A Review of Surveys on Software Effort Estimation

Kjetil Moløkken and Magne Jørgensen
Simula Research Laboratory
P.O. Box 134
NO-1325 Lysaker
NORWAY
Tel.: +47 67828200)
{kjetilmo,magnej}@simula.no

Abstract

This paper summarizes estimation knowledge through a review of surveys on software effort estimation. Main findings were that: (1) Most projects (60-80%) encounter effort and/or schedule overruns. The overruns, however, seem to be lower than the overruns reported by some consultancy companies. For example, Standish Group's 'Chaos Report' describes an average cost overrun of 89%, which is much higher than the average overruns found in other surveys, i.e., 30-40%. (2) The estimation methods in most frequent use are expert judgment-based. A possible reason for the frequent use of expert judgment is that there is no evidence that formal estimation models lead to more accurate estimates. (3) There is a lack of surveys including extensive analyses of the reasons for effort and schedule overruns.

Keywords: Effort estimation, surveys, software projects.

1. Introduction

The software industry's inability to provide accurate estimates of development cost, effort, and/or time is well known. This inability is described in reports from project management consultancy companies, case studies on project failures, articles in the computer press, and estimation surveys. The common belief seems to be that the cost overruns are very large, and we have experienced that few software professionals and researchers react with disbelief when being presented with the inaccuracy figures reported in Standish Group's Chaos Report [1], i.e., an average cost overrun of 89%. There may be several reasons for this attitude.

The introduction part of many research papers on software effort estimation includes references to the estimation studies with the most extreme findings. This may, in part, be due to the authors' need to legitimize their own research. Surveys, which report none, or small, overruns may not be cited as often, since they do not 'contribute' to defining effort estimation as a central problem area in software engineering.

The estimation accuracy results reported may in some cases be biased towards high inaccuracy, e.g., studies conducted by consultants selling estimation advice, journalists writing a story on project failures, or software houses who sell estimation tools.

It is difficult to get a balanced view on the software industry's estimation performance without unbiased information from a representative set of projects and organizations. The surveys presented in scientific journals and conferences may be a source for such unbiased information. This paper summarizes estimation results from surveys on software estimation. To our knowledge, there has not been conducted any structured review of the estimation surveys with the aim of summarizing our knowledge of software effort estimation performance. Besides summarizing results from estimation surveys, the main purposes of this paper are to challenge some common estimation beliefs that may be inaccurate, to discuss methodical aspects related to completion of industrial surveys, and, to support future surveys on software effort estimation.

In the remaining part of this paper, we present several research questions relevant to the field of software project estimation (Section 2). Then, we present an overview of relevant surveys (Section 3). The research questions are then discussed in light of the findings of the surveys (Section 4). Finally, we conclude and outline further research (Section 5).

2. Research questions

The rationale of this review of surveys is to summarize findings on central aspects of software estimation, to enable a more balanced estimation debate, and to suggest further research. The central estimation aspects are presented as research questions. We have only included questions addressed by more than one of the reviewed surveys. Our research questions (RQs) are:

1. To what extent do software development projects deviate from the original plan, with regard to cost, schedule and functionality?
2. Which methods are used to estimate software effort, and do these systematically differ in accuracy?
3. How important is accurate effort estimation perceived to be, and to what extent is the level of accuracy considered a problem in the software industry?
4. What are the main causes for software projects to deviate from their original plan?

Due to the limited number of surveys, the different research designs, and the methodological limitations of many of the surveys, we do not expect to discover definite answers to these questions. We do, however, hope to challenge some estimation myths and to enable a better foundation for further estimation research.

3. Estimation surveys

The surveys reviewed are listed in chronological order based on year of their first appearance, see overview in table 1.

The surveys are named after the first author of the paper in which the survey first appeared. The remaining part of this section provides a brief outline of the survey designs and research foci.

In order to avoid too much influence from one particular organization, case studies with just one subject is omitted in this review.

Table 1. Survey overview.

