# Software Effort Estimation Terminology: The Tower of Babel

Stein Grimstad, Magne Jørgensen, Kjetil Moløkken-Østvold
*Simula Research Laboratory*
*{steingr,magnej,kjetilmo}@simula.no*

## Abstract

*This paper provides a review of how software development effort estimation terms are used in software engineering textbooks and research papers. We found that the term 'effort estimate' frequently is applied without any clarification of its meaning. It is therefore difficult to determine whether the authors' intended interpretation is an estimate of 'most likely effort', 'planned effort', 'budgeted effort', or something else. This is problematic as these terms are not equivalent and are used for different purposes. The lack of clarity of 'effort estimate' lowers the quality and interpretability of surveys on software effort estimation accuracy, i.e., it is not clear what the estimation accuracy results really mean. This reduces the estimation evaluation and learning possibilities. We suggest guidelines on how to reduce this terminology ambiguity. To the authors' knowledge, this is the first published review of software cost estimation terminology.*

## 1. Introduction

The following two case stories illustrate the importance of precise communication of estimation related information, and motivated the review and recommendations presented in this paper.

*Case story 1*: In 2003, two of the authors performed a survey on project estimation in Norwegian software companies[1]. The goal was to get an in-depth understanding on estimation practice and to examine factors with impact on effort estimation accuracy. The basis for the estimation accuracy measurement was a comparison of the actual use of effort with the estimated most likely effort provided in the planning stage of the project, i.e., how much effort does the contractor believe that the project will require, regardless of the price to the customer or the budget. An interesting result was the observation that governmental projects on average had significantly higher deviations between estimated most likely efforts and actual efforts than private projects[2]. This observation made the headlines in Norway's largest morning newspaper. The day after the results were presented, the front page of the newspaper stated '*Yearly overruns of 6 billions [Norwegian Kroner] in governmental IT-projects*'[3]. The debate that followed was heated, and culminated with the research results being discussed in the Norwegian parliament. In particular, there were members of parliament that applied our results as evidence of a governmental waste of money on IT-projects. Our results, however, did not say anything about the *customers'* budget overruns or losses. Neither had we studied the software providers' budget overruns or losses. What we had studied was the overruns related to what the software providers believed was the *most likely effort* of a project. The newspaper article, which was the basis for the debate, did not point out this. Budgeted costs typically include a risk buffer added to the most likely effort. The cost overrun we found may therefore have been much higher than the software organizations' and the customers' budget overrun. A consequence of misinterpretation of the term 'most likely estimate' was that the public discussion mainly focused on whether one should believe the high cost overrun number or not, and much less on how governmental projects could be better managed, i.e., on improvement of their role as software customers.

*Case story 2*: Some time ago, one of the authors was hired as a solution architect of a software project. It was a high risk project for a number of reasons; the functionality to be developed was complex, several stakeholders with conflicting goals were involved and a non-extendable deadline was set. The initial analysis suggested that the project would involve about 40 people and changes had to be made to five systems, all in operation. Our time and effort estimates suggested that it was probable that we could deliver before the deadline, but with small margins. Not surprisingly, during development we ran into trouble and the changes of one of the systems were two weeks delayed. The changes of this system were on the project's critical path and the entire project was therefore two weeks delayed. Moving the deadline was, of course, unacceptable to the customer because this would ruin the announced launch. However. what happened was that we delivered all functionality on time and on budget. How did we manage that? We did it the same way that many other software development teams do in similar situations, i.e., we reduced the amount

of testing. The project went into operations, and luckily only minor failures occurred. How accurate were our estimates? From the outside, i.e., as would have been observed in most estimation surveys, we had only minor effort estimation error and no schedule overrun. In reality, however, the project would have had larger estimation error and a time overrun if it had completed the testing process as planned, i.e., with the promised level of quality. This case story shows that common measurement of effort estimation accuracy may give a misleading picture of the real estimation accuracy and hence a misleading picture of the need for estimation process and project management improvement.

The two case stories indicate that proper communication, interpretation and improvement of estimation accuracy measurements may be a problem when there is no precise use of estimation related terms. That problem motivates the review and guidelines provided in this paper.

