluated by independent software professionals. The most notable difference in wording is that Group-LOW participants were asked to complete a "minor extension", Group-CONTROL to complete an "extension", and Group-HIGH to develop "new functionality". All the estimators received the instruction: "*You should not assess how much [the client's name] is willing to spend on this project, but what is required by development work with normal delivery quality*". The median effort estimates were as follows:

- Group-LOW (n = 22): 40 work-hours
- Group-HIGH (n = 20): 80 work-hours
- Group-CONTROL (n = 23): 50 work-hours

(Kruskal-Wallis test:  $p = 0,15$ .)

The results show that the changes in the wording affected the effort estimates strongly. Estimators whose task was described as a minor extension seem to have regarded it as smaller than estimators whose task was described as demanding the development of new functionality. Sometimes the chosen formulation contains important information, sometimes not. The challenge is to identify when the choice of formulation does not provide relevant information and avoid the impact from it.

### **Impact from Future Opportunities (Study 3)**

Project descriptions sometimes include information about opportunities for future business that may come the developers' way if they are selected as the provider for the project. We recruited 32 software professionals to study how much an explicit description of future opportunities may affect the effort estimate if they were instructed to regard this as information as irrelevant to estimation.

The software professionals were divided randomly into two groups: Group-WISHFUL and Group-CONTROL. Group-WISHFUL participants were presented with information that we believed would induce a situation in which the wish to expend little effort would make the estimator believe that little effort would be used. The following is an excerpt from that information:

*"[the client] has invited many providers (more than 10) to implement these extensions and will use the providers' efficiency on this project as important input in the selection of a provider for the development of the new ticketing system ... Estimate the work effort you think you MOST LIKELY will use to complete the described extension to the existing ticketing system. The estimate will not be presented to [the client] and should be the effort you most likely will need.*

From studies in other domains, we know that people frequently are poor at separating between their goals and their most realistic level of performance, i.e., we expected that the above information would lead to lower effort estimates.

Group-CONTROL participants did not receive any information about future opportunities, i.e., they were presented with a situation that was more neutral with respect to wishful thinking. All participants estimated the effort based on exactly the same requirement specification. The resulting median effort estimates were the following:

- Group-WISHFUL (n = 15): 40 work-hours
- Group-CONTROL (n = 17): 100 work-hours

(Kruskal-Wallis test:  $p = 0,4$ .)



It is possible that the estimators believed that they would work more efficiently, and increase the probability that the effort expended would be low, when motivated by a low estimate. While this may be true, results from work-in-progress by Buehler and Griffin at Wilfrid Laurier University suggest that this type of motivation effect is low, is manifest mainly in the beginning of the project, and easily over-rated. We asked the Group-WISHFUL participants about how much they thought they had been affected by the information about the future opportunity. As many as nine out of the 15 participants in Group-WISHFUL believed that had been affected by less than 10%. The other six participants typically believed that they have been affected by less than 30%. As in Study 1, the impact was strongly underestimated.

#### **Amount of Irrelevant Information (Study 4)**

The previous three studies include information that was intended to manipulate the effort estimate. To examine the effect of less manipulative (neutral), irrelevant information on the effort estimates we recruited 76 software professionals and divided them randomly into two groups: Group-CONTROL and Group-IRRELEVANT. The participants received the same estimation instructions and information about the programming task to be estimated, except that a substantial amount of neutral, estimation irrelevant information was added to the Group-IRRELEVANT participants' requirement specifications, e.g., information about systems with which their implementations would *not* have to integrate. The irrelevance to estimation of the added information was confirmed by independent software professionals. The resulting median effort estimates were as follows:

- Group-IRRELEVANT (n = 38): 28 work-hours
- Group-CONTROL (n = 38): 15 work-hours

(Kruskal-Wallis test:  $p = 0,015$ )

The results suggest that the amount of information, and not only its relevance, was used as an indicator of the required development effort. The results also suggest

that a large amount of neutral, irrelevant information is not likely to explain the tendency towards over-optimistic effort estimates, i.e., adding neutral, irrelevant information seems to lead to higher, rather than lower, effort estimates.

### **Ways of Reducing the Impact**

There are several means by which the impact from irrelevant and misleading information in software effort estimation work could be reduced. We evaluated the following:

- Using formal effort estimation models to a greater extent
- Highlighting relevant, or remove irrelevant information from the requirement specification.
- Selecting the most competent developers as estimators.