#	Name	Year	Sample	Respondents	Response Rate	Country	Rnd. sample	Interview
1	Jenkins	1984	N/A	72	N/A	USA	No	Yes
2	McAulay	1987	280	120	42.9%	New Zealand	No	No
3	Phan	1988	827	191	23.1%	USA	No	No
4	Heemstra	1989	2659	598	22.5%	Netherlands	No	No
5	Lederer	1991	400	112	28.0%	USA	No	No
6	Bergeron	1992	374	89	23.8%	Canada	No	No
7	Moores	1992	115	54	47.0%	UK	No	No
8	Standish	1994	N/A	365	N/A	USA	No	No
9	Wydenbach	1995	515	213	41.4%	New Zealand	No	No
10	Addison	2002	70	36	51.4%	South Africa	No	No

3.1. Survey 1: Jenkins

Jenkins, Naumann and Wetherbe [2] conducted a large empirical investigation in the beginning of the 1980s. The study focused on the early stages of system development. It included development aspects, such as user satisfaction, development time, and cost overruns. They interviewed managers from 23 large organizations and collected data on 72 projects. The average project cost was $103,000, and the average duration was 10.5 months. The study included projects that were considered small, medium and large relative to the organizations standards. A majority of the projects developed new software systems (55%), but redesign (33%) and enhancement (11%) of existing software systems were also represented. The survey measured three success factors; user satisfaction, being “on-time” and being “on-budget”.

3.2. Survey 2: McAulay

McAulay [3] conducted a survey on software metrics in New Zealand. This survey is unpublished, but some information is presented by Purvis, MacDonell et al. [4]. In order to investigate information system development projects, questionnaires were sent to 280 organizations. Out of these, 120 were returned.

3.3. Survey 3: Phan

Phan et al. [5, 6] tried to asses to what extent, and for what reasons, software development projects encountered cost and schedule overruns. Questionnaires were sent out to 827 professionals, and they received 191 responses. The projects involved were fairly large, with an average of 102 person months.

3.4. Survey 4: Heemstra

Heemstra and Kusters [7-9] conducted a survey of cost estimation in Dutch organizations. The goal was to provide an overview of the state oft the art of estimation and controlling software development costs. They sent out 2659 questionnaires, and got responses from 598 organizations. Estimation methods, original project estimates and actual effort were analyzed.

3.5. Survey 5: Lederer

Lederer and Prasad [10-13] conducted a survey concerning software development cost estimates. Through a questionnaire, 112 software managers and other professionals (out of 400 possible) reported their views on a wide variety of cost estimation aspects. The respondents represented fairly large companies, with an average of 478 employees in the information systems department.

3.6. Survey 6: Bergeron

Bergeron and St-Arnaud [14] performed a study to identify estimation methods, and to what extent they were used. They also investigated how choice of method, and underlying factors and variables, influenced estimation accuracy. In total, 374 Questionnaires were sent to 152 organizations. The companies each received 1-4 copies of the questionnaire. The 89 responses received came from 67 different organizations. All projects included were larger than 150 person-days.

3.7. Survey 7: Moores

Moores and Edwards [15] sought to investigate why there was an apparent lack of use of software cost estimating tools. A total of 115 large UK corporations were approached, and they received 54 responses.

3.8. Survey 8: The Standish Group 'Chaos report'

Although not a strictly scientific survey, the Standish Group's 'Chaos report' seems to have made such a strong impact (perhaps more than any scientific survey) on common estimation beliefs that it deserves to be included in this review. The first and most cited version is the 'Chaos report' from 1994, but Standish Group has continued data collection during the nineties. Total sample size of the 1994-report was 365 respondents. All projects were classified as, success (delivered as planned), challenged (delivered over time, and over budget and with fewer than specified features) or impaired (cancelled). The sample selection process of organizations and projects is unknown (we have made several inquiries to the Standish Group about properties such as the sample involved. The have refused to provide these details, claiming them to be 'business secrets'.), along with other important design and measurement issues. It is possible that the estimation accuracy reported by Standish Group is misleading. For example, from our inspection of the survey questionnaire available on their web-site, it seems as projects completed ahead of plans had to be registered as projects with "less than 20% overrun". If our assumption is correct, this may have led to too high average cost estimation overrun values.