Related work is presented in section 2 of this paper. Section 3 further elaborates on the consequences of imprecise use of effort estimation terms. The consequences are illustrated by observations of software projects in a Norwegian software development organization. In Section 4 we review the actual use of effort estimation terminology in popular software engineering textbooks, research papers suggesting estimation guidelines and in estimation surveys. Based on the discussion in Section 3 and the review in Section 4 we provide, in Section 5, recommendations aimed at improvement of use of software effort estimation terminology and hence enabling estimation process improvement. Finally, Section 6 concludes the paper.

## 2. Related work

The problems of the imprecise software cost estimation terminology have been addressed by several leading software engineering researchers.

Kitchenham[4] advises that before you improve estimation processes, you should make sure that you do not have a management problem. The lack of sufficiently precise effort estimation terminology, e.g., to reduce the probability of interpreting an estimate of most likely effort as a budget, is clearly a management problem.

In Waltzing with bears [5], a book about managing risk in software projects, DeMarco and Lister list 'schedule flaw' as the largest risk of software projects. Schedule flaw means to make no distinction between the most optimistic estimate (with virtually no probability of achieving), the goal (that the project aims for), the estimate (the most likely outcome) and the schedule (what the project commits for).

In 'Software Estimation Perspectives' [6], Boehm and Fairley state two important points about software estimation: 1) It is best to understand the background of an estimate before you use it and, 2) It is best to orient your estimation approach to the use that you are going to make of the estimate.

Edwards and Moore [7] raise the question whether an estimate is 'a rough guide' to the cost of a project or 'applying numbers to the detailed project plan'. They conclude that both are estimates, although these meanings of estimates are different with respect to uncertainty, usage and motivation. They later argue that the lack of clear distinction between these two types of estimates in estimation tools explains why estimation tools are not commonly used in the industry.

The 'Parametric Cost Estimating Handbook' [8] points at the importance of the relationship between the estimate and the product being estimated: *This definition [of parametric estimating] establishes the clear linkage between cost and a product's (or end item) technical parameters. Without this linkage, a product cost cannot be effectively defined.*'

In 'Software Engineering' [9], one of the most widely used books in Software Engineering courses, Sommerville addresses the problem of comparing outcomes of different quality when discussing productivity rates in projects. He argues that produced quality as well as functionality must be considered, and concludes that comparison of productivity rate is not meaningful when solutions with different attributes (quality) are produced.

In [10], we explain through comparison with vacation cost estimation and by an industrial case study why similar estimates with similar accuracy can have huge differences in estimation performance. In that paper we exemplify the conflicting goals of different types of estimates; 'most-likely software development cost', 'risk-minded planned development cost' and 'cost-reducing planned development costs'.

## 3. The Importance of Precise Effort Estimation Terminology

Software development effort estimates are the basis for project bidding and planning, both critical practices in the software industry. The consequences of poor budgets and plans can be dramatic. If budgets and plans are too pessimistic, business opportunities can be lost, while over-optimism may be followed by significant losses. The importance of accurate estimates is supported by the Standish Group [11] that, based on their CHAOS reports, concludes that 'reliable estimates' is one of the top ten most important success factors in software projects. It is therefore unfortunate that a comparison of estimation surveys [12] indicates that there has not been much improvement in software cost estimation accuracy over the last 20 years. We believe that one reason for this lack of improvement, on a topic as important as effort

estimation, is the imprecise use of effort estimation terminology. Obviously, most of the inherent problems with effort estimation are not solved with more precise or standardized terminology. We believe, however, that a *necessary* condition for sustainable improvement is a precise use of important terms, because lack of precise estimation terminology easily leads to:

- A mix of values with different purposes, e.g., a mix of values with focus on realism (most likely effort), incentives leading to efficient development work (planned effort), avoidance of budget overrun (budgeted effort), and winning a bid (price-to-win). The lack of separation between these purposes have been found to reduce the realism of estimates [13-15].
- A mix of related values based on different outcomes, e.g., the solution that the estimator assumes is going to be made (the basis for the most likely effort estimates) might differ from the solution actually produced (the basis for the actual results) with respect to functionality (functional parameters) or the level of quality (technical parameters). The relationship between effort and functional and technical parameters is widely acknowledged.
- Improper evaluation, comparison and reporting of effort estimation performance, including lower ability to learn from experience [16, 17].