### **More Use of Formal Effort Estimation Models**

A mechanical use of an unbiased, formal estimation model with only objectively determined input values would eliminate the effect of irrelevant and misleading information. However, such use might not be possible with common software development models, e.g., [6, 7], where important input to the model is, to some extent, the product of human judgment. As an illustration, the huge variation in productivity among software development teams means that the development team's capability in relation to the type of project has to be assessed and provided as input to the models. Unfortunately, as far as we know, there are currently no accurate, objective methods of assessing development team capability in a format that can be used as input to estimation models.

The fact that some of the input to the estimation models is the product of human judgment may be the underlying reason for observing approximately the same level of estimation accuracy and the same tendency towards over-optimism when using formal models and expert judgement; for an overview, read our review in [8]. Hence, the

available evidence suggests that using formal effort estimation models would not eliminate the effect of irrelevant and misleading information. However, there are studies that suggest that the impact may be reduced in comparison to pure expert judgment-based effort estimates, e.g., studies reporting lower effort overruns on large projects [9] or a lower frequency of highly over-optimistic effort estimates [10] when estimation models are used.

### **Highlight Relevant or Remove Irrelevant Information (Study 5)**

In Study 4, we documented the effect of neutral, irrelevant information on the effort estimates. To examine whether simple improvements in requirement analysis would eliminate this effect we recruited 170 software professionals. These were the same as those in Study 1. The difference between the 165 subjects in Study 1 and the 170 in this study is due to five participants not providing meaningful answers in Study 1. We divided the participants randomly into four groups, gave them the same programming task to estimate, but varied the presence of neutral, irrelevant information and instructions for the analysis of requirements. The irrelevance of the information in relation to use of effort was confirmed by independent software professionals. We assumed that if the proposed changes in the analysis of the requirement specification did not lead to significant improvements when only neutral irrelevant information was provided, then it would not help much when more misleading information was provided, either.

The groups were as follows:

- Group-IRRELEVANT: Received a task specification with much irrelevant information added. No requirement analysis instructions.
- Group-HIGHLIGHT\_RELEVANT: Received a task specification with much irrelevant information added. Instructed to highlight with a marker pen the relevant parts of the specification when reading the specification.

- Group-REMOVE\_IRRELEVANT: Received a task specification with much irrelevant information added. Instructed to strike through the irrelevant information with a thick, black-ink pen (this method may be termed the “Black-ink” or “Redact” method”) and reread the specification before estimating.
- Group-CONTROL: Received a tasks specification without irrelevant information. No requirement analysis instructions.

If the effort estimates of the participants in the Group-HIGHLIGHT\_RELEVANT or REMOVE\_IRRELEVANT were close to those in Group-CONTROL and different from those in Group-CONTROL, this would indicate that we had found a simple, promising means of reducing the influence of irrelevant and misleading information. The median effort estimates were as follows:

- Group-IRRELEVANT (n = 43): 40 work-hours
- Group-HIGHLIGHT\_RELEVANT (n = 48): 40 work-hours
- Group-REMOVE\_IRRELEVANT (n = 36): 35 work-hours
- Group-CONTROL (n = 43): 25 work-hours

(Kruskal-Wallis test:  $p = 0.08$ .)

The results suggest that identifying and highlighting the relevant information had no effect, and that removing the irrelevant information had a positive effect, but did not eliminate the impact entirely.

### **Selection of the Most Competent Software Developers (Study 2)**

To examine the extent to which selecting more competent software estimators would reduce the effect of irrelevant information, we re-analyzed the data described in Study 2. In that study, after the estimate of the development task had been completed, we asked the 65 software professionals to assess their competence level related to completing the specified development task. The competence categories were described as follows:

*“My competence in completing the task is: A: Very good, B: Good, C: Satisfactory, D: Close to satisfactory, E: Unsatisfactory”.*

In the analysis, we combined categories A and B to form the category “High skill”, categories C and D to form the category “Medium skill” and denoted “E” as “Low skill”. The groups are based on self-assessed skill relative to one particular task and will, consequently, only provide an indication of the relationship between skill and impact. Table 1 reports the median estimates for the different groups and skill categories. Notice that there are few subjects in some of the fields. As expected, the estimates of those with lower skills were higher.

**Table 1: Group vs Skill**

	<b>Low Skill</b>	<b>Medium Skill</b>	<b>High Skill</b>
<b>Group-LOW</b>	65 work-hours (n=1)	40 work-hours (n=17)	27 work-hours (n=4)
<b>Group-HIGH</b>	250 work-hours (n=6)	80 work-hours (n=9)	50 work-hours (n=5)
<b>Group-CONTROL</b>	180 work-hours (n=4)	55 work-hours (n=12)	24 work-hours (n=7)

The data in Table 1 suggests that the estimators with the lowest skill were affected much more by the irrelevant difference in task wording than those with the highest skill, i.e., the Group-LOW and the Group-HIGH participants’ estimates deviated more from the Group-CONTROL participants’ estimates in the low skill compared to the high skill group. The results also suggest that even those with the highest skill (in Group-HIGH) were affected by changes in wording that were irrelevant to estimation.

Hence, it may be difficult to avoid the effect of irrelevant and misleading information completely by selecting skilled software developers.

## **Recommended Process Changes**

The previous results suggest that the only effective way of avoiding the influence of information that is misleading and irrelevant to estimation is to avoid exposure to it. The following process elements describe one way of achieving this. The recommended process changes are described with a traditional estimation process in mind, i.e., a process in which the client has specified his requirements and expects a fixed price for a project. In spite of this limitation, we believe that most of the recommended process elements will be useful in, and easy to adapt, to other estimation contexts, e.g., to estimation in projects that follow an iterative or incremental development model. Our recommendation should not be regarded as a substitute for, but instead an addition to, good estimation practice as described in, for example, [11].

The changes we recommend include the following elements:

### **Element 1: Estimation Information Preparation**

People other than those estimating the effort should prepare a filtered estimation information package that includes relevant and neutral estimation information only, i.e., where irrelevant and misleading information is removed or at least neutralized. When developing this information package it is important that:

- The software professionals in charge of producing the estimation package are highly skilled software professionals, e.g., skilled senior software developers or project managers.
- The relevance of a piece of information is measured in relation to its impact on the most likely use of effort (Purpose: Realism) and *not* relative to what should be the planned effort (Purpose: Project control), the bid or price (Purpose: Profit), or, the budget (Purpose: Budget control). In other words, this requires that the people in

charge of developing the information package are aware of, and skilled in separating, the different goals of estimation of most likely effort, planning, bidding and budgeting.

- If a piece of information is potentially misleading and highly relevant at the same time, it should be rewritten so that the relevant, misleading information becomes more neutral. Information about the client's unrealistic schedule expectations, for example, may be of this type. It may be relevant for the use of effort, but also lead the estimators to think of the system as small and simple due to wishful thinking.

### **Element 2: Estimation Work**

When estimating the most likely effort it will be important to:

- Ensure that everybody involved in the estimation work understands clearly that the purpose of the estimation work is to derive the most likely use of effort and not, for example, the bid or the planned effort. The importance of this separation is examined in, e.g., [12].
- Exclude estimators that deliberately or accidentally gain access to misleading or irrelevant information that can bias the estimates. In particular, the estimator should not know the "desired" outcome of the estimation process, because this probably will induce wishful thinking.
- Exclude estimators with vested interests in the outcome of the estimation process, e.g., estimators that are very keen on starting the project and may easily fall prey to wishful thinking.

### **Element 3: Adjustments**

When the estimation work has been completed, there may be a need for adjustments and re-estimation. This may, for example, be the situation if the estimated effort is believed to be too high in light of the clients' expectations or if the estimate is

not likely to lead to a bid that can win the bidding round and make a profit at the same time. This situation may be highly vulnerable to wishful thinking and should be treated very carefully. We recommend that the software professional in charge of producing the estimation information package, and not the estimators, updates the information to include less functionality, lower quality, simplified design, or apply other means of reducing the required effort. The estimators should then be asked to re-estimate the effort based on the updated estimation information package. Under no circumstances should the estimators know the desired outcome or receive information that suggests that they need to estimate more optimistically.

### **Examples of Information to be Removed**

To evaluate the potential benefits of the proposed process changes, we examined the project information for software development projects published during the preceding 30 days (search conducted, December 6, 2006) at the project bidding database Doffin ([www.doffin.no](http://www.doffin.no)). Doffin is a Norwegian database that provides bidding information. It advertises government projects and invites government clients to publish requests for all types of projects and services. We identified six software development projects regarding which we could access the relevant estimation information. All projects asked for a fixed price from the bidders.

Our examination of the information confirmed our previous experience from our own estimation work, i.e., that much information that is misleading and irrelevant to estimation was included in the documents. Observations derived from the analysis of this information include the following:

- Two of the clients provided their cost expectations by including their budget in the document where the project requirements were specified. Neither of these clients described what their cost expectations were based on, i.e., it was not possible to ascertain the degree to which the expectations were based on a realistic estimate of



the work involved. As illustrated in Study 1, knowledge of client expectations may strongly reduce the estimators' ability to be realistic. We believe, for that reason, that this information should be removed from the estimation information.

- The projects with the poorest, e.g., the most vague and incomplete, requirement specifications, typically had the largest proportion of the text containing information that was irrelevant to estimation (up to about 80%!). Although there was no obvious misleading information in the estimation-irrelevant information in those specifications, it may nevertheless be important to remove this information to reduce the risk of unwanted effects. The importance of this removal is illustrated by the study described in Study 2, where even innocent changes in wording of the irrelevant information had a large impact on the effort estimate.
- Information about future opportunities related to the software to be developed, e.g., further development and support of the software, were frequently emphasized in the estimation documents. This is no surprise, since the clients typically also asked for the price per hour for maintaining and supporting the software. However, this information may have the unfortunate effect that the estimator starts thinking about the future benefits of winning this contract and, as illustrated in Study 3, fall prey to wishful thinking. Information about future opportunities should, we believe, be removed from the estimation information.
- Published, explicit criteria for selecting providers are mandatory in Norway for government projects of this type. This may have the consequence that a "bidding mode" is induced when estimating the effort, i.e., the goal of winning a contract mixes with the goal of realism. This, in combination with the observed focus on low price described in the selection criteria, may easily lead to wishful thinking, similar to that in Study 3. We believe that the selection criteria should be separated from the estimation information, or, alternatively, be neutralized by a replacement of them with more requirement-directed text that reflects the priorities of the client.

The underlying reason for the existence of irrelevant and misleading information in the above project documents seems, to a large extent, to be that the clients request a price and not an estimate of most likely effort of the project. Understandably, this means that the clients provide information related to pricing purposes, which includes information not necessarily related to and possibly harmful to the estimation of most likely use of effort. The providers should on the other hand, in accordance with good project planning practice [12], begin by providing an estimate of most likely effort and not be impacted by pricing goals.

The proposed changes are clearly not sufficient to remove all types of bias in the estimation work, such as biases that result from lack of information due to incomplete requirement specifications or the problem of forgotten activities due to a lack of checklists and relevant experience. If our recommendations are to work well, they must be used together with high quality estimation processes.

A final warning: Our results may be misused by software clients to deliberately manipulate software developers to provide over-optimistic effort estimates and bids. This is not advisable and may easily lead to, for example, low quality and time overrun; see [13] for evidence. We have started work on how to design bidding processes to reduce the likelihood of receiving bids based on over-optimistic estimate of development cost.

### **References:**

1. Moløkken, K. and M. Jørgensen. *A review of software surveys on software effort estimation*. Proceedings of the International Symposium on Empirical Software Engineering. 2003. Rome, Italy: IEEE Computer Society. p. 223-230.
2. Jørgensen, M. and K. Moløkken-Østvold, How Large Are Software Cost Overruns? Critical Comments on the Standish Group's CHAOS Reports. *Information and Software Technology*, 2006. 48(4): p. 297-301.
3. Grimstad, S. and M. Jørgensen, Inconsistency in Expert Judgment-based Estimates of Software Development Effort. To appear in *Journal of Systems and Software*, 2007.

4. Jørgensen, M. *The "Magic Step" of Judgment-Based Software Effort Estimation*. International Conference on Cognitive Economics 2005. Sofia, Bulgaria: New Bulgaria University. p. 105-1114.
5. Hogarth, R.M., *Educating intuition*. 2001, Chicago: University of Chicago Press.
6. Boehm, B., et al., *Software cost estimation with Cocomo II*. 2000, New Jersey: Prentice-Hall.
7. Symons, C.R., *Software sizing and estimating - Mk II FPA*. 1991, Chichester, U.K.: John Wiley&Sons.
8. Jørgensen, M., Forecasting Software Development Work Effort: Evidence on Expert Judgment and Formal Models. To appear in: *International Journal of Forecasting*, 2007.
9. 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.
10. Jørgensen, M. *An empirical evaluation of the MkII FPA estimation model*. Proceedings of the Norwegian Informatics Conference. 1997. Voss, Norway: Tapir, Oslo. p. 7-18.
11. Lederer, A.L. and J. Prasad, *Nine management guidelines for better cost estimating*. Communications of the ACM, 1992. 35(2): p. 51-59.
12. Edwards, J.S. and T.T. Moores, A conflict between the use of estimating and planning tools in the management of information systems. *European Journal of Information Systems*, 1994. 3(2): p. 139-147.
13. Jørgensen, M. and S. Grimstad. *Over-optimism in Software Development Projects: "The winner's curse"*. Proceedings of the IEEE CONIELECOMP. 2005. Puebla, Mexico: IEEE Computer Society. p. 280-285.