3.9. Survey 9: Wydenbach

Wydenbach and Paynter [16] investigated the estimation practices in New Zealand on basis of a previous survey [7]. They sent questionnaires to what was believed to be a representative sample of companies (515), and received 213 usable responses.

3.10. Survey 10: Addison

Addison and Vallabh [17] investigated the perceptions of project managers on software project risks and controls. A "snowball sample" (explained in Section 4.1) was used to identify 70 managers, of whom 36 returned the questionnaire. Although not a study with a focus on estimation, it reports on aspects related to budgets and plans.

4. Discussion

As seen in Section 3, the surveys are different in many aspects. This makes it challenging to compare, combine and present the results. On the other hand, the differences may also add value and insight to our research questions. This section provides a review of general properties and methodological aspects of the surveys and addresses the research questions described in Section 2.

4.1. Survey designs

The surveys addressed companies, projects, software professionals, or a combination of these. Unfortunately, none of the surveys involved a procedure to ensure random samples. There was a variety of sample selection methods, such as contacting members of a society, snowball sampling (encouraging first round survey participants to suggest second round survey participants, etc.) or deliberately targeting specific categories of companies (e.g., large). The lack of procedures to ensure random samples, and the lack of proper analyses of the population represented by the samples, may be a major problem with all of the surveys, potentially leading to difficulties when interpreting and transferring results. This, in combination with a low response rate, may have unwanted influence on the validity of the results. For those who have provided response rates, the rates ranged from 22,5% [7] to 51,4% [17].

Only one of the surveys (Bergeron) used personal interview as a data collection method [2]. The other studies mailed questionnaires to potential respondents. This may have influenced the quality of the data, since mailed questionnaires lessens the involvement and commitment of the participants [18], and make misinterpretations of answers and questions more likely. Another problem may be that many of the surveys relied on the managers being unbiased when, for example, reporting magnitude of and reasons for overruns. It is possible that managers may be biased towards under-reporting overruns and have a tendency to over-report customer issues as reasons for overruns.

Although the surveys span four continents, it is essential to observe that the surveys were conduction in only six, similar, western countries, i.e., we do not know much about the preferred estimation methods and their performance in other cultures.

There are also several possible, and relevant, research questions that we are unable to address on basis of the reviewed surveys. One such aspect is possible differences of project properties based on the type of developing organization. Do, e.g., CMS (Content Management System)-projects differ from defence projects on choice of estimation methods? And how does in-house development versus development for customers affect estimation accuracy? Such topics have been addressed by other studies, e.g., [19]. The surveys presented in this paper also make such differentiations, to some extent.

The survey by Heemstra and Kusters [7] differentiates between in-house and other types of development. Some of the other surveys also distribute the organizations into different sectors [13-15]. However, none of the papers on the surveys report any differences between type of development projects, related to either choice of method or accuracy. They mainly report the distribution of industries in order to show that their survey included a diverse sample. For this reason, it was impossible to investigate possible differences due to project, application, development method or organization in this review.

An important aspect of project management is how to avoid project abortion and/or restarts. This may sometimes relate to effort estimation, but may also be caused by market changes, customer orders, company restructuring or similar reasons. None of the surveys presented relate the possibility of failure to estimation method, and it is therefore not treated in this review.

4.2. Survey results

The discussion in this section is structured around the research questions (RQs) presented in Section 2.

RQ 1: *To what extent do software development projects deviate from the original plan, with regard to cost, schedule and functionality?*

The results from surveys investigating the frequency or magnitude of overruns are displayed in table 2.

