Contrasting Ideal and Realistic Conditions as a Means to Improve Judgment-based Software Development Effort Estimation

Magne Jørgensen
Simula Research Laboratory & Institute of Informatics, University of Oslo

Abstract:

Context: *The effort estimates of software development work are on average too low. A possible reason for this tendency is that software developers, perhaps unconsciously, assume ideal conditions when they estimate the most likely use of effort. In this article, we propose and evaluate a two-step estimation process that may induce more awareness of the difference between idealistic and realistic conditions and as a consequence more realistic effort estimates. The proposed process differs from traditional judgment-based estimation processes in that it starts with an effort estimation that assumes ideal conditions before the most likely use of effort is estimated.*

Objective: *The objective of the paper is to examine the potential of the proposed method to induce more realism in the judgment-based estimates of work effort.*

Method: *Three experiments with software professionals as participants were completed. In all three experiments there was one group of participants which followed the proposed and another group which followed the traditional estimation process. In one of the experiments there was an additional group which started with a probabilistically defined estimate of minimum effort before estimating the most likely effort.*

Results: *We found, in all three experiments, that estimation of most likely effort seems to assume rather idealistic assumptions and that the use of the proposed process seems to yield more realistic effort estimates. In contrast, starting with an estimate of the minimum effort, rather than an estimate based on ideal conditions, did not have the same positive effect on the subsequent estimate of the most likely effort.*

Conclusion: *The empirical results from our studies together with similar results from other domains suggest that the proposed estimation process is promising for the improvement of the realism of software development effort estimates.*

Keywords: Effort estimation, expert estimation, human judgment

1. Introduction

According to published surveys, most software projects are based on estimates that are too low (Jenkins, Naumann et al. 1984; Phan, Vogel et al. 1988; Bergeron and St-Arnaud 1992; Heemstra 1992; Sauer and Cuthbertson 2003; Moløkken-Østvold, Jørgensen et al. 2004; Yang, Wang et al. 2008). These surveys typically report that the average effort overrun is about 30%. There is no convincing evidence to suggest that there has been a systematic improvement in estimation accuracy or increase in bias over time[1]. Neither are there evidence to support that the problem of inaccurate and biased estimates is removed with the use of formal estimation models instead of the use of expert judgment (Aranda and Easterbrook 2005; Jørgensen 2007). Possible reasons for the lack of benefit from formal effort estimation models in this field are that important input to the formal estimation models is judgment-based and that essential relationships are not sufficiently stable and general to enable robust estimation models (Dolado 2001; Jørgensen 2004). There is, however, some evidence to suggest that there are contexts that favor the use of one estimation method over another (Shepperd and Kadoda 2001; Menzies, Zhihao et al. 2006), that some estimators are more realistic than others (Jørgensen, Faugli et al. 2007), and that there are situations in which the estimates are unbiased or even biased towards effort estimates that are too high (Gray, MacDonnell et al. 1999). Strategies that have been evaluated and found to reduce, but not remove, the bias towards effort estimates that are too low are the use of pessimistic scenarios (Newby-Clark, Ross et al. 2000), better use of historical data (Roy, Mitten et al. 2008), and the use of estimators with highly relevant development experience (Jørgensen and Grimstad 2008). Interestingly, all the above strategies for removing bias have in common that they may increase the awareness of the difference between realistic and idealistic conditions. Pessimistic scenarios may increase the awareness of what typically goes wrong in software projects. Historical data may remind the estimator of realistic scenarios for similar tasks. More development experience may make it more likely that estimator will be aware of the complexities and risks of the development work. This may be a significant finding, because people frequently have difficulty in separating idealistic from realistic assumptions when making predictions, as has been reported in numerous studies (Konecni and Ebbesen 1976; Henry and Sniezek 1993; Busby and Payne 1999; Newby-Clark, Ross et al. 2000; Pezzo, Pezzo et al. 2006). This difficulty may be an important reason for the tendency towards underestimation of software development effort. Realistic estimates of software development effort do not necessarily follow from requests to be realistic, but rather from processes that enable the estimators to better separate realistic conditions from pessimistic or idealistic ones.

This paper proposes a process for judgment-based effort estimation (expert estimation) consistent with the above findings. The process assumes that increased awareness of the difference between idealistic and realistic conditions is useful to achieve more accurate effort estimates. Although the proposed process is designed to be used in judgment-based effort estimation processes, such as work-break down estimation processes (Tausworthe 1980), the steps may potentially also be useful to ensure realistic judgment-based input to model-based effort estimation.

[1] The huge improvement in estimation accuracy from 1994 to the present day, as claimed by the Standish Group in their Chaos Reports is, as far as we can evaluate, not trustworthy. See our critique of the Chaos Report in Jørgensen, M. and Moløkken-Østvold, K. (2006). "How large are software cost overruns? A review of the 1994 CHAOS report." Information and Software Technology **48**(4): 297-301..

The remaining part of the paper is organized as follows: Section 2 describes the proposed estimation process and its motivation. Section 3 describes three studies evaluating the proposed estimation process. Section 4 discusses the results in light of other results on human judgment and exemplifies how the process may be integrated into common judgment-based software development effort estimation methods. Section 5 discusses limitations of the studies. Section 6 concludes.

2. A Two-Step Process for Judgment-based Effort Estimation

Well-documented cognitive and motivational mechanisms potentially contributing to idealistic assumptions in situations where the intention is to be realistic include: i) The cognitive difficulty in separating what we want to be and what is more likely to be the outcome in terms of software project effort usage and presence of problems, i.e., "wishful thinking" (Harvey 1992), ii) The tendency to over-rate how much in control of the outcome we are, i.e., "illusion of control" (Langer 1975), iii) The motivation to present estimates consistent with an image of ourselves as more efficient and less error prone than we really are to avoid the so-called "cognitive dissonance" (Festinger 1957), and, iv) The optimism-inducing effect of planning step-by-step what has to be done, i.e., the optimism caused by "looking forward" (Kahneman and Lovallo 1993).

The potential presence of these mechanisms motivates the two main research questions addressed in this paper:
RQ 1: Are judgment-based software development effort estimates requested to reflect realistic conditions likely to be based on idealistic assumptions?
RQ 2: Would a process explicitly asking for effort estimates assuming ideal conditions before asking for effort estimates assuming realistic conditions improve the accuracy of judgment-based effort estimates?

The systematic tendency towards under-estimation in software development suggests a confirmatory answer on RQ 1. Furthermore, if there is an insufficient separation of ideal and realistic conditions in effort estimation, it is, as argued earlier, not unreasonable to expect that making the estimators more aware of this difference will lead to more realistic effort estimates of most likely effort, i.e., that the answer on RQ 2 will be confirmatory as well.

The process we propose and evaluate in order to answer RQ 1 and RQ 2 is a simple two-step process which we believe can easily be integrated into most judgment-based effort estimation processes. An integral part of the proposed process is the concept of "ideal effort". Ideal effort may for example be defined as the effort needed assuming that the work is completed without disturbance, full productivity all the time and no major problems. Ideal effort is in many ways similar to the concept of "ideal days" in agile estimation (Cohn 2006). An essential difference to ideal days in agile estimation is, however, that we use ideal effort only as a contrast to realistic (most likely) use of effort, while the number of ideal days is the final result of the estimation process used in the planning of agile software projects. An assumed implication of our use of ideal effort as an intermediate step and not as the end result is that a consistent interpretation of "ideal" is not essential as long as the understanding of ideal effort enables the estimator to contrast what he or she considers to be the effort usage in ideal conditions with the effort usage in realistic conditions, i.e., the most likely use of effort. This way we may avoid the frequently reported challenge in agile estimation related to "my ideal days are not your ideal days"(Cohn 2006).

The proposed estimation process assumes that the estimator has read and understood the software requirement, preferably conducted some risk analysis and is ready to provide the estimates of the effort of the project as a whole or per activity, user story, feature, use case, etc. Instead of applying the traditional one-step process, where the estimator is requested to provide the most likely use of effort directly, we propose the use of a two-step process emphasizing the contrast between ideal and most likely use of effort:

Step 1: Request the developer to assume that the development is completed under ideal conditions and to estimate the use of effort under these conditions. The description of the ideal condition should ensure that it is meaningful to contrast ideal with typical conditions. This implies that the described ideal conditions should deviate substantially from typical conditions, but not so much that the ideal scenario cannot be used as meaningful reference point.

Step 2: Remind the estimator of the difference between ideal and realistic conditions and, then, request the developer to provide an estimate of the most likely use of effort (the realistic use of effort). The reminder should be sufficiently strong to trigger an estimation process contrasting ideal and typical use of effort.

3. The Empirical Studies

The three empirical studies described in this section compare the judgment-based estimates of most likely effort produced by the proposed two-step estimation process with those produced by the traditional one-step process. To test the robustness of the proposed estimation process, we evaluate it using different formulations of ideal conditions, different reminder formulations and different estimation tasks in the three studies. In addition, we test whether the use of the probabilistic thinking-based concept of “minimum effort”, described as the effort usage only 5% likely to underrun, yields the same effect as produced with the presumably more scenario thinking-based concept of ideal effort.

The progress in results from Study A to Study C is as follows: Study A provides the first evidence in support of that the proposed estimation process lead to higher and more realistic effort estimates than the traditional estimation process. This study also reports that there is not much difference between effort estimates assuming idealistic and realistic conditions, i.e., that many developers seem to think too idealistically when estimating most likely effort. Study B replicates the results from Study A in another domain and with instructions assuming even more idealistic conditions than in Study B. Study B, in addition, reports that the use of a probabilistically defined minimum effort has not the same effect as the use of ideal effort. This supports the assumption that it is the contrast between ideal and realistic assumptions and not, for example, a mechanical upwards adjustment from a value that must be lower than the estimated most likely effort that is the underlying mechanism of the observed effect. Study C replicates the results in a context where the software developers apply a user story-based estimation process and estimated under conditions not very different from their ordinary estimation conditions.

3.1 Study A

3.1.1 The Study Design

Fifty-three software developers attending a seminar on effort estimation participated in Study A. All the developers estimated the effort that they thought was required to develop and test a small web-based software application. The specification was as follows (translated from Norwegian):

"A shoe-vendor expects a high number of requests for information and advices on a trade fair. In order to automate parts of the advices and answers the vendor wants to develop a software program that asks a potential jogging shoe buyer about weight in kg, the typical running surface, whether the shoes are for training or competition, etc. and then gives a recommendation about which shoe type that is suitable, ranked by price. The recommendations should be based on approx. 10 questions to the potential jogging shoe buyer and around 40 rules set by the shoe manufacturer. The rules will be of the type "the shoes xx1 and xx2 are not suitable for persons over 80 kg" and "the shoes yy1-yy5 suitable only for asphalt." There is a total of approx. 20 types of jogging shoes. The system will only be used on this trade fair and you can hard code the rules and assume that there will be no future extensions. The program will however be used by many novice users, some of them with little expertise in the use of computer systems, and should be very robust against input errors and provide easily understandable error messages when the input is incorrect. You choose the programming language and technology you like. Assume that you should do all development and testing yourself."

The developers were divided randomly into two groups: IDEAL-ML (n=26) and ML-IDEAL (n=27). The developers in the IDEAL-ML group were provided the above specification and then instructed to estimate the number of work-hours they would, assuming ideal conditions, need to develop and test the system. As part of the estimation instructions, we described how they should interpret effort usage under ideal conditions. This was described as: "… *the number of work-hours you would need to develop and test the software assuming that you can work concentrated, without disturbance and be fully productive."*

When the developers had completed the estimation work given ideal conditions, they were given the following information and instructions:

"In reality, there will sometimes be disturbance and other events that make it difficult to be fully productive all the time. Number of work-hours given ideal conditions will consequently seldom be the same as the number of work-hours given realistic conditions. What do you think is the most likely number of work-hours needed to complete the development and testing of the software application?"

Following the above instructions, the participants in the IDEAL-ML group estimated the most likely effort.

The developers in the ML-IDEAL group completed similar estimation work in the opposite direction. First, they were asked about the number of work-hours they most likely would need to complete the development and testing of the specified software applications. When that part was completed, they estimated the effort under ideal conditions based on the following instructions:

"Assume now that you can work concentrated, without disturbance and be fully productive. Estimate the number of work-hours required to develop and test the software application given this assumption of ideal conditions."

We do not know how much effort each of the developers would actually spend on completing the development work. This would probably vary quite a lot dependent on, among other things, the developer's expertise, interpretation of usability requirements and choice of technology. On a previous occasion, we had asked a company with extensive experience in the same type of application to estimate how much effort they would have needed. Using comparison with several similar projects and a thorough estimation process as a basis, this company estimated that the work would require about 100 work-hours. The average developer participating in the current study would hardly have a greater amount of relevant experience and skill than the developers in the company we asked to provide an independent estimate. Using the independent estimate as a basis, we argue that estimates much lower than 100 work-hours would, on average, be too low, i.e., indicate a bias towards over-optimistic effort estimates.

The software professionals assessed their own development competence for the specified software. Only the forty-two software professionals who assessed their knowledge to be "satisfactory" or better were included in the analysis. This resulted in group sizes of twenty-two (IDEAL-ML) and twenty (ML-IDEAL).

3.1.2 The Results

The effort estimates provided by the software developers are given in Figure 1. Figure 1 shows that the estimated most likely effort is much higher in the IDEAL-ML than in the ML-IDEAL group. The median estimate of the most likely use of effort in the IDEAL-ML group is 105 work-hours, whereas in the ML-IDEAL group it is only 51 work-hours. A one-sided Kruskal-Wallis test of equality of the median estimates of most likely effort of IDEAL-ML and ML-IDEAL gives the p-value 0.1. Notice also that the median most likely effort estimate of the IDEAL-ML group is close to the estimate provided by the independent company, i.e., 105 vs 100 work-hours.

Figure 1: The Estimates of Study A

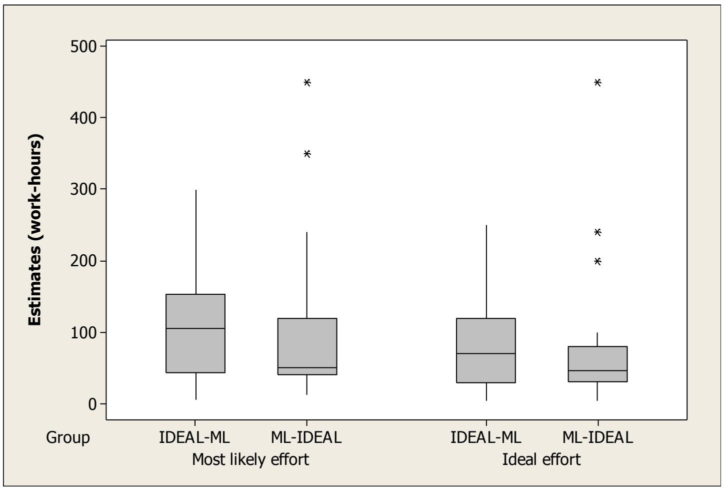
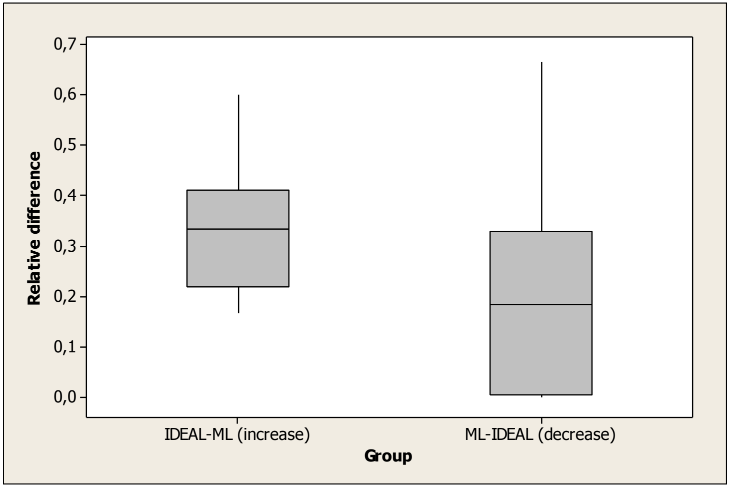


Figure 2 presents the relative increase from ideal to most likely effort of the developers in the IDEAL-ML group and the relative decrease from most likely effort to ideal effort for those in the ML-IDEAL group. Both the decrease and the increase are calculated as: *(most likely effort – ideal effort)/most likely effort*. If the explanation for higher estimates of most likely effort when starting with the estimation of ideal effort is an increased ability to separate ideal and realistic conditions, we would expect a greater increase from ideal to most likely effort than the corresponding decrease from most likely to ideal effort. As can be seen, Figure 2 confirms this expectation to a significant extent. While the median increase from ideal to most likely effort is 33%, the corresponding median decrease from most likely to ideal effort is only 13%. A one-sided Kruskal-Wallis test of equality of the median relative differences (relative increases and decreases) gives the p-value 0.01. An analysis of the absolute difference between median ideal and most likely effort estimates gives similar results. Those in the IDEAL-ML group had a median increase of 35 work-hours, while those in the ML-IDEAL group had a median decrease of only 10 work-hours. A one-sided Kruskal-Wallis test of equality of the median absolute difference in estimates gives the p-value 0.004. Another observation that indicates that the estimates of the most likely effort given by those in the ML-IDEAL group included idealistic assumptions is the finding that as many as 25% of them chose not reduce their estimate when subsequently instructed to estimate the ideal effort! We interpret this as suggesting that, when they were instructed to estimate the ideal effort and began this estimation task, they became aware that their estimate of most likely effort was, in reality, an estimate of ideal effort. This contrasts with the observations that all the developers in the group that started by estimating the ideal effort (the IDEAL-ML group) increased their effort estimates when they were afterwards instructed to estimate the most likely effort.

Figure 2: Relative Difference in Estimates of Study A



3.2 Study B

3.2.1 The Study Design

Study B was designed to test the robustness of the results in Study A and to determine whether the same beneficial effect could be produced by replacing ideal effort with minimum effort. We hypothesized that starting with minimum effort would lead to similar effects if either: i) the change from estimating the minimum to estimating the most likely effort resulted in the developers becoming more aware of their idealistic assumptions, in the same way that they became aware of these assumptions when they were asked to change from estimating the ideal to estimating the most likely effort, or ii) the effect found in Study A was simply due the estimators believing that the most likely effort should be different from (in this case, substantially larger than) the ideal or minimal effort. This behavior is not necessarily based on an increased awareness of the difference between ideal and realistic assumptions; they can also be based on a feeling that there should be some difference between the ideal or minimum and the most likely use of effort. The examination of an estimation process starts that with the estimation of minimum effort is also of interest because many software companies already use a process providing input to project planning that is based on predicting minimum-maximum effort intervals (Jørgensen, Teigen et al. 2004). If minimum effort had the same effect as ideal effort, the process may therefore be easier to integrate in existing processes.

A total of 98 software developers attending two different seminars on effort estimation participated in the study. There was no overlap with the participants in Study A. The participants estimated the total effort to complete: i) the photocopying of 25 copies of the first 113 pages of the book "Software Engineering" by Ian Sommerville, zooming each double page of the book to A4 format, ii) the punching of holes in each of the copies, and iii) the insertion of the copies into ring binders with index dividers separating the five book chapters that comprised the first 113 pages. This work has similarities with software development tasks, in that there are risks related to human error and technical problems. Before we analyzed the results, we conducted a test of the effort required to do the work and found that, with only minor technical problems and minor human errors, we needed about 140 minutes to complete the specified work, when using a modern, medium-fast copying machine that made copies at about 20 pages per minute.

The participants were divided into three groups, IDEAL-ML (n=34), MIN-ML (n=30) and ML-IDEAL (n=34), all of which estimated the total effort of the photocopying/punching/ring binding task described above. The participants in the IDEAL-ML group first estimated the effort (in minutes) they would need to complete the work given ideal conditions. Ideal conditions were in this study described as "*… the work is completed without disturbance, you are able to work with full productivity all the time, and no problems occur.*" Notice that these ideal conditions are even more idealistic than those in Study A, due to the inclusion of the condition that "*no problems occur*". When estimating the most likely effort, they were instructed to: "*Assume a normal situation, i.e., what you think is realistic (most likely), and estimate how much effort you would need to complete the work.*"

The participants in the MIN-ML group were instructed to estimate the minimum effort through the following question: "*What is the minimum effort you will need to complete this work?*" Minimum effort was described as "*.. the effort usage it is only 5% likely (i.e., very unlikely) to underrun.*" This description is close to the usual description of the lower boundary of effort prediction intervals, e.g., as implemented in the PERT method, see Moder et al. (1995). As documented in (Jørgensen, Teigen et al. 2004), changing the confidence levels does typically not affect the judgment-based assessment of minimum or maximum effort very much, which means that it is not likely to be important whether we chose 5% or a lesser value as the confidence level in this study. When completing this estimate, they were, as those in the IDEAL-ML group, instructed to estimate the most likely effort.

The participants in the ML-IDEAL group were first asked to estimate the most likely effort. Then, they estimated the effort under ideal conditions based on the instructions: "*Assume that the* *work is completed without disturbance, you are able to work with full productivity all the time, and no problems occur. How much effort do you think you need to complete the work given such idealistic assumptions?"*

3.2.2 The Results

The effort estimates provided by the participants are given in Figure 3. As in Study A, the participants who started by estimating the ideal effort provided much higher estimates (median of 120 minutes) than those who started by estimating the most likely effort (median of 75 minutes). A Kruskal-Wallis, one-sided test of equal median values gives p=0.01. Starting by estimating the minimum effort did not have the same effect on the estimate of most likely effort. In fact, the participants who started by estimating the minimum effort gave slightly lower (no significant difference) estimates of the most likely effort than those who started by estimating the most likely effort (median of 63 vs 75 minutes). A comparison with the effort we used when actually completing the work (140 minutes) suggests that the developers who started by estimating the ideal effort provided, on average, the most realistic effort estimates.

The estimated ideal effort of those in the IDEAL-ML group was significantly higher than the estimated minimum effort of those in the MIN-ML group (Kruskal-Wallis, one-sided test of equal median values gives p=0.02; medium of 83 vs 50 minutes). More interestingly, those who started by estimating the most likely effort provided an estimate of most likely effort that did not differ significantly from the estimates of ideal effort provided by those in the IDEAL-ML group (median of 75 vs 83 minutes). It would seem that the estimates of most likely effort included idealistic assumptions to the same extent as the estimation of ideal effort.

Figure 3: The Estimates of Study B

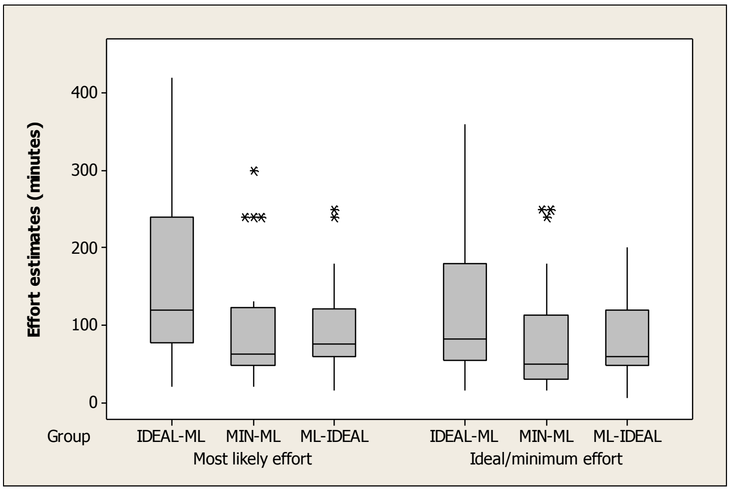
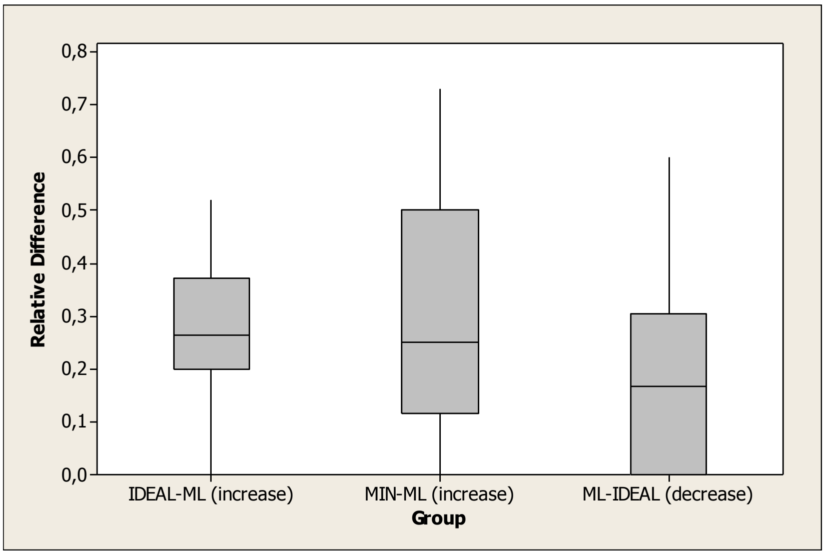


Figure 4 compares the differences in relative increase from the estimates of ideal effort to those of most likely effort for those in the IDEAL-ML group, the relative increase from the estimates of minimum effort to those of most likely effort for those in the MIN-ML group, and the relative decrease from the estimates of most likely effort to those of ideal effort for those in the ML-IDEAL group. The relative increase or decrease is measured as: *(most likely effort – ideal effort)/most likely effort* or as *(most likely effort – minimum effort)/most likely effort*. Figure 4 shows a median relative increase from ideal to most likely effort estimates of 26% and a median relative decrease from most likely to ideal effort estimates of 17%. These results are similar to those gained from Study A. A one-sided Kruskal-Wallis test of equal median values gives p=0.01. The median increase from minimum to most likely effort estimates was 25%, which is about the same as the increase from ideal to most likely effort estimates. However, this similarity hides a greater willingness to increase the absolute number of estimated minutes in the IDEAL-ML group than in the MIN-ML group. Those in the IDEAL-ML group increased their estimates by a median of 30 minutes, while those in the MIN-ML group increased their estimates by a median of only 18 minutes. A one-sided Kruskal-Wallis test of equal median values of these two groups gives p=0.05. The median absolute decrease of the effort estimates of those in the ML-IDEAL group was, as a comparison, 15 minutes, which is much lower than the corresponding increase of those in the IDEAL-ML group. Taken together, these results support the previous findings that starting by estimating the most likely effort leads to estimates that are, to a great extent, based on idealistic conditions.

Figure 4: Relative Difference in Estimates of Study B



3.3 Study C

3.3.1 The Study Design

Study C was designed to evaluate the findings found in Studies A and B in a more realistic software development effort estimation context. For this, we used the concept of “ideal effort” as defined in the context of agile software development. In agile software development, ideal effort is typically described as the effort required assuming no interruptions, no unexpected events, and full productivity all the time, i.e., the same definition as we used in Study A. The requirements of the application to be estimated were specified as so-called “user stories” (Cohn 2006). The user stories were divided into three releases. The three estimation processes to be used by the developers were the IDEAL-ML, ML and SP process. The SP process is based on the estimation of “story points” and is not relevant for the analysis of this paper.

The participants using the IDEAL-ML process first estimated the ideal effort of each of the user stories and activities described in the specification[2]. The instructions per release were as follows: “*Estimate the ‘ideal effort’ you would need to develop and test each of the user stories included in this release. ‘Ideal effort’ estimation is based on the concept of ‘ideal work-hours’. When estimating the number of ‘ideal work-hours’ of a user story you should, as is common in agile estimation, assume that you are able to work without interruptions, that there are no unexpected events and you are fully productive all the time. Estimate the user stories in the sequence you find most natural. You are, of course, allowed to divide each user story into activities/tasks to derive the estimates. Activities not naturally covered by the estimates per user story (if any), should be described and estimated in the row denoted “Other effort”. “Other effort” may for example include testing activities and other work not naturally belonging to one of the user stories.”*

When this part of the estimation work was completed, the participants estimated the most likely effort based on the following instruction: *“In reality, you may experience interruptions, unexpected events and you will not be fully productive all the time. The number of work-hours you most likely need will therefore normally be higher than the ideal number of work-hours. Estimate the effort you most likely would need to develop and test each of the user stories included in this release. You are, of course, free to look back on your estimates of ideal number of work-hours.”*

The participants using the ML process started with the estimation of most likely effort. The instructions were as follows: *“Estimate the effort you most likely would need to develop and test each of the user stories included in this release. Estimate the user stories in the sequence you find most natural. You are, of course, allowed to divide each user story into activities/tasks to derive the estimates. Activities not naturally covered by the estimates per user story (if any), should be described and estimated in the row denoted “Other effort”. “Other effort” may for example include testing activities and other work not naturally belonging to one of the user stories.”*

Twenty-one competent developers were selected by the management in a Polish company to participate in the study. The company was paid ordinary fees for their estimation work. The estimation work was completed over three consecutive days, with one release of user stories estimated per day. The developers estimated their own work effort and were allowed to assume that the development technology they knew best would be used, as long as it was suited for the development work. The developers were divided into three equally sized groups, and used estimation processes as specified in Table 1. As an illustration, Table 1 shows that those in Group 2 estimated Release 2 on Day 2, using the IDEAL-ML estimation process.

Table 1: The Estimation Processes per Group and Day

Group	**Day 1 (Release 1)**	**Day 2 (Release 2)**	**Day 3 (Release 3)**
1	IDEAL-ML	ML	SP
2	SP	IDEAL-ML	ML
3	ML	SP	IDEAL-ML

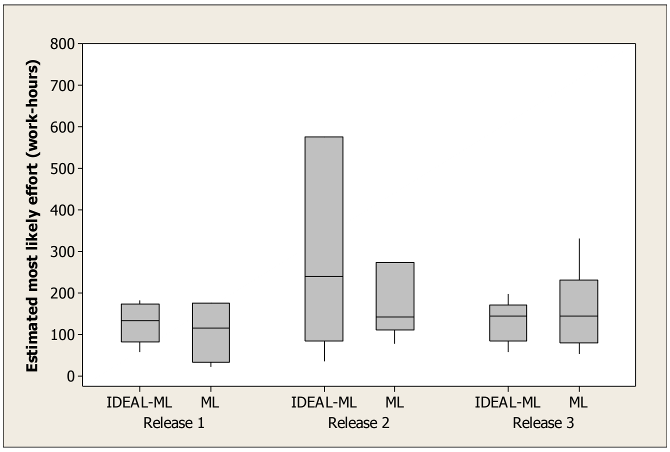
[2] The requirement specification will be sent upon request to magnej@simula.no to interested readers.

The actual effort needed to develop the specified functionality (sum of all three releases) was about 600 work-hours. The developers who actually programmed and tested the application were highly experienced, i.e., at least as experienced as the developers participating in this experiment. Median effort estimates lower than 600 work-hours would, we believe, suggest a bias towards too low estimates.

3.3.2 The Results

The effort estimates of most likely effort provided by the software developers are shown in Figure 5. As in the two previous studies, the developers who started by estimating the ideal effort provided higher median estimates of the most likely effort. A comparison of the sum of the median estimates of the three releases per estimation process reveals that the IDEAL-ML estimates of most likely effort are 29% higher than the ML estimates of most likely effort (median of 520 and 403 work-hours). Above, we indicated that median estimates of lower than 600 work-hours would be over-optimistic. If this is correct, both estimation processes led to effort estimates that were too low, but the IDEAL-ML estimates of most likely effort were less over-optimistic. The low power of the study (only seven developers in each group) means that we cannot expect significant differences. The one-sided Kruskal-Wallis tests per release are p=0.85, p=0.66 and p=0.48. In accordance with principles for replications and meta-studies (Rosenthal 1978; Hallahan and Rosenthal 1996), the results nevertheless provide support for the results yielded by Studies A and B. A statistical meta-analysis of the three studies gives that the combined significance of the difference between the estimated most likely effort when following the proposed and the traditional process is quite strong. Using a Stouffer's z trend[3] (Whitlock 2005), which combine and correct the p-value for sample sizes and the effect directions of independent studies, we get a p-value of 0.0022. While this is only slightly lower than the combined p-value of the studies A and B alone (p-value of 0.0026), it supports the argumentation that even studies (such as Study C) with non-significant results contribute to the strength of a finding as long as the result is in the expected direction.

Figure 5: The Estimates of Study C



[3] We treated each release of Study C as a separate experiment and used the tool MetaP (Author: Dongliang Ge, PhD, people.genome.duke.edu/~dg48/metap.php) to combine the results of Study A, B and C. The results should only be interpreted only as a rough prediction of the combined effect as the test is based on several assumptions that are not necessarily fully met.

The sum of the median estimates of the ideal effort was 334 work-hours, which is not much lower than the median estimates of most likely effort when the ML estimation process was applied (403 work-hours). This suggests, as in Studies A and B, that there were a lot of idealistic assumptions included in the estimate of most likely effort when the ML estimation process was applied.

4. Discussion

There is a wealth of studies reporting that people tend to view the future too bright, see for example (Armor and Taylor 2002). As argued in Section 2, there are several theories potentially explaining this tendency, including theories based on cognitive and motivational processes. The estimation process described and evaluated in this paper assumes that the lack of separation of idealistic and realistic conditions is also a problem in the context of effort estimation and that an increase of the awareness of differences these two conditions improves the accuracy of the effort estimates. The results reported in this article confirm these assumptions and give confirmatory answers to Research Questions 1 and 2, as presented in Section 2.

Our results are consistent with results reported in other domains, e.g., (Newby-Clark, Ross et al. 2000; Tanner and Carlson 2009). Of particular interest are the results reported in (Tanner and Carlson 2009). Tanner and Carlson report on a series of experiments in which, when first asked to think idealistically, people became more realistic regarding predictions of their own blood donation behaviour, frequency of weekly exercise, savings discipline, completion time for watching an educational DVD, completion time for writing a report, how many songs they would save on their iPod, and, self-assessment of skills (math, music, and juggling). In the case of Tanner and Carlson's subjects, thinking idealistically was based on the instruction: "*In an ideal world, how often would you donate your blood, exercise, etc.*" The authors state that the main elements of their process are: i) drawing attention to the idealistic standard through requests for prediction given an ideal world, and ii) self-assessment query based on the testing of a more realistic hypothesis than the default idealistic self-assessment. This description is, as we interpret it, in essence the same as our descriptions of the contrasting mechanisms leading to the positive effect of starting the effort estimation process by assuming ideal conditions. Their conclusion further supports the similarity of their and our explanation of the mechanism: "*… the key to more realistic prediction of future behavior lies not in exhorting consumers to ignore the ideal, but in getting them to acknowledge it.*"

Our finding in Study B that that the use of a probabilistically described "minimum effort" did not give the same effect is similar to the finding that a starting with a probabilistically oriented "optimistic estimate" of time usage, described as an 1% fractile, did not impact the best guess estimate in (Byram 1997). The results suggest that requests for and contrasts with ideal effort and minimum effort evoke different mental processes. There are, as far as we know, no evidence-based accounts of how people assess the minimum use of effort for a given confidence level. One possible account is that people estimate minimum effort by first estimating the most likely effort and then calculating the minimum effort as a proportion of that effort or as an amount that contrasts with it. If this account is correct, it would explain why the estimates of most likely effort did not increase following the estimation of minimum effort, whereas they did increase following the estimation of ideal effort. Another possible account is related to the differences between how experience is accessed in probabilistic (minimum effort) and scenario-based (ideal effort) thinking. Previous experience in effort usage of similar software projects is typically not stored in our memory as a distribution of possible outcomes, which would make the experience suitable as input for calculation of probabilistic minimum effort values. It may consequently be difficult to activate previous software development project experience when asked to provide a probabilistically-oriented minimum effort. A good illustration of the problem software developers have to handle probabilistically defined minimum values is provided in (Jørgensen, Teigen et al. 2004), where the average minimum-maximum effort interval width was almost the same regardless of receiving instructions to be 99%, 90%, 75% or 50% confident to include the actual effort in the interval. Scenario-based thinking may, on the other hand, enable a better fit between the request format and how project experience is stored in the memory.

Our finding that going from an ideal to a most likely condition led to more change in the effort estimates than going from a most likely to an ideal condition has, as far as we are aware of, not been reported elsewhere. The finding is, however, a natural consequence of a situation where estimates of most likely effort tend to include idealistic assumptions and an increased awareness of the difference between idealistic and realistic conditions improves the realism of the estimates of most likely effort.

When we proposed the two-step process in Section 2 we assumed that the exact description of ideal conditions did not matter and that a precise description was not required as long as the description was useful for the estimator to contrast with realistic conditions. Our results support this assumption. Clearly, this does not mean that any type of idealistic conditions will work equally well. More studies are needed to examine exactly what level of idealism is optimal to achieve realistic effort estimates.

The proposed two-step estimation process can easily be integrated in most existing judgment-based estimation processes. Assume, for example, the following bottom-up effort estimation process:

1. Collect and understand information relevant for the estimation work.
2. Decompose the work into more manageable elements, such as activities, user stories, use cases and features.
3. Estimate the required effort of each of the decomposed elements.
4. Assess the uncertainty of the effort estimate, either per element and/or of the total estimate.
5. Re-estimate, e.g., after each release, milestone or when receiving feedback suggesting that the previous estimates are inaccurate.

The proposed two-stage estimation process is meant to replace Step 3 in the above sequence in situations where there effort estimates are judgment-based. Judgment-based estimation is the dominating estimation approach in software development contexts, see our review in (Jørgensen 2007). The proposed process may also replace Step 5, but this will depend on the context. If, for example, there are high quality data about the actual productivity on previous releases available, such as in some agile development contexts, it may be natural to use the historical data directly. In contexts where the projects apply agile estimation processes it is possible to base our proposed estimation process upon the concept of ideal days, perhaps already implemented in the project. The only change will be to add a step where most likely use of effort is contrasted with ideal days and to use the estimated most likely effort rather than the ideal days in the planning process.

There may be many other applications of the two-step process of starting with judgments given ideal conditions, with subsequent judgment given realistic conditions. As suggested by the results presented in (Tanner and Carlson 2009), the benefits of this process seems to be quite general and the process applicable to many types of judgment. Judgments that may well benefit from use of the process include the assessment of the delivery date of a software product, the assessment of the effects of changing from one development method or technology to another, judging the benefits of using a software component, the provision of judgment-based input to effort estimation models, and the development of plans for allocating resources.

5. Limitations

The results of all three studies point in the same direction and are consistent with those of previous studies in other domains. In spite of this, there are several limitations that it is important to be aware of when interpreting and using the results:

- The estimation situations, especially those in Studies A and B, deviate to some extent from those in typical software development projects. In (Jørgensen and Grimstad in press) we compared the effort estimation research results when laboratory (artificial), rather than field, settings are used. We found that estimation bias results obtained in laboratory settings were also obtained in field settings, but that the effect sizes typically were lower. The results in (Jørgensen and Grimstad in press) do not, of course, guarantee that the effects described in this article are present in more typical software development field settings. Nevertheless, they do point to the relevance of studying processes in laboratory settings to establish that different estimation processes do have different effects. To know more about size of the effects in different contexts, and to determine their utility, e.g., how much starting the estimation process with idealistic thinking would help in typical field settings, there is a need for field studies. The presented results should mainly be interpreted as relevant for contexts similar to those we have studied, i.e., in situations with a focus on estimation of small projects and activities, and as an indication of that the proposed estimation process is promising.
- In field settings, the estimators will use an estimation method repeatedly and possibly learn from and adapt the use of it. Our experiments mainly report on the first-time use of the proposed process and do not offer much insight into long-term effects. The effect of repeated use of the proposed process should also be a subject for further studies.
- We excluded the software professionals in Study A who assessed their development skill as being “not satisfactory”, we selected a task in Study B where it was likely that all participants had some experience, and, we had explicit skill criteria for accepting the developers who participated in Study C. In spite of these precautions, we cannot be sure that the estimation skills of the participants were as good as they would have been in typical field settings. On the one hand, this may not be essential, because less than competent developers would be likely to be distributed equally across the different groups. On the other hand, it may be that a higher level of expertise would lead to the use of the proposed estimation process having weaker effects. As reported in (Jørgensen and Grimstad 2008), the estimates of more experienced developers may be more robust towards changes in the estimation process than those of inexperienced developers.
- The developers did not complete the tasks they estimated. We do consequently not know how much each individual effort estimate deviates from the their actual use of effort. Optimally, we

would have let all developers complete the tasks and calculated accuracy improvement of difference estimation approaches based on the actual deviation between the estimated and the actual effort. This would however have been a very costly approach and in practice exclude the study of software professional estimating the effort of other than very small projects. The approach taken in the studies included in this paper is instead that we use evidence from different sources to argue that it is highly likely that the *average* estimate of one group is less over-optimistic than that of another group. The sources used for this purpose are: i) The estimates are substantially lower than those of companies with highly relevant competence, ii) The estimates are substantially lower than the actual effort spent by other companies or people, and iii) The well-documented general tendency towards over-optimistic estimates in the software industry when estimating most likely use of effort. In total, we believe this constitutes a sufficiently strong argumentation for the claim made, i.e., the claim that the proposed estimation approach is promising.

6. Conclusion

We report evidence supporting the hypothesis that estimates of most likely effort frequently are based on idealistic (over-optimistic) assumptions. Previous debiasing methods have to a large extent failed to remove this tendency towards over-optimism in effort estimation. This motivated us to evaluate the effect of a two-step estimation process, which explicitly requests an estimate of effort under ideal conditions before it requests the estimate of most likely use of effort. The rationale of this two-step process is that it may lead to a stronger awareness of the difference between idealistic and realistic effort usage assumptions and, as a consequence, lead to an improvement of the realism of the estimates. The results from three empirical studies suggest that the proposed process is a promising means to gain more realism in the effort estimation process, i.e., that the process led to more accurate estimates in the studied estimation contexts. The proposed process can, we argue, easily be integrate into existing estimation processes and is, we argue, a promising candidate to improve the accuracy of effort estimates in field settings. The proposed process has, however, not yet been evaluated in field settings. To assess whether the proposed process will lead to similar effects in field settings with larger projects, we need more studies. Another topic for future studies is to better understand the underlying cognitive processes leading to different estimates when starting with ideal assumptions. This may lead to important knowledge to better understand when the proposed process is likely to lead to more accurate estimates and when not.

References:

Aranda, J. and Easterbrook, S. (2005). "Anchoring and adjustment in software estimation." Software Engineering Notes **30**(5): 346-355.

Armor, D. A. and Taylor, S. E. (2002). When predictions fail: The dilemma of unrealistic optimism. Heuristic and biases: The psychology of intuitive jdugment. T. D. Gilovich, D. Griffin and D. Kahneman. Cambridge, UK, Cambridge University Press**:** 334-347.

Bergeron, F. and St-Arnaud, J. Y. (1992). "Estimation of information systems development efforts: a pilot study." Information and Management **22**(4): 239-254.

Busby, J. S. and Payne, K. (1999). "Issues of organisational behaviour in effort estimation for development projects." International Journal of Project Management **17**(5): 293-300.

Byram, S. J. (1997). "Cognitive and motivational factors influencing time prediction." Journal of Experimental Psychology-Applied **3**(3): 216-239.

Cohn, M. (2006). Agile estimation and planning. NJ, Pearson Education.

Dolado, J. J. (2001). "On the problem of the software cost function." Information and Software Technology **43**(1): 61-72.

Festinger, L. (1957). A theory of cognitive dissonance. Stanford, CA, Stanford University Press.

Gray, A., MacDonnell, S. and Shepperd, M. (1999). Factors systematically associated with errors in subjective estimates of software development effort: the stability of expert judgment. International Software Metrics Symposium, Boca Raton, Florida, USA, IEEE Comput. Soc, Los Alamitos, CA, USA.

Hallahan, M. and Rosenthal, R. (1996). "Statistical power: Concepts, procedures, and applications" Behaviour Research and Therapy **34**(5-6): 489-499.

Harvey, N. (1992). "Wishful thinking impairs belief-desire reasoning: A case of decoupling failure in adults?" Cognition **45**(2): 141-162.

Heemstra, F. J. (1992). "Software cost estimation." Information and Software Technology **34**(10): 627-639.

Henry, R. A. and Sniezek, J. A. (1993). "Situational Factors Affecting Judgments of Future Performance." Organizational Behavior and Human Decision Processes **54**(1): 104-132.

Jenkins, A. M., Naumann, J. D. and Wetherbe, J. C. (1984). "Empirical investigation of systems development practices and results." Information and Management **7**(2): 73-82.

Jørgensen, M. (2004). "A review of studies on expert estimation of software development effort." Journal of Systems and Software **70**(1-2): 37-60.

Jørgensen, M. (2007). "Forecasting of software development work effort: evidence on expert judgement and formal models." International Journal of Forecasting **23**(3): 449-462.

Jørgensen, M., Faugli, B. and Gruschke, T. (2007). "Characteristics of software engineers with optimistic predictions." Journal of Systems and Software **80**(9): 1472-1482.

Jørgensen, M. and Grimstad, S. (2008). "Avoiding irrelevant and misleading information when estimating development effort." IEEE Software **May/June**: 78-83.

Jørgensen, M. and Grimstad, S. (in press). "The Impact of Irrelevant and Misleading Information on Software Development Effort Estimates: A Randomized Controlled Field Experiment " IEEE Transactions on Software Engineering.

Jørgensen, M. and Moløkken-Østvold, K. (2006). "How large are software cost overruns? A review of the 1994 CHAOS report." Information and Software Technology **48**(4): 297-301.

Jørgensen, M., Teigen, K. H. and Moløkken, K. (2004). "Better sure than safe? Over-confidence in judgement based software development effort prediction intervals." Journal of Systems and Software **70**(1-2): 79-93.

Kahneman, D. and Lovallo, D. (1993). "Timid choices and bold forecasts: A cognitive perspective on risk taking." Management Science **39**(1): 17-31.

Konecni, V. J. and Ebbesen, E. B. (1976). "Distortions of Estimates of Numerousness and Waiting Time." Journal of Social Psychology **100**(1): 45-50.

Langer, E. J. (1975). "The illusion of control." Journal of Personality and Social Psychology **32**(2): 311-328.

Menzies, T., Zhihao, C., Hihn, J. and Lum, K. (2006). "Selecting Best Practices for Effort Estimation." Software Engineering, IEEE Transactions on **32**(11): 883-895.

Moder, J. J., Phillips, C. R. and Davis, E. W. (1995). Project management with CPM, PERT and precedence diagramming. Wisconsin, U.S.A, Blitz Publishing Company.

Moløkken-Østvold, K., Jørgensen, M., Tanilkan, S. S., Gallis, H., Lien, A. C. and Hove, S. E. (2004). A survey on software estimation in the Norwegian industry. 10th International Symposium on Software Metrics, Chicago, IL, IEEE Computer Soc.

Newby-Clark, I. R., Ross, M., Buehler, R., Koehler, D. J. and Griffin, D. (2000). "People focus on optimistic scenarios and disregard pessimistic scenarios when predicting task completion times." Journal of Experimental Psychology: Applied **6**(3): 171-182.

Pezzo, S. P., Pezzo, M. V. and Stone, E. R. (2006). "The social implications of planning: How public predictions bias future plans." Journal of Experimental Social Psychology **42**(2): 221-227.

Phan, D., Vogel, D. and Nunamaker, J. (1988). "The search for perfect project management." Computerworld: 95-100.

Rosenthal, R. (1978). "Combining results of independent studies." Psychological bulleting **85**: 85-193.

Roy, M. M., Mitten, S. T. and Christenfeld, N. J. S. (2008). "Correcting memory improves accuracy of predicted task duration." Journal of Experimental Psychology-Applied **14**(3): 266-275.

Sauer, C. and Cuthbertson, C. (2003). The state of IT project management in the UK 2002-2003. U. o. Oxford.

Shepperd, M. and Kadoda, G. (2001). "Comparing software prediction techniques using simulation." IEEE Transactions on Software Engineering **27**(11): 1014-1022.

Tanner, R. J. and Carlson, K. A. (2009). "Unrealistically Optimistic Consumers: A Selective Hypothesis Testing Account for Optimism in Predictions of Future Behavior." Journal of Consumer Research **35**(5): 810-822.

Tausworthe, R. C. (1980). "The work breakdown structure in software project management." Journal of Systems and Software **1**(3): 181-186.

Whitlock, M. C. (2005). "Combining probability from independent tests: the weighted Z-method is superior to Fisher's approach." J Evol Biol **18**: 1368-1373.

Yang, D., Wang, Q., Li, M. S., Yang, Y., Ye, K., Du, J. and Acm (2008). A Survey on Software Cost Estimation in the Chinese Software Industry. ACM/IEEE International Sympsoium on Empirical Software Engineering and Measurement, Kaiserslautern, GERMANY, Assoc Computing Machinery.