Recent observations of software projects in a Norwegian software development organization illustrate this. Over a period of two years we logged estimation information of 32 software projects in that company (the logging is part of an on-going study on reasons for estimation errors). As a part of the logging we requested that the person responsible for the estimation documented the estimate of 'most likely effort'. An analysis of the description of how the estimate of 'most likely effort' was derived showed, however, a wide variety of interpretation. In most cases, the estimate was *not* of most likely effort, as we had required. Instead, the effort estimate was typically described as most likely effort added a risk buffer of varying size, i.e., it was interpreted as the planned or the budgeted effort, or it was described as the effort (derived from the price) agreed with the customer. We have logged similar estimation information in other organizations and believe that this is a common pattern [18]. The term 'effort estimate' seems to be used, within the same company, to denote as different values as 'most likely effort', 'planned effort', 'budgeted effort' and 'price-to-customer'. The decisions on 'bid', 'planned effort' and 'most likely effort', however, have conflicting goals. A bid should, optimally, be low enough to get the job and high enough to maximize profit, the planned effort should enable a successful project and motivate to efficient work, and the estimate of the most likely effort should represent the most realistic use of effort. Mixing these goals may be dangerous.

A consequence of the imprecise use of 'effort estimate' in the studied organization is that the estimation accuracy of different projects was hard to compare and evaluate. We compared the subset of projects that we assessed to be effort estimates of 'most likely effort' (n=6) with those we assessed to be more of type price-to-customer or planned effort (n=17), i.e., where a risk buffer typically was added to the most likely effort. The remaining projects (n=9) were left out of the comparison because they were difficult to classify. The estimates of most likely effort had on average an effort overrun of 11%, while the estimates including a risk buffer had an average effort *underrun* of 8%. From the description of the estimation process it seems as if a common risk buffer was 10-20% of most likely effort. When removing the specified risk buffer from the estimates with risk buffers (n=17) we found that they had, on average, almost the same estimation accuracy (about 10% overrun) as the estimates described as 'most likely effort' (n=6). Finding the average estimation accuracy of all projects, without adjustments, would be like adding 'apples and oranges'.

In the studied organization we also found it necessary to adjust the actual effort for the decreases and increases in delivered functionality to enable a proper interpretation. Seven out of the 32 projects had increases or decreased in functionality of more than 10%. In most cases this adjustment led to better estimation accuracy, i.e., many estimates of most likely effort was accurate, but looked inaccurate because of added or removed functionality. For example, a project went from 50% effort overrun to 10% overrun when we adjust for the increase in functionality. Again, without this adjustment a comparison of estimation accuracy would not give a correct picture of the estimation performance of many of the projects.

These results are supported by the survey described in the first section of this paper. For instance, the survey indicates that the separation between price and most likely effort is blurred in many organizations.

The discussion in this section suggests that a precise estimation terminology is important and may hinder estimation process improvements. So, what is the practice in software engineering textbooks? and, what is the practice in the research on software effort estimation? Is a precise estimation terminology used? If not, this may be a possible reason for the lack of precise estimation terminology in the software industry. We could not find any review on the use of estimation terminology in software textbooks and research. The review in the next section may therefore be the first review of this type.

## 4. The Review of Textbooks and Research

### 4.1. Design of Review Process

The review investigates the use of software effort estimation terminology in software engineering textbooks, estimation surveys and in research papers proposing estimation guidelines. Different approaches were used for selection of study material. We have selected what we believe are the most popular textbooks in software engineering lecturing. As we are only aware of a limited number of software cost estimation surveys, we have included all of them. For research papers proposing estimation guidelines, we searched our own comprehensive online estimation literature library (The BESTWeb Library).

The review focused on the questions (Q1-Q3) listed below. The questions are derived from the software cost estimation terminology problems discussed in Section 3.

- Q1: Is the term 'estimate' precisely defined?
- Q2: Is there a clear and consistent distinction made between estimates of 'most likely effort', 'plan', 'budget' and 'price-to-customer'?
- Q3: When evaluating estimation accuracy, are the estimates and the actual effort comparable? (This may be achieved either through adjustments of actual effort or removal of projects in cases where estimated and actual effort are not comparable.)

If we were unable to answer the question from the text in the books and papers, we provide the value 'Unknown'. When a question is not a topic in a textbook or a paper, we provide the value 'N/A' (not applicable). The review was conducted by two of the authors, independently of each other. Disagreements were discussed. There were only minor disagreements to be resolved.

### 4.2. Review of Software Engineering Textbooks

In the FASE newsletter's rating of the top ten contributions to the Software Engineering Education, Training, and Profession (SEET&P) community[19] the software engineering textbooks by Sommerville [9] and Pressmann [20] are included. In addition to these two, we included another widely used book 'Software Engineering – theory and practice' by Pfleeger [21] in the review. The result of the review is presented in Table 1.

**Table 1 Software engineering textbooks**

| Author/Question | Q1 | Q2 | Q3 |
|---|---|---|---|
| Pressmann | No | No | Unknown |
| Sommerville | No | No | Unknown |
| Pfleeger | No | No | Unknown |

In our opinion, none of the software engineering textbooks provide a precise definition of what they mean by effort estimate. Pressman writes that an estimate is: … *your attempt to determine how much money, effort, resources and time it will take to build a specific software-based system or product (p.642).* Sommerville writes that: *Software estimation is a related activity [to planning] that is concerned with estimating the resources required to accomplish the project plan (p. 47).* Similarly, Pfleeger writes that: *For most projects the biggest component of cost is effort. We must determine how many staff-days of effort will be required to complete the project (p. 99).* None of these descriptions of estimation make a clear distinction between effort estimates and budgets, plans and price. We were unable to find better definitions/descriptions of effort estimates in their books, which each include a separate section on software cost estimation.

The lack of definitions means that when terms like 'effort estimate' and 'estimate' are used, it is difficult to be sure whether estimate refers to the most likely effort, most likely effort added a risk buffer, planned effort, or something else. An example of this difficulty is the following guideline for estimating by Sommerville*: ... estimate as if nothing will go wrong then increase that estimate to cover anticipated problems. A further contingency factor to cover unanticipated problems may also be added to the estimate (p. 53).*

Both Pressmann and Sommerville state that technical as well as functional parameters have a large impact on estimates. Pressmann argues that quality is an important factor in estimation, and claims that deadlines can be met by reducing on quality*: but understand that this will increase risk of poor quality due to the tight deadline (p. 676).* Sommerville states that technical parameters are important in estimation*: What we really want to estimate is the cost of deriving a particular system with given functionality, quality, performance, maintainability and so on (p. 595).* He also touches the topic of incomparable solutions when productivity rates are discussed: *One solution may execute more efficiently while another may be more readable. When solutions with different attributes are produced, comparing their production rate is not really meaningful (p. 592).* However, neither of the software engineering textbooks reviewed discusses the challenges on how to assert estimation accuracy implied by differences in scope and quality between estimated and actual effort. No guidelines are given on how the accuracy should be computed, how to ensure that the estimated results and the actual result are comparable or how estimates can be adjusted to achieve comparability.

### 4.3. Review of Estimation Surveys

The surveys reviewed and our evaluations of their estimation terminology are presented in Table 2.

Table 2 Estimation surveys

| Author / Question | Q1 | Q2 | Q3 |
|---|---|---|---|
| Jenkins [22] | No | No | No |
| Heemstra[23] | No | No | No |
| Lederer [24] | No | Unknown | No |
| Moores [25] | No | Unknown | N/A |
| Bergeron [26] | No | Unknown | No |
| Wyndenbach [27] | No | Unknown | N/A |
| Standish [28] | No | No | No |
| Standish [29] | No | No | No |
| Moløkken [1] | Yes | Yes | No |
| Phan [30] | No | Unknown | No |
| Addison [31] | No | Unknown | N/A |

Only one of the estimation surveys attempts to define what they mean by an effort estimate (Q1). Four surveys clearly mix estimates and other project values, and six of the surveys used such a terminology that we were unable to evaluate if effort estimates were separated from plans and budgets (Q2). Only one of the surveys used a precise terminology. But even if a precise terminology is used in the surveys, this is not enough to ensure that surveys collect the intended values. If no explicit actions are taken to ensure that the participants in the surveys differentiate between the terms, the results of this review indicate that respondents are likely to mix the terms, and hence the surveys will collect different values than intended. Moløkken et al. address this problem by applying interviews instead of questionnaires to improve the consistency in use of estimation terms.

None of the surveys measure estimation accuracy in a way that we find satisfactory (Q3), even if several of the studies partly address the problem. The Standish Group, for example, considers implemented versus estimated functionality and features as a parameter when assessing project success. None of the surveys report that they have adjusted the effort estimates relative to risk buffer added or other factors that lead to incomparability of estimated and actual effort. Lederer and Prasad is one of the few that discuss, among other variables, the impact of reduced quality on estimation accuracy, but the information about reduced quality is not used to improve the meaningfulness of the estimation accuracy measurement.

## 4.4. Review of estimation guideline papers

The estimation guideline papers reviewed were identified by searching for 'guidelines' in the BESTWeb library. The BEST library is an online database that contains abstracts of and references to papers on software effort estimation. It also includes a number of papers that are closely related to software effort estimation. The paper by Armstrong [32] is an example of a paper that does not directly address software cost estimation. However, as the paper suggests generally applicable estimation guidelines, it was included in this review. The result of the review is presented in Table 3.

Table 3 Estimation guideline papers

| Author / Question | Q1 | Q2 | Q3 |
|---|---|---|---|
| Armstrong [32] | No | Yes | N/A |
| Lederer [33] | No | Unkown | N/A |
| Jørgensen[16] | Yes | Yes | N/A |

One of the research papers proposing estimation guidelines defines what is meant by an effort estimate (Q1), and two out of three clearly separate effort estimates from budgets, plans and price (Q2). This is apparent when we among Jørgensen's guidelines find '*avoid confliction goals*' which addresses the difference in goals of different estimate types. A similar guideline is stated by Armstrong: '*Make sure forecasts are independent of politics*' which is elaborated to mean that the estimation process should be separated from the planning process. In the remaining paper, we were not able to determine whether effort estimates are mixed with budgets, plans and price.

Neither of the papers reports estimation accuracy (Q3), but Armstrong as well as Lederer and Prasad provide guidelines that are directly relevant for estimation accuracy assessment. When discussing the guideline '*Anticipate and control user changes*', Lederer and Prasad say that if changes to scope are large enough, they will invalidate the estimates. Armstrong suggests several guidelines relevant for estimation accuracy assessment. Among those are '*clean the data*' that addresses adjustments and 'use objective tests of assumptions' that addresses the validity of the assumptions made when forecasting.

## 4.5. Discussion of Results

Our reviews suggest that a reason for the lack of precise use of estimation terminology in software organizations is the lack of precise terminology in software textbooks and research papers. The relationships may, however, also be in the opposite direction. It is difficult to survey estimation practice when important estimation terms are vague, undefined and used inconsistently by software organizations. Consequently, improvements should start in both camps. The motivation for the improvements may be different.

Software organizations should improve their use of estimation terminology to avoid misunderstandings, to increase the realism in the estimates, and to improve learning from experience. Software researchers need a precise terminology to increase the validity of their research results, e.g., when comparing two formal estimation models.

Possible reasons for the current imprecise terminology and unadjusted estimation accuracy measurement are:

- Authors of estimation literature take a "deterministic" instead of a "probabilistic" view on effort estimation. A probabilistic view means here that 'most likely effort', 'planned effort', 'budgeted effort', etc. are values (with different probability of being exceeded by actual effort) on an effort probability distribution. Without a probabilistic basis of effort estimation terminology a precise differentiating between most likely, planned, and budgeted effort may be difficult to achieve.
- Software organizations do not regard estimation as a separate activity, but regard it as an integrated part of project scheduling, project pricing and project budgeting. As pointed out earlier, mixing processes may mean mixing terminology.
- Authors of estimation literature are impacted by improper analogies (mental models) when discussing estimation accuracy. The lack of adjustments in actual effort for changes in functionality and quality suggests that improper estimation analogies are applied, e.g., analogies from forecasting of weather or economic growth.
- Software organizations do typically not collect the data necessary to validate and adjust the actual effort. Our experience is that most organizations have an immature view on how to measure estimation accuracy measurements and no resources allocated to in-depth analysis of estimation accuracy data across projects.

## 5. Guidelines for Estimation Terminology

Software cost estimation terminology is a large topic and it is beyond the scope of this paper to provide suggestions for a complete terminology. We propose two simple guidelines that we believe, if adopted, will contribute to improved use of software cost estimation terminology and act as a basis for improved software estimation processes. The guidelines are aimed at all users of software cost estimation terminology including authors, practitioners, researchers and reviewers. The two guidelines that we believe are the most important are:

**Do not mix estimation of most likely effort with planning, budgeting or pricing.**

Implication of guideline for researchers:

- Use different terms for different concepts. In particular, separate between estimated 'most likely effort', 'planned effort' and 'budgeted effort'.
- When conducting surveys or logging estimation information, do not assume that your terminology is understood even if you define it. In-depth studies and triangulation may be needed to ensure that all you data are based on the same understanding of your estimation terminology.

Implication of guideline for practitioners:

- Use different terms for different concepts. In particular, separate between estimated 'most likely effort', 'planned effort' and 'budgeted effort'.
- Perform estimation of most likely effort as an independent activity and do not mix it with planning, budgeting and pricing. People in charge of bidding should for example not be in charge of the estimation of most likely effort, to ensure that pricing and realism are not mixed. Planning tools should not be used as estimation tools, or with great care to avoid mix of concerns.

This guideline contributes, we believe, to increased realism, better communication, and better learning from experience.

**When assessing estimation accuracy, make sure that the estimate and the actual effort are comparable.**

Implication of guideline for researchers:

- Adjust the actual efforts so that they are comparable to the estimated effort with respect to technical and functional parameters before calculating estimation accuracy. If functional and quality requirements are not available, investigate the project plan and use interviews to find any changes in scope and/or quality. If estimates are of other types than most likely effort estimates, transform them to most likely estimates before calculating the accuracy.
- When estimates cannot be reliably transformed to values that are comparable to the actual result, take great care when using these results or remove the projects from the data set.

Implication of guideline for practitioners:

- Record the scope and other assumptions of the estimate of most likely effort, e.g., by including quantifiable quality requirements in the requirement specification. Specify the version of the requirement specification and other documents that estimate of most likely effort is based on.
- Record deviation from estimated scope, quality, and development process.

This guideline contributes, we believe, to better evaluations, valid comparisons and better reporting of estimation performance.

More estimation guidelines are found in [16, 32].

## 6. Summary

Effort and schedule overruns are among the most serious problems in the software industry. In this paper we argue that the lack of a precise software effort estimation terminology is an important obstacle for estimation accuracy improvement. We reviewed industrial practice, software engineering textbooks, estimation surveys and estimation research papers proposing guidelines for effort estimation. We found that estimates of most likely effort are frequently mixed with planned effort, budgets and price-to-customer. In addition, effort estimation accuracy is frequently measured without adjustments for differences in scope and/or quality assumed when estimating the effort and the system actually implemented.

In order to improve effort estimation accuracy, a more precise software effort estimation terminology is needed. We provide two guidelines for this purpose: 1) Do not mix estimation of most likely effort with planning, budgeting or pricing, and 2) When assessing estimation accuracy, make sure that the estimate and the actual effort are comparable.

## 7. Acknowledgement

Thanks to Barbara Kitchenham for useful comments on an early draft of this paper. Also, thanks to the Norwegian Research Council who sponsored this work through the SPIKE project.

### References

1. Moløkken, K. *A Survey on Effort Estimation in Norwegian Software Industry*. Accepted for *METRICS 2004*. 2004. Chicago, USA.
2. Moløkken, K., et al., *Project Estimation in the Norwegian Software Industry – A Summary*, in *Simula Report 2004-03*. 2004, Simula.
3. Haugnes, G.M., *Offentlig IT-sprekk for 6 mrd. hvert år*, in *Aftenposten*. 2004: Oslo.
4. Kitchenham, B., *Software Metrics: Measurement for Software Process Improvement*. 1996: Blackwell Publishers.
5. DeMarco, T. and T. Listener, *Waltzing with bears : managing risk on software projects*. 2003: Dorset House.
6. Boehm, B. and R. Fairley, *Software estimation perspectives.* IEEE Software, 2000. **17**(6): p. 22-26.
7. Edwards, J.S. and T.T. Moores, *A conflict between the use of estimating and planning tools in the management of information systems.* European Journal of Information Systems, 1994. **3**(2): p. 139-147.
8. Abran, A. and P.N. Robillard, *Function points analysis: an empirical study of its measurement processes.* IEEE Transactions on Software Engineering, 1996. **22**(12): p. 895-910.
9. Sommerville, *Software Engineering*. 5. ed. 1996: Addison Wesley.
10. Jørgensen, M., *How much does a vacation cost?* Software Engineering Notes, 2003. **28**(6): p. 30.
11. Johnson, J., et al., *The Criteria for Success - Industry Trend or Event*, in *Software Magazine*. 2001.
12. Moløkken, K. and M. Jorgensen. *A review of software surveys on software effort estimation*. in *International Symposium on Empirical Software Engineering*. 2003. Rome, Italy: Simula Res. Lab. Lysaker Norway.
13. Cosier, R.A. and G.L. Rose, *Cognitive conflict and goal conflict effects on task performance.* Organizational Behaviour and Human Performance, 1977. **19**(2): p. 378-391.
14. Keen, P.G.W., *Information systems and organizational change.* Social Impacts of Computing, 1981. **24**(1): p. 24-33.
15. Buehler, R., D. Griffin, and H. MacDonald, *The role of motivated reasoning in optimistic time predictions.* Personality and Social Psychology Bulletin, 1997. **23**(3): p. 238-247.
16. Jørgensen, M., *A review of studies on expert estimation of software development effort.* Journal of Systems and Software, 2004. **70**(1-2): p. 37-60.
17. Jørgensen, M., L. Moen, and N. Løvstad. *Combining Quantitative Software Development Cost Estimation Precision Data with Qualitative Data from Project Experience Reports at Ericsson Design Center in Norway*. in *Conference on Empirical Assessment in Software Engineering*. 2002. Keele, England: Keele University.
18. Jørgensen, M. and D.I.K. Sjøberg, *Impact of effort estimates on software project work.* Information and Software Technology, 2001. **43**(15): p. 939-948.
19. FASE, *Top Ten Contributions.* FASE Newsletter, 1999. **9**(12).
20. Pressman, R.S., *Software engineering - a practitioner's approach*. 6. ed. 2001: McGraw-Hill.
21. Pfleeger, S.L., *Software Engineering - theory and practice*. 1998: Prentice Hall.
22. Jenkins, A.M., J.D. Naumann, and J.C. Wetherbe, *Empirical investigation of systems development practices and results.* Information and Management, 1984. **7**(2): p. 73-82.
23. Heemstra, F.J. and R.J. Kusters. *Controlling Software Development Costs: A Field Study*. in *International Conference on Organisation and Information Systems*. 1989. Bled, Yugoslavia.
24. Lederer, A.L. and J. Prasad, *Information systems software cost estimating: a current assessment.* Journal of Information Technology, 1993. **8**(1): p. 22-33.
25. Moores, T.T. and J.S. Edwards, *Could Large UK Corporations and Computing Companies Use*

*Software Cost Estimating Tools? - A Survey*. European Journal of Information Systems, 1992. **1**(5): p. 311-319.

26. Bergeron, F. and J.Y. St-Arnaud, *Estimation of information systems development efforts: a pilot study*. Information and Management, 1992. **22**(4): p. 239-254.
27. Wydenbach, G. and J. Paynter, *Software Project Estimation: a Survey of Practices in New Zealand.* New Zealand Journal of Computing, 1995. **6**(1B): p. 317-327.
28. *The CHAOS Report*. 1994, The Standish Group.
29. *Chaos Chronicles Version 3.0*. 2003, The Standish Group: West Yarmouth, MA.
30. Phan, D., D. Vogel, and Nunamaker, *The Search for Perfect Project Management.* Computerworld, 1988: p. 95-100.
31. Addison, T. and S. Vallabh. *Controlling Software Project Risks - an Empirical Study of Methods used by Experienced Project Managers*. in *SAICSIT 2002*. 2002. Port Elizabeth, South Africa.
32. Armstrong, J.S., *Standards and practices for forecasting*, in *Principles of forecasting:A handbook for researchers and practitioners*. 2001, Kluwer Academic Publishers: Boston.
33. Lederer, A.L. and J. Prasad, *Nine management guidelines for better cost estimating.* Communications of the ACM, 1992. **35**(2): p. 51-59.