Table 2: Estimation accuracy results

Study	Jenkins	Phan	Heemstra	Lederer	Bergeron	Standish
Cost overrun	34% (median)	33% (mean)			33% (mean)	89% (mean)
Projects completed over budget	61%		70%	63%		84%
Projects completed under budget	10%			14%		
Schedule overrun	22%					
Projects completed after schedule	65%		80%			84%
Projects completed before schedule	4%					

Although the surveys report different results; the tendency is clear, a majority of the projects encounter overruns. Phan [6] included a report on overruns on organizational level that is not directly comparable with the other results in table 2. He found that cost overruns occurred always in 4%, usually in 37%, sometimes in 42%, rarely in 12%, and, never in 4% of the 191 organizations surveyed. Similarly, schedule overruns occurred always in 1%, usually in 31%, sometimes in 50%, rarely in 15%, and, never in 2% of the organizations. Bergeron and St-Arnaud [14] found that 58% of the projects surveyed had cost overruns of more than 20%.

Heemstra and Kusters [7] found that overruns increased with project size. Very large projects were defined as projects consuming more than 200 man-months. These projects had overruns of more than 10% in 55% of the cases. For all projects, overruns of 10% or more occurred in 28% of the sample. A similar tendency on larger overruns for larger projects was found by the Standish Group. In Jenkins' study, however, the occurrence of overruns was equally likely for small, medium, and large projects.

The degree of delivered functionality is difficult to measure, since it may be subject to differences in opinions. None of the surveys concerned the degree of delivered functionality opinions of the actual users, but according to the managers in the Jenkins study, 72% of the users was satisfied or very satisfied with the functionality [2]. Similarly, 70% of Phans respondents claimed that user requirements and expectations were usually met. The Standish Group [1], on the other hand, claimed that only 7.3% of the projects delivered all the functionality originally specified.

To summarize, it seems as if most projects (60-80%) are completed over budget and/or schedule. Most results also indicate that the percentage and magnitude of overruns increase as projects grow in size. The magnitude of overruns may, however, not be as dramatic as reported by Standish Groups' Chaos Report. Most surveys, e.g., [2, 6, 14], and other case studies [20, 21] suggest that a more likely average effort and cost overrun is between 30 and 40%.

RQ 2: *Which methods are used to estimate software effort, and do these systematically differ in accuracy?*

It is difficult to compare the surveys that investigated estimation methods since none of them operated with the same categories of estimation methods. A further complicating factor is that the respondents may have interpreted pre-defined categories, e.g., analogy-based estimation, differently. We have grouped the estimation methods in three main categories: Expert judgment-based methods, model-based methods and "other". Model based estimates include formal estimation models such as COCOMO, Use-Case-based estimation, FPA-metrics or other algorithm driven methods. In the category of "other" there are methods that are not "pure" estimation methods, e.g., capacity related and price-to-win-based methods, and methods than can be used in combination with other models (top-down and bottom-up). An overview is presented in Table 3. An 'X' in the table indicates this alternative was not an option in the survey. In the McAulay column, we have joined three different software cost model method alternatives of that study. The original study found that 11% applied Function Point Analysis, 2% lines of code based models and 0% Halstead Metrics.

Table 3: Estimation methods results

		McAulay (n=114)	Heemstra (n=369)	Wydenbach (n=209)	Bergeron (n=89)
Estimation Methods		**Percentage used (more than one method possible)**			**Importance (1-4)**
Expert based	Expert consultation	X	26%	86%	1.8
	Intuition and experience	85%	62%	X	3.3
	Analogy	X	61%	65%	2.5
Model based	Software Cost Models	13%	14%	26%	1.3
Other	Price-to-win	X	8%	16%	1.2
	Capacity related	X	21%	11%	1.6
	Top-down	X	X	13%	1.4
	Bottom-up	X	X	51%	2.4
	Other	12%	9%	0%	1.5

A problem with Table 3 is the overlapping of categories. For example, one could argue that "estimation by analogy" and "top-down" in many cases are two aspects of the same estimation method. How the respondents interpreted the estimation method categories is not possible to derive from the survey results.

The respondents were not asked about the extent of use of one method in the organization, only about whether an estimation method was used or not. This means that an estimation method used to estimate very few projects per organization gets a too high percentage in Table 3.

It is also essential to note that projects may be estimated by a combination of two or more different methods, e.g. model and expert-based. Such combination has been advocated in other studies [22]. To which extent such combination methods were used was not addressed by any of the surveys. We were therefore unable to draw conclusions on a possible beneficial effect of combining methods in this review.

'Price-to-win' is listed as an estimation method, but we believe that most managers would not report this as an estimation method even if "price-to-win" had an impact on their estimates. This may be the case because mangers are not aware of the effect of customer expectations on effort estimates [23]. They may not feel that that it is an estimation method, or they believe that they should not be impacted by the "price-to-win".

Lederer and Prasad [10] applied a different approach. Instead of asking what methods were used, they asked how estimates were influenced. Respondents rated alternatives on a five point Likert scale (max=5). The results are displayed in Table 4.

Responses 1, 2, 3 and 5 seem to be expert judgment-based, responses 7 and 8 to be model-based, while responses 4 are 6 are more difficult to interpret, i.e., the expert judgment-based methods seem to be more important in this survey as well.

The results from all identified surveys point in one direction: Expert estimation is by far the preferred method for software estimation. This is further backed by a variety of case studies not included in this review [21, 24-26]. In fact, we have not been able to identify a single study reporting a dominant use of formal estimation methods.

Table 4: Estimation responses ranked by importance

	Response Categories	Mean
1	Comparison to similar, past projects based on personal memory	3.77
2	Comparison to similar, past projects based on documented facts	3.41
3	Intuition	3.38
4	A simple arithmetic formula (such as summing task durations)	3.09
5	Guessing	2.76
6	Established standards (such as averages, standard deviations, etc.)	2.33
7	A software package for estimating	1.80
8	A complex statistical formula (such as multiple regression, differential equations, etc.)	1.49

Cultural similarities between the samples or organizations may, however, limit the transfer of the results to other cultures. Perhaps surveys conducted in China, India, or Germany would have yielded a different outcome.

Heemstra and Kusters [7] found that estimation accuracy did not improve when formal models were used. In fact, projects estimated with Function Point Analysis had larger overruns than the other projects. Similarly, Bergeron and St-Arnaud [14] found that price-to-win methods and algorithmic models were associated with less accurate estimates. The methods associated with the most accurate estimates were based on analogy and expert opinion. However, this may be coincidental since few respondents used price-to-win and algorithmic methods.

On a general basis, it is important to be aware of there are several aspects that may affect choice of estimation methods. It may be the case that especially challenging and/or large projects more often rely on formal estimation methods, or that the projects applying, e.g., algorithmic estimation methods may be different from the other projects. None of the surveys address this possibility.

The lack of evidence that estimation models are more accurate than expert judgment [27], may be an important reason for the widespread use of expert estimation. In addition, expert estimation may have the advantage of being easy to use and flexible.

RQ 3: *How important is accurate effort estimation perceived to be, and to what extent is the level of accuracy considered a problem in the software industry?*

The only survey that directly addressed RQ 3 was conducted by Lederer and Prasad [11]. On a five point Likert scale, the average importance rating reported by managers was 4.17. Although that survey found that managers perceived estimation as important; this does not necessarily imply that projects as a rule are estimated. We found three surveys on the proportions of software organizations that estimated costs. Heemstra [9] reports that 65% and McAulay [3] report that 95% of the organizations as a rule estimate projects. Wydenbach and Paynter [16] report that 88% of the companies estimate at least half of their projects.

One survey studied the estimation percentage at project level [11]. They found that 87% of the companies' large projects were estimated.

Moores and Edwards [15] found that 91% of the responding managers answered 'yes' to the question 'do you see estimation as a problem?', while only 9% answered 'no'. An interesting finding is that the accepted level of estimation accuracy was typically +/- 20%. A finding supporting the belief that estimation is perceived as a problem was reported by Addison and Vallabh [17]. They found that the risk factor viewed as most problematic by software professionals was 'unrealistic schedules and budgets'.

Combined, these surveys indicate an awareness of estimation as a problem and an important activity. There are, however, many other important software development activities, e.g., contract negotiations with customers. An appropriate approach to reveal how important the companies regard effort estimation accuracy in practice is to investigate the organizations actual focus on improving estimation processes and effort spent to achieve accurate estimates. We have found no such studies.

RQ 4: *What are the main causes for software projects to deviate from their original plan?*

A problem when analyzing reasons for project overruns is that the respondents may be biased and/or affected by selective memory. Ideally, estimation accuracy reviews should be conducted by uninvolved reviewers to, for example, avoid the "blame the others"-attitude. The results on reasons for project overruns presented here should therefore be interpreted relative to the role and perspectives of the respondents, typically project managers.

In the survey by Phan [6], the participants were asked for the reasons why projects had cost or schedule overruns. The respondents believed that cost overruns were most often caused by over-optimistic estimates (51%), closely followed by changes in design or implementation (50%). The reasons for schedule overruns were optimistic planning (44%), followed by frequent major (36%) and minor (33%) changes from the original specifications.

In Lederer and Prasad's study [11] the respondents rated 24 possible reasons for inaccurate estimates on a five point Likert scale. The top rated causes were ‘Frequent requests for changes by users’ (3.89), ‘Users’ lack of understanding of their own requirements’ (3.60) and ‘Overlooked tasks’ (3.60), i.e., the problems with the users were the two most important reasons for estimation inaccuracy.

In sum, over-optimistic estimates and user changes or misunderstandings were all important reasons for project overruns, from the perspective of the managers of the software provider organization.

Over-optimism does not necessarily describe properly what happens when a cost estimate is too low. For example, we have experienced that many projects initially have realistic cost estimates. Then, the management believes that the estimates are unacceptable high and put a pressure on the estimators to reduce the estimates, i.e., it may not be over-optimism but cost reduction pressure from customer or management that lead to estimates reported as “over-optimistic”.

5. Conclusions and further studies

Our search for, and review of, surveys suggest that there are few that are directly related to software effort estimation. Also, the design of these surveys often make transfer of results problematic, e.g., non-random samples, low response rates, and frequent use of data collection techniques (questionnaires) potentially leading to low data quality.

The following observations, derived from the surveys, should therefore be interpreted carefully:

- Expert estimation is the method in most frequent use. There is no evidence that the use of formal estimation methods on average leads to more accurate estimates.
- Project overruns are frequent, but most projects do not suffer from major overruns. The average cost overrun reported by Standish Group’s Chaos Report (89%) is not supported by other surveys. An average cost overrun of 30-40% seems to be the most common value reported.
- Managers state that accurate estimation is perceived as a problem.
- The reasons for overruns are complex, and not properly addressed in software estimation surveys. For example, software managers may have a tendency to over-report causes that lies outside their responsibility, e.g., customer-related causes.

Forthcoming surveys should seek to investigate more thoroughly on several aspects related to when, how and why estimation methods are chosen. In this way we can learn how choice of methods, or combinations of methods, may be influenced by properties such as project type or size. Only when such information is provided may we be able to compare the level of accuracy of different methods. Perhaps we may be able to find more about topics like when and how to combine different methods.

It would also be interesting to know what level of estimation accuracy managers are satisfied with, and how this varies depending on type of development.

We are currently conducting a large in-depth survey of Norwegian companies and projects, were we, amongst others, address the research questions treated in this paper and aim to compare the current situation in Norway with previous estimation surveys. A goal of that survey is to find out how the different projects actually are estimated. For example, if a project applies an estimation model, we will find out how the model is used. Do the experts adjust the output from the model? And when a project applies expert estimation, is this done with the aid of checklist or experience databases, and is it based on a combination of different experts’ estimates? