Anchoring and Adjustment in Software Estimation

Jorge Aranda
University of Toronto
10 King's College Road
Toronto, Ontario, M5S 3G4, Canada
1-416-946-8864
jaranda@cs.toronto.edu

Steve Easterbrook
University of Toronto
40 St. George Street
Toronto, Ontario, M5S 2E4, Canada
1-416-978-3610
sme@cs.toronto.edu

ABSTRACT
Anchoring and adjustment is a form of cognitive bias that affects judgments under uncertainty. If given an initial answer, the respondent seems to use this as an 'anchor', adjusting it to reach a more plausible answer, even if the anchor is obviously incorrect. The adjustment is frequently insufficient and so the final answer is biased. In this paper, we report a study to investigate the effects of this phenomenon on software estimation processes. The results show that anchoring and adjustment does occur in software estimation, and can significantly change the resulting estimates, no matter what estimation technique is used. The results also suggest that, considering the magnitude of this bias, software estimators tend to be too confident of their own estimations.

Categories and Subject Descriptors
D.2.9 [**Software Engineering**]: Management – *time estimation, cost estimation.*

General Terms
Management, Economics, Experimentation.

Keywords
Effort estimation, cognitive bias, anchoring and adjustment, empirical software engineering

1. INTRODUCTION
Anchoring and adjustment is a widely observed and documented phenomenon in cognitive psychology. Its effects consist of biasing the answer to a complex question towards an *anchor* (an initial, possible answer). We seem to adjust this anchor to reach a more plausible answer, but the adjustment tends to be insufficient, and our answer biased. Since software estimation is performed by people, under uncertainty, it is subject to cognitive biases such as this. If that is the case, then this heuristic deserves a deeper consideration than it has had to date.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ESEC-FSE'05, September 5–9, 2005, Lisbon, Portugal.
Copyright 2005 ACM 1-59593-014-0/05/0009...$5.00.

We conducted an experiment to test the effects of this form of bias on software estimates. Participants were given a detailed project description, and asked to estimate the time needed for a specific team to complete the project. In two conditions, the participants were given an initial estimate, one very low, the other very high. In a third, control condition, no initial estimate was given. Participants were asked to provide both an estimate, and a confidence interval for their estimate.

Section 2 presents fundamentals and related work on the two subjects of this paper: Software estimation and anchoring and adjustment. Sections 3 through 5 describe the design and execution of the experiment. Finally, sections 6 and 7 provide the experiment results and conclusions.

2. FUNDAMENTALS AND RELATED WORK
There is a lack of research relating the work in Cognitive Psychology on anchoring and adjustment with the research in Software Engineering on software estimation. However, there is a wealth of information on each field separately.

2.1 Anchoring and Adjustment
Anchoring and adjustment is a cognitive bias observed when people must make choices under uncertainty, and is particularly evident when the result of the choice can be expressed as a number in a range. If judgment of the matter is difficult, we appear to grasp an *anchor*, that is, a tentative and possibly unrelated answer to the problem; and *adjust* such answer up or down according to our intuition or experience to reach the final result. The adjustment applied to the initial anchor is frequently insufficient to compensate for the negative effects of the anchor. Anchors, then, have the effect of attracting answers towards them and away from the correct result.

Tversky and Kahneman [24] first reported this phenomenon with the following experiment: Participants were individually presented a wheel of fortune with numbers from 0 to 100. The experimenter spun the wheel in front of the participant, and after it stopped in an evidently random position, he questioned the participant to estimate various quantities, stated in percentages. For example, participants were asked to give the percentage of African countries that were members of the United Nations. Participants were first asked to indicate if the correct answer to the question was higher or lower than the random number that came up in the roulette, and then to estimate the correct value by moving upward or downward from the random number.

Tversky and Kahneman report that the arbitrary initial numbers obtained from the roulette had a marked effect on estimates: the median estimate for the African countries question was of 25 for people that received a 10 as their anchor, and 45 for those who received a 65. The researchers summarized the phenomenon as "different starting points yield different estimates, which are biased toward the initial values".

Since then, the phenomenon has been studied thoroughly, and although the cognitive processes involved in it have not been singled out, its existence is now rarely questioned. It has been shown to occur in situations as diverse as general knowledge issues, probability estimates, legal judgment, pricing decisions and negotiation [21].

For example, [7] indicates that anchoring occurs in legal applications, and suggests that "plaintiffs would do well to request large compensation awards" to bias awards granted by jurors. [22] demonstrated that professional real estate pricing decisions are also subject to anchoring biases, altering the pricing decisions of both experienced and inexperienced real estate professionals.

Initial anchors do not even need to be recognized as starting points for a solution. [2], for example, affirms that the duration of a criminal sentence partially depends on numbers that are fresh in the mind of the sentencing judge. However, semantic anchoring effects are more potent than purely numeric effects; that is, the anchor is more effective if it is regarded as a possible, meaningful solution to the problem at hand [21].

Finally, a series of experiments by Wilson, Houston and Brekke [25] indicate that (a) anchoring occurs if people pay sufficient attention to the anchor value, (b) knowledgeable people are less susceptible to anchoring effects, and (c) anchoring appears to operate unintentionally: it is difficult to avoid even when people are forewarned.

2.2 Software Estimation

Effort estimation for software projects has proven to be an elusive and expensive problem in software engineering. On one hand, stakeholders expect precise estimates in the early stages of a project; on the other hand, reliably producing those numbers is extremely difficult and may well be technically infeasible. Boehm et al. [5] report that estimating a project in its first stages yields estimates that may be off by as much as a factor of 4. Even at the point when detailed specifications are produced, professional estimates are expected to be wrong by ±50%.

This precision problem is compounded with the confusion surrounding the term 'estimate'. While managers and clients make their plans assuming that software projects are likely to be finished at, or close to, the estimated time, developers tend to produce estimates that only work for a best-case scenario. According to DeMarco [8], the default definition of estimate among professionals is "the most optimistic prediction that has a non-zero probability of coming true". He argues that a better definition is "a prediction that is equally likely to be above or below the actual result", which seems to be the definition that most software estimation researchers use (although it is still too risky for most real business plans).

If estimates are predictions, we should explore the psychology of human prediction processes. But unfortunately, according to Brown and Siegler [6], "psychological research on real-world quantitative expert estimation has not culminated in any theory of estimation, not even in a coherent framework for thinking about the process". It is not surprising then that software engineers prefer to create mathematical estimation models than to explore the intricacies of human judgment applied to software estimation.

The Constructive Cost Model (COCOMO, [4]) is probably the most widely known method for software estimation. In its original incarnation, its core effort equation uses lines of code (LOC) as an input, and the equation can be adjusted to account for particulars of each software project. Boehm claims that the intermediate version of the model renders results that are within 20% of actual numbers 68% of the time. However, other empirical validations suggest that the performance of the model is much worse, especially if it has not been carefully calibrated for the organization in charge of the project [20, 14].

There are several arguments against the use of COCOMO and other LOC-based models. One of the most powerful is that they require the estimator to predict the number of lines of code the future system will have, a quantity that is as unknown to the estimator as the time it will take to produce them, but even less intuitive. Estimators are better at estimating effort than size, which cancels the benefits of size-based models (although they generally do not seem to be very good at either) [12]. Critics argue that COCOMO disguises the guesswork of estimating, but it does not eliminate it [15]. Furthermore, an analysis of the reported project data of several empirical validations of estimation models shows that the size-effort correlation is not evident and size may not be the primary determinant of project effort [10]. It is the creative content and the quality imbued in the code, not its number of lines, which determines the required effort for an application.

Another popular set of estimation techniques is based on *function points* (FPs) [15]. FPs remove many of the inconveniencies of LOC metrics since they are based on the required functionality of the desired software product.

However, there are still factors that make FPs an inaccurate technique. One is the variation in the productivity of developers. For example, it has been found that the best programmers are 10 times more productive than the worst, and 2.5 times better than the median [9]. Team performance variations are also extremely wide. Another factor is an incomplete or defective specification. Badly stated requirements can increase a project's time and cost to several times its intended values. And finally, software development needs a degree of creativity, inventiveness, and social interaction that is extremely difficult to capture in an estimation model.

Model-based estimation is not the only alternative available for software engineers, nor the most widely adopted [12]. Learning-based techniques [23], for example, help estimators to establish analogies between the project at hand and previous experiences, and they are helpful when performed in a familiar, predictable environment.

The most commonly used estimation method, which can be called *expert-based estimation*, is arguably the method with the worst standing among software engineering researchers. Although there are ways to structure this technique, such as the Delphi process [11] or work breakdown structure analyses [3], its basic feature is the lack of a mechanical process to estimate. Instead, experts are

assigned the responsibility of reaching an estimate by whichever means they see reasonable. However, for all its fuzziness, it is not clear that other methods are more effective than expert-based techniques, as empirical validations provide conflicting results on the superiority of any technique [16].

Even though much research has focused on software estimation, it is still an ambiguous process. This is relevant for us because ambiguous and uncertain thought processes are prime candidates to be victims of judgmental biases.

The relevance of human thought processes is present in all estimation techniques, even model-based, where humans need to define the input parameters that models require.

There is a growing amount of research exploring software estimation as a primarily human activity [16]. It has been found, for example, that the confidence estimators have in their own estimates is unjustifiably high, that they do not seem to distinguish between several degrees of confidence in an estimate [17], and that experience is not a good indicator of expertise when it comes to software estimation [13]. Of most relevance for this paper, it has also been recently found that anchoring and adjustment affects estimates on coursework for computer science students [19], and that customer expectations affect estimates of short software tasks when using work breakdown structure analyses [18].

Although these studies hint that it is reasonable to expect anchoring and adjustment biases in software estimation, we are not aware of any empirical study explicitly exploring this effect in the estimation of software projects. Considering the economical and personal impact that incorrect estimations carry, it is important to inquire experimentally the influence of this cognitive bias on the matter.

3. RESEARCH QUESTIONS

Software estimation is essentially a human judgment activity, and as such it is subject to judgmental biases. Efforts to standardize estimation, although successful in giving shape to such an activity, do not eliminate or reduce the effect of human bias.

In order to find out if software estimation is affected by anchoring and adjustment, we set the following as our research questions:

- Does the phenomenon of anchoring and adjustment take place in software estimation processes?
- Is the influence of anchoring and adjustment weaker for estimators that have had previous experience estimating software projects?
- Is the influence of anchoring and adjustment stronger for estimators that rely solely on expert-based estimation, as opposed to estimators that use a model-based technique?
- Does the confidence (or lack thereof) estimators have in their answers compensate for possible anchoring and adjustment biases?

The experiment reported here provides some answers to all of these.

4. EXPERIMENT PLAN AND DESIGN

4.1 Experiment Design

The experiment consisted of a software estimation exercise that participants worked on individually. They were asked to estimate the time it would take for a specific development team to deliver a particular software application. The application, a fictional software project for international commerce statistics based on a real project developed by one of the authors, was described in a ten-page requirements document and a three-page project setting document [1]. The requirements document stated the functionality necessary for the system to be developed, as well as notes for relevant non-functional requirements. The project setting document gave participants informal data on two areas: the client organization (their work culture, hierarchy, and "quotes" from interviews with them) and the development team in charge of the project (their language experience, previous projects performance and team dynamics).

Participants were given as much time as necessary to produce their estimates, but most of them reported taking around two hours to complete the exercise. Participants had complete freedom on their choice of estimation techniques, as long as they worked on the exercise by themselves. They could use software estimation tools to aid their judgment if they desired.

Once participants performed their calculations, they were given a questionnaire. The two most relevant questions were:

- "Give your estimate for the duration of the project described in the attached documentation, in months, to the nearest integer", and
- "I think that if this project was really developed, my estimate might be off by as much as ___%"

Additionally, participants were asked to give a justification for their estimates, to describe their previous estimation experience, and to rate the information they read in the documentation.

Each participant was paid $10 for their involvement in the study.

The experiment had three conditions. The only difference between them was a paragraph in the second page of the project setting document. In a box with quotes from a middle manager of the client organization, a sentence was altered in each group:

For the experiment's control condition, the manager was quoted as saying: *"I'd like to give an estimate for this project myself, but I admit I have no experience estimating. We'll wait for your calculations for an estimate."*

For a second, "2 months" condition, the quote was modified to include an anchor. It read as follows: *"I admit I have no experience with software projects, but I guess this will take about 2 months to finish. I may be wrong of course, we'll wait for your calculations for a better estimate."*

Finally, a third, "20 months" condition, had another anchor in the manager's statement. It was exactly the same as the second group, except for a change from *"…I guess this will take about 2 months to finish..."* to *"…I guess this will take about 20 months to finish…"*. The conditions were equal in all other aspects.

There are several issues worth noting at this point: First, the difference among anchors is of an order of magnitude. This difference is quite large, but for early stages of a software project, not completely far-fetched [5]. Second, the anchor given to participants is semantically linked to the answer they are asked to provide. According to Mussweiler and Strack [21], semantic anchors are more powerful than simple numeric anchors. Third, the manager does not push for his guess to be considered as a starting point for negotiation. He admits that he has no experience with software projects, that he may be wrong, and labels his own quantity as a guess. And fourth, participants did not hear the individual saying this sentence, they did not meet him in person, and the sentence was not highlighted in the document. Participants are thus less likely to be socially influenced and to try to please the manager by giving a final estimate that confirms his guess than if they had actually sat with him in an interview and were told just that. Attempting to please is also a judgmental bias, but of a social, not cognitive, nature [2]. The research questions of this study focus on cognitive aspects, and therefore it was important to limit the influence of social biases in the experimental design.

4.2 Variables

The following variables were recorded:

Independent Variable: The only independent variable was the anchoring statement discussed in the previous section. It had three values: "2 months" anchor, "20 months" anchor, and no anchor.

Controlled Variable: While assigning participants to each condition, we attempted to reach a similar number of experienced participants in all conditions. Experience was classified in three levels: Experience in large/medium software projects estimation; experience in small software projects estimation; and only academic experience. Experience was self-assessed; each participant's definitions of project size, involvement in estimation, and amount of time dedicated to learning estimation processes were not probed.

Dependent Variables: Three dependent variables were considered of relevance for this study: (a) the actual estimate as given by participants, which is a positive integer representing a number of months; (b) confidence range, expressed as a percentage that can be added or subtracted from an estimate to reach an acceptable range of results; and (c) estimation method. Participants were not asked to name the estimation method that they used, but they were asked to provide a justification for their estimate. These justifications were analyzed to classify the estimation technique as being either model-based or expert-based. Further classifications within each subgroup were LOC-based or FP-based (for model-based techniques) and WBS (work breakdown structure) or unstructured process (for expert-based techniques). If a participant used more than one technique, an assessment of their primary technique was performed.

4.3 Hypotheses

The following are the null hypotheses for this experiment:

$H_{0,\ LOW\text{-}HIGH}$: Estimates of participants given a low ("2 months") anchor are not statistically different from estimates of participants given a high ("20 months") anchor.

$H_{0,\ LOW\text{-}CONTROL}$: Estimates of participants given a low ("2 months") anchor are not statistically different from estimates of participants given no anchor at all.

$H_{0,\ HIGH\text{-}CONTROL}$: Estimates of participants given a high ("20 months") anchor are not statistically different from estimates of participants given no anchor at all.

A similar set of hypotheses was generated for analyzing the results of experienced participants, model-based techniques users and expert-based techniques users.

To address our last research question, regarding the confidence of estimators in their own results, one additional hypothesis was generated:

$H_{0,\ MaxLow\text{-}MinHigh}$: The maximum estimates of participants given a low ("2 months") anchor are not statistically different from the minimum estimates of participants given a high ("20 months") anchor.

4.4 Threats to Validity

The following discussion on threats to validity is based on the list of threats proposed by Wohlin et al. [26].

Conclusion validity: The group of participants that performed the software estimation exercise was relatively heterogeneous, consisting of software industry professionals and computer science graduate students. We have no way of assessing whether these participants are a representative sample of the broader population of estimators. Another aspect of this threat is that some participants (43%) had only academic experience of software estimation (through coursework and self-learning) and had not been asked to produce an estimate in a real-world software project previously. However, all participants had at least basic qualifications to perform real software estimation; that is, they all were potential software estimators with enough authority, either because of background, academic formation or a combination of both, to produce estimates in real software development projects. For this reason they may be regarded as part of the same group, and this threat is reduced.

It is also possible that the task might not have been representative of real estimation tasks. For example, it may be unusual to require that estimates be provided in months, or to estimate a project for an organization with which the estimator has had no direct contact. However, we think our study replicates real estimation tasks adequately, within the restrictions of a controlled experiment.

Internal validity: Respondents might be reacting to a social bias rather than the intended cognitive bias. Furthermore, it is possible that participants might have put even more importance on the anchor than we intended, interpreting it as a hint about available resources. However, as discussed previously, we tried to minimize these possibilities with the way our anchor was presented. The opposite is also possible: respondents might fail to notice the sentence with the manager's estimate. We believe that if this was the case it would lead to very different results than the ones we obtained.

Construct validity: This experiment might suffer from a mono-operation bias, since it used only one set of project specification documents. It would have been interesting to perform it with

several (at least two) different software projects and to see if the relations between conditions are replicated among them. Economical limits and a difficulty to find participants prevented the study from going in that direction.

External validity: Our participants were volunteers who responded to an invitation. However, volunteers are especially motivated, and are therefore not representative of the whole population [26]. This was a necessary evil, since hiring a significant number of professional estimators and paying them their usual fees for their services was not economically feasible.

Another threat to external validity lies in the fact that real software estimation carries consequences that may be suffered for a long time by the people involved, potentially altering their career paths. Participants in this study knew that they would not be held accountable for their answers, and this difference between the experiment and real estimation experiences may affect the results. This, unfortunately, is a consequence of performing a controlled experiment. The alternative would be to observe real software estimations within their natural environments.

5. EXPERIMENT EXECUTION

Our experiment took place during the second half of 2004. Participants were recruited through email invitations sent to graduate computer science students and software developers. After candidate participants expressed their interest they were visited at the time and place of their choice and they were given their documentation package.

Participants were not told the purpose of the study. They were told that they were participating in a software estimation experiment, without going into further detail. All participants signed a consent form and were guaranteed anonymity.

Participants were allowed to work on the study whenever they wanted, although most of them reported having finished within three days of being visited. They could use software tools and reference books if they wished.

23 people participated in the study. The majority of them (78%) were graduate students in Computer Science, the remaining 22% were professionals from the software industry. 57% of participants declared they had been involved in real software estimation activities before (22% were involved in medium to large software projects, 35% only in small projects). 43% had only academic experience in the area.

An even distribution among conditions was intended. The final number of respondents, however, was variable among conditions due to participation cancellations. The "2 months" condition received 9 responses, the control condition, 6 responses, and the "20 months" condition, 8 responses.

Each participant's answers were recorded, and their estimates were analyzed using independent *t*-tests for each hypothesis.

6. DATA ANALYSIS AND INTERPRETATION OF RESULTS

6.1 General Results

The responses presented a very wide range of estimates: from 3 to 28 months. The average estimate was 10.9 months. Participants gave their estimates a confidence interval of ±26% on average (minimum 10%, maximum 60%).

Two types of estimation techniques were used: model-based and expert-based estimation. 31% of estimators chose primarily a model-based technique (22% LOC-based, 9% FP-based). The remaining 69% used an expert-based technique (39% with work breakdown structures, 30% unstructured process).

Figure 1, on the next page, presents all estimates. The chart is divided in three areas. The lower area corresponds to the "2 months" condition, the middle area to the control condition and the higher area to the "20 months" condition. For each estimate the graph includes the confidence interval of its estimator. The graph also shows the mean estimate for each group and the anchors for the "2 months" and "20 months" conditions.

Although the patterns on each condition are visible on the chart, the following numbers help to clarify it. The "2 months" participants had a mean estimate of 6.8 months. The control condition has a slightly higher mean estimate, at 8.3 months; and the "20 months" condition's mean estimate is 17.4 months.

Within each group there is a considerable variation as well. The "2 months" condition's greatest estimate is 4.33 times higher than its lowest. The corresponding proportion is 3.75 for the control condition and 2.80 for the "20 months" condition.

The *t*-test results are: For hypothesis $H_{0,\ LOW\text{-}HIGH}$, $t = 4.273$, the null hypothesis is rejected ($p < 0.001$). For hypothesis $H_{0,\ LOW\text{-}CONTROL}$, $t = 0.661$, the null hypothesis cannot be rejected ($p > 0.1$). And for hypothesis $H_{0,\ HIGH\text{-}CONTROL}$, $t = 3.137$, the null hypothesis is rejected ($p < 0.01$).

Therefore, these results show that the anchoring and adjustment bias takes place in software estimation processes, at least when the effects of providing a high anchor are compared with the effects of providing a low anchor or no anchor at all. However, no significant difference between low anchors and no anchors was found.

6.2 Experienced participants' results

If we consider only the results of those participants who declared to have real-life estimation experience (57% of the total number of participants) we obtain Figure 2 (on next page).

As can be seen in the chart, the pattern remains unchanged after removing inexperienced estimators, although the statistical significance is slightly weaker due to the reduced number of participants.

Within this subgroup of experienced estimators, the "2 months" condition mean is 7.8 months. The control condition has a mean of 9.0 months; the "20 months" condition is at 17.8 months.

The *t*-tests for the subgroup of experienced participants provide the following results: For hypothesis $H_{0,\ LOW\text{-}HIGH,\ Experienced}$, $t = 3.150$, null hypothesis rejected ($p < 0.02$). For hypothesis $H_{0,\ LOW\text{-}CONTROL,\ Experienced}$, $t = 0.425$, null hypothesis cannot be rejected ($p > 0.1$). For hypothesis $H_{0,\ HIGH\text{-}CONTROL,\ Experienced}$, $t = 2.462$, null hypothesis rejected ($p < 0.05$).

These results indicate that the effect found in the generality of participants was also suffered by experienced estimators in particular.

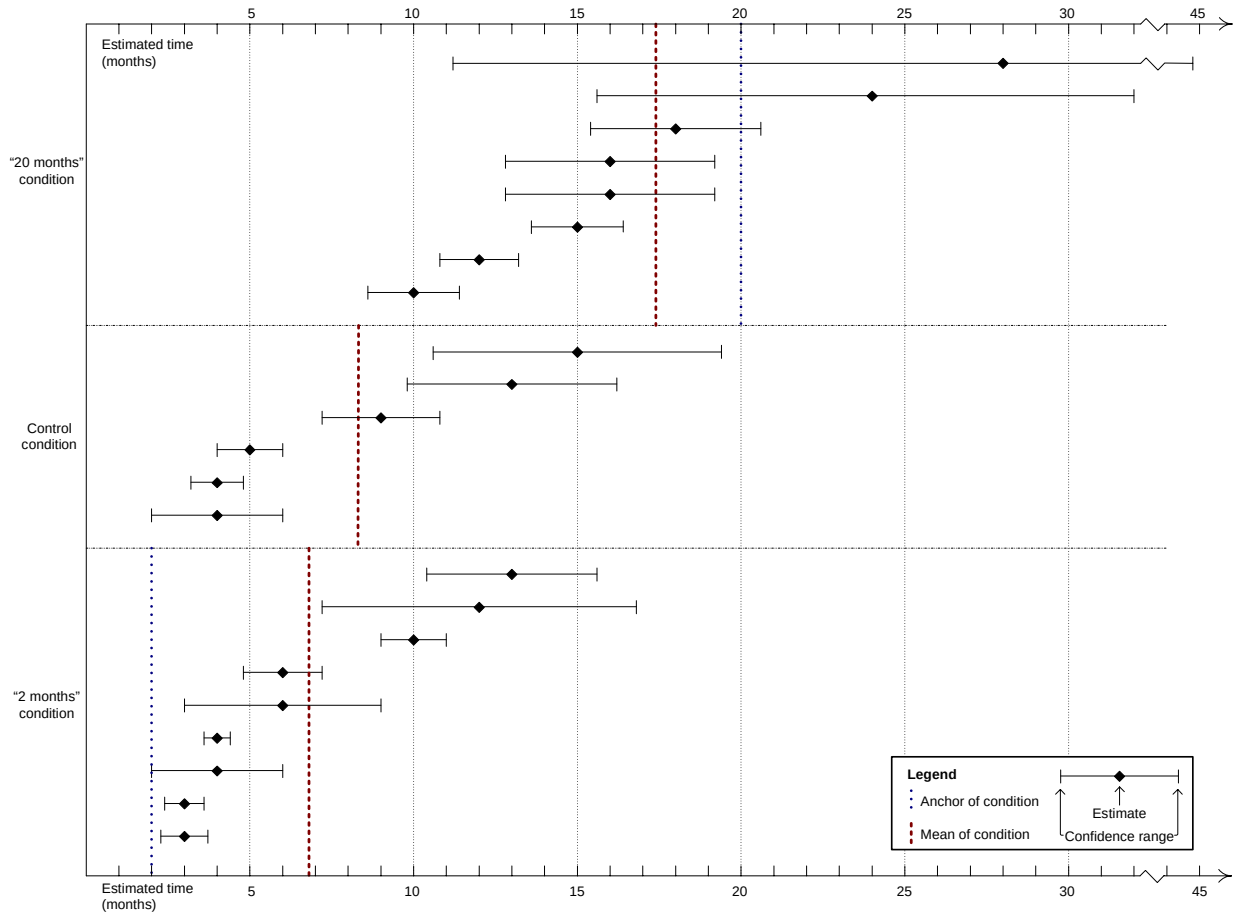


Figure 1. All estimators' results

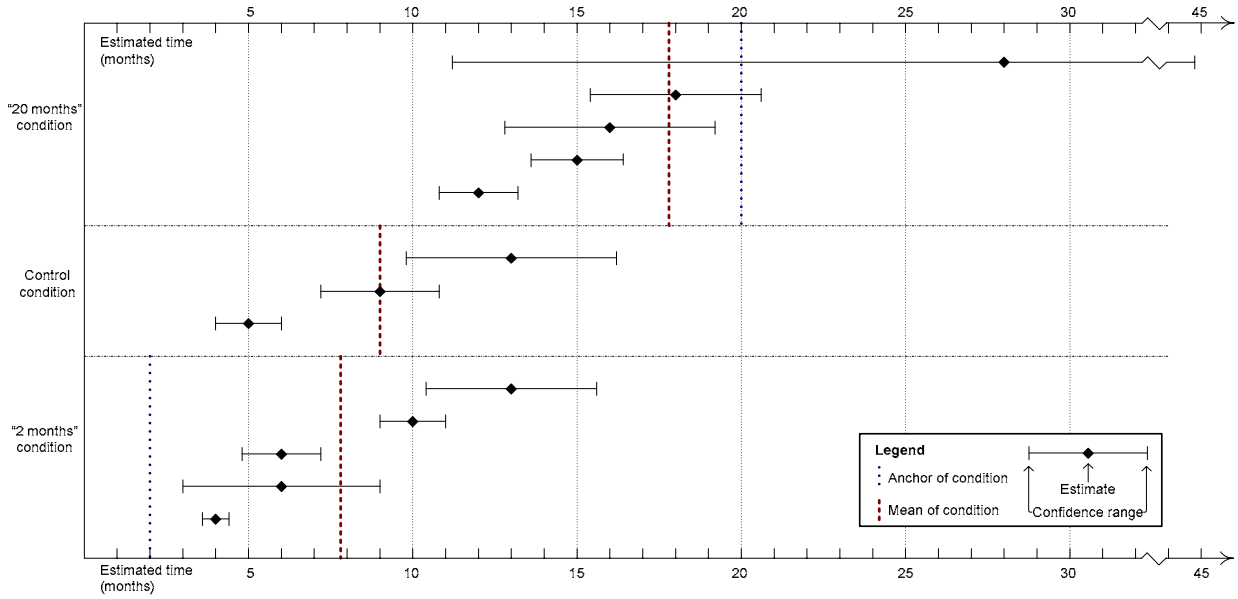


Figure 2. Experienced estimators only

6.3 Expert-based estimators' results

As shown in Figure 3, the same general pattern appears for estimators who used an expert-based technique, with two notable differences: First, the averages are lower than in the complete pool of participants. Second, standard deviations are also lower, indicating more homogeneous results.

The particular numbers are as follows: For the "2 months" condition, the mean is 5.1 months. For the control condition it is 7.8 months. For the "20 months" condition it is 15.4 months. Applying independent *t*-tests for each of the three relevant null hypotheses yields the following: Hypothesis $H_{0,\ LOW\text{-}HIGH,\ Expert\text{-}based}$, $t = 7.567$, rejected ($p < 0.001$). Hypothesis $H_{0,\ LOW\text{-}CONTROL,\ Expert\text{-}based}$, $t = 1.154$, cannot be rejected ($p > 0.1$). Hypothesis $H_{0,\ HIGH\text{-}CONTROL,\ Expert\text{-}based}$, $t = 3.358$, rejected ($p < 0.02$).

Again, this shows that effects of anchoring and adjustment are suffered by estimators who choose an expert-based approach. In fact, these results were among the most powerful of the experiment. However, an effect comparing low anchor estimates and no anchor estimates was not found in this subset either.

6.4 Model-based estimators' results

The complement of the previous subgroup is that of estimators who used primarily a model-based technique to reach their results. Figure 4 shows their data.

Since only 7 estimators chose to use models, there are not enough data points to reach conclusions for them. Even though the same pattern as in previous groups is noticeable here, the sample was not large enough to provide statistically significant results.

The numbers for this group are as follows: The mean of the "2 months" condition is 12.5 months, for the control condition 9.5 months, and for the "20 months" condition 20.7 months. The three null hypotheses concerning model-based estimators could not be rejected with independent *t*-tests ($p > 0.1$ in all cases). It is impossible to know if this was due to the low number of participants choosing model-based techniques or due to a weaker influence of anchoring and adjustment effects on this subgroup. Although existing data seems to indicate the former, it is not conclusive.

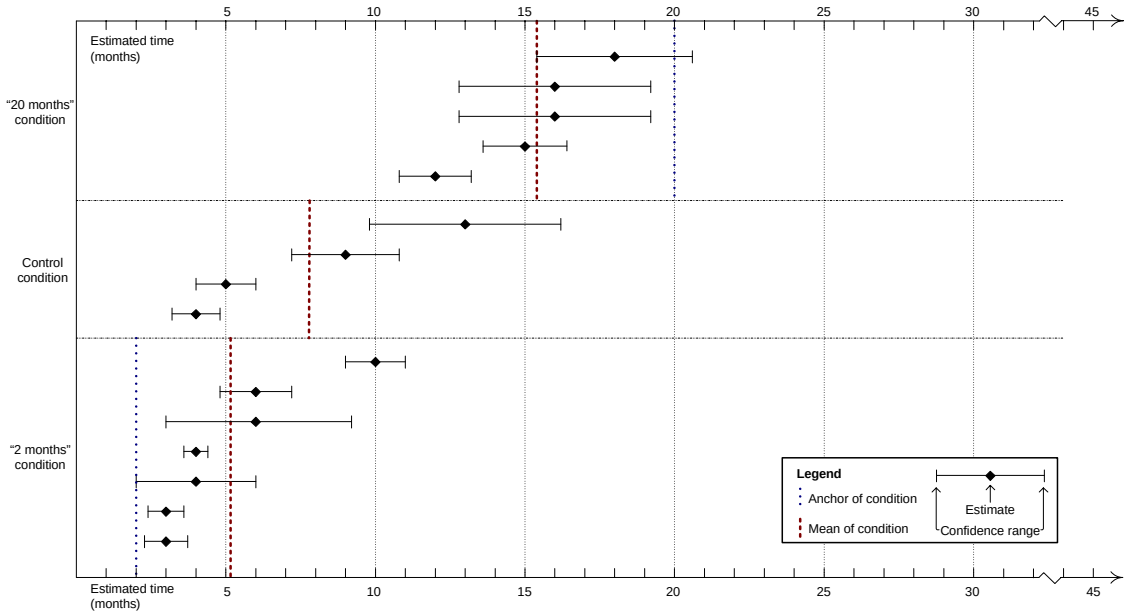


Figure 3. Expert-based techniques users

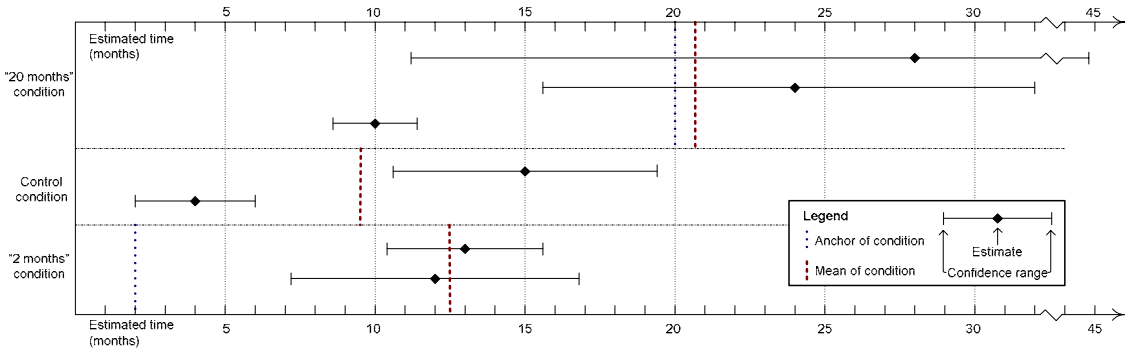


Figure 4. Model-based techniques users

6.5 Confidence ranges

To explore whether participants' confidence ranges compensate for anchoring biases, we considered the data from Figure 1 again. A *t*-test between the "2 months" and the "20 months" conditions was performed, but considering the *maximum* (worst-case) estimates from the "2 months" group and the *minimum* (best-case) estimates from the "20 months" group. This addresses our last experimental hypothesis.

The new numbers are as follows: The best-case estimates on the "2 months" condition have a mean of 8.7 months. The worst-case estimates on the "20 months" condition average 12.8 months. The result of the *t*-test for $H_{0,\ MaxLow\text{-}MinHigh}$ yields $t = 2.182$, and the null hypothesis is rejected ($p < 0.05$).Therefore, the effects of anchoring and adjustment seem to be so high that giving estimators the opportunity of including a confidence range in their estimates does not compensate for their biases.

Additional insights are found if the data from each condition are concentrated to show the general agreement that estimators may have among themselves. Figure 5 displays, for each condition, the percentage of estimators who included each month within their confidence range. An initial observation is that agreement among estimators is rather low. In the "2 months" condition, agreement peaks at 56%, at the 4 months line. For the control condition the maximum agreement is 50%, at the 4, 5 and 11 months points. Finally, the maximum agreement for the "20 months" condition is 63%, at the 16 and 17 months points.

As participants in all conditions actually estimated the exact same project, it is reasonable to merge the three charts and see the general agreement among estimators. Figure 6, on the next page, shows this information. Once all estimators are considered, the maximum agreement is very low (39%), and appears at two points (11 and 16 months).

This indicates that, were this project truly developed, *at least* 61% of the estimators would have been wrong in their predictions, no matter how long the project actually took. This is perhaps not perplexing considering how often estimates miss their targets in real software projects.

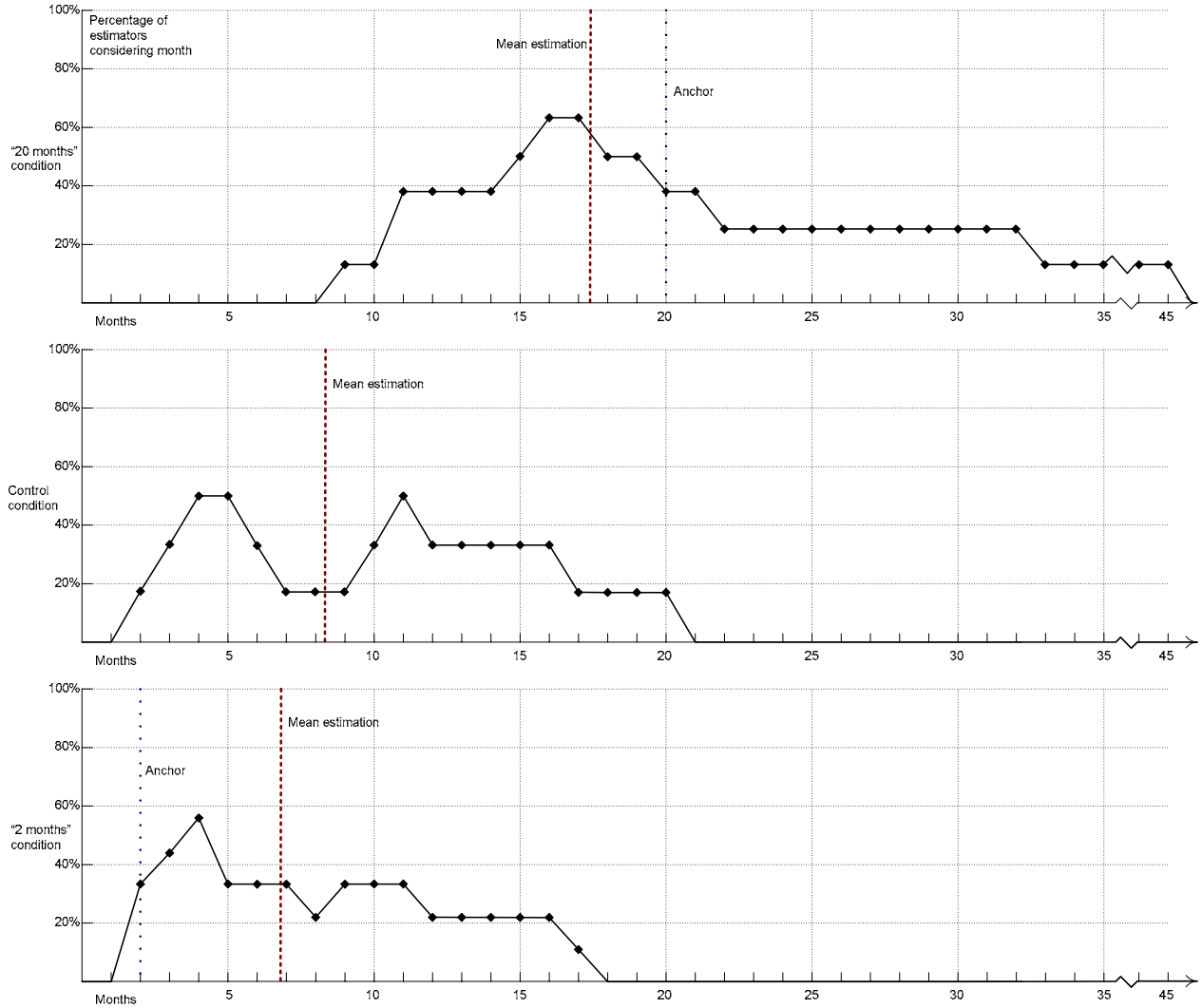


Figure 5. Estimate ranges results by condition

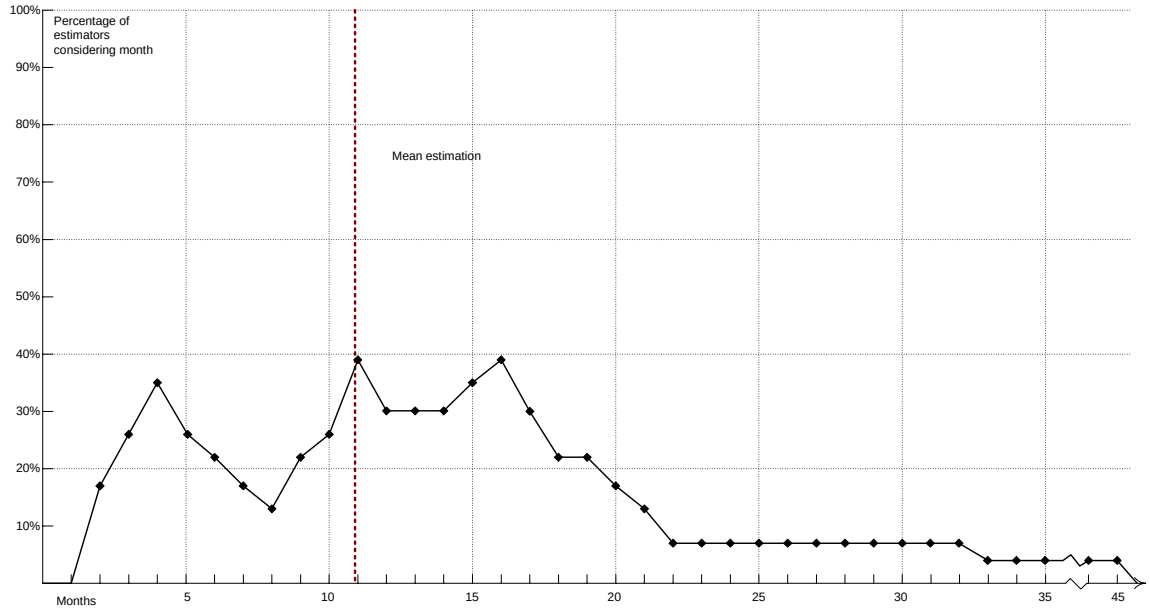


Figure 6. Estimate ranges results, concentrated

7. DISCUSSION AND CONCLUSIONS

Our results show that the anchoring and adjustment heuristic does take place in software estimation processes. When estimators are given a high anchor their estimates are significantly higher than when they are given a low anchor or no anchor at all. This effect is too strong to be ignored. On average, estimates on the high anchor condition were more than twice as long as those in the low anchor condition. The effects were so large that even the worst-case scenario produced by estimators in the low anchor condition is significantly more optimistic than the best-case scenario from the high anchor condition.

Furthermore, the effect is maintained across experienced estimators and users of expert-based techniques (who presented the strongest effects of this bias). However, although the trend seems to occur in users of model-based techniques as well, their data were not conclusive. The difference in effects between low anchors and no anchors was not conclusive either.

There are at least four possible reasons why no difference was found between low anchor and no anchor conditions. First, estimators may be optimistic by nature, so participants in the control condition could be substituting external anchoring effects with their internal optimism. Second, the value for the low anchor (2 months) may not have been low enough. Third, a greater pool of participants may be needed to identify the differences between these two groups. And fourth, it is possible that low anchors do not affect estimation processes as powerfully as high anchors.

There are several ways to expand this experiment to continue exploring this effect. In particular, we could run the experiment with other estimation units. Estimators were asked to provide their estimate in months, and the anchor was also provided in months. It would be interesting to see the effect of giving estimates in weeks when the anchors “2 weeks” and “20 weeks” are provided. Another possibility is to explore estimates at different stages of the project lifecycle, to see if the effect of anchors diminishes as projects are more and more detailed. The experimental materials are available to other researchers who wish to repeat the experiment, by contacting the authors.

There are several things that can be done to compensate for anchoring biases. Ideally, estimators should be shielded from anchors. However, this is not always feasible. Estimators should be aware that anchors may bias their own results. Unfortunately, previous studies have shown that anchoring effects take place even when participants are forewarned [25].

Giving wide estimation intervals would help to compensate for the optimism in our calculations: Boehm [4] indicates that confidence ranges of about 50% or 60% are adequate at early project stages, and estimators should resist the temptation to narrow their estimates. Finally, some development lifecycles are riskier than others because of the weight they give to deadlines, and this study is further evidence that lifecycles such as the spiral model or incremental development are safer than others like the waterfall model.

It is interesting to note that our software estimation exercise does not have a right answer. Even if the project was developed, it may be true that project goals are partially set based on estimates, and that a low estimate would produce a smaller product than a high estimate. If that is the case, the power of a seemingly innocuous anchor can shape a project as forcefully as its specifications.

Anchoring and adjustment biases may not be the biggest problem of software estimation. Considering that estimation is frequently done irrationally, that estimation processes tend to resemble bargaining matches, and that accuracy expectations of initial estimates are unreasonable, there are more factors involved in flawed estimations than a misleading anchor. But the need to consider the effect of anchoring on estimates is nonetheless important if we intend to treat software estimation as anything more than guesswork.

8. ACKNOWLEDGMENTS

We would like to thank the people who volunteered to this study for their invaluable participation; and to Eric Yu, Greg Wilson and Björn Regnell for their insights and comments on this work. Funding was provided by NSERC and Bell University Labs (BUL).

9. REFERENCES

[1] Aranda, J., Easterbrook, S. *Anchoring and Adjustment in Software Estimation – Experiment Documentation Package*, 2005. Available at: http://www.cs.toronto.edu/~jaranda/anchoring/package.html

[2] Aronson, E., Wilson, T. D., and Akert, R. M. *Social Psychology.* Prentice Hall, 4th Ed., 2002.

[3] Baird, B. *Managerial Decisions Under Uncertainty.* Wiley, 1989.

[4] Boehm, B. *Software Engineering Economics.* Prentice Hall, 1981.

[5] Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., and Selby, R. Cost models for future software life cycle processes: COCOMO 2.0. *Annals of Software Engineering, Special Volume on Software Process and Product Measurement* (1995).

[6] Brown, N. R., and Siegler, R. S. The role of availability in the estimation of national populations. *Memory and Cognition, 20* (1993), 406-412.

[7] Chapman, G. B., and Bornstein, B. H. The more you ask for, the more you get: Anchoring in personal injury verdicts. *Applied Cognitive Psychology, 10,* 6 (1996), 519-540.

[8] DeMarco, T. *Controlling Software Projects.* Prentice Hall, 1982.

[9] DeMarco, T., and Lister, T. *Peopleware.* Dorset House, 2nd Ed., 1999.

[10] Dolado, J. J. On the problem of the software cost function. *Information and Software Technology, 43* (2001), 61-72.

[11] Helmer, O. *Social Technology.* Basic Books, 1966.

[12] Hihn, J., and Habib-agahi, H. Cost estimation of software-intensive projects: A survey of current practices. *International Conference on Software Engineering* (1991), 276-287.

[13] Hill, J., Thomas, L. C., and Allen, D. E. Experts' estimates of task durations in software development projects. *International Journal of Project Management, 18,* 1 (2000), 13-21.

[14] Jeffery, R., Rune, M., and Wieczorek, I. A comparative study of two software development cost modeling techniques using multi-organizational and company-specific data. *Information and Software Technology, 42,* (2000), 1009-1016.

[15] Jones, C. *Applied Software Measurement.* McGraw Hill, 1996.

[16] Jørgensen, M. A review of studies on expert estimation of software development effort. *The Journal of Systems and Software, 70,* (2004), 37-60.

[17] Jørgensen, M., Teigen, K. J., and Mølokken-Østvold, K. Better sure than safe? Overconfidence in judgement based software development effort prediction intervals. *The Journal of Systems and Software, 70,* (2004), 79-93.

[18] Jørgensen, M., and Sjøberg, D. The impact of customer expectation on software development effort estimates. *International Journal of Project Management, 22,* (2004), 317-325.

[19] Jørgensen, M., and Sjøberg, D. Impact of effort estimates on software project work. *Information and Software Technology, 43,* (2001), 939-948.

[20] Kemerer, C. F. An empirical validation of software cost estimation models. *Communications of the ACM, 30,* 5 (1987).

[21] Mussweiler, T., and Strack, F. The semantics of anchoring. *Organizational Behavior and Human Decision Processes, 86,* 2 (2001), 234-255.

[22] Northcraft, G. B., and M. A. Neale. Experts, amateurs and real estate: An anchoring and adjustment perspective on property pricing. *Organizational Behavior and Human Decision Processes, 39,* 1 (1987).

[23] Shepperd, M., and Schofield, M. Estimating software project effort using analogies. *IEEE Transactions on Software Engineering, 23,* 12 (1997).

[24] Tversky, A., and Kahneman, D. Judgment under uncertainty: Heuristics and biases. In *Judgment under uncertainty: Heuristics and biases; Kahneman, D., Slovic, P., and Tversky, A. (Eds.)* Cambridge University Press, 1982.

[25] Wilson, T. D., Houston, C. E., and Brekke, N. A new look at anchoring effects: Basic anchoring and its antecedents. *Journal of Experimental Psychology: General, 125,* 4 (1993), 384-402.

[26] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. *Experimentation in Software Engineering: An Introduction.* Kluwer Academic Publishers, 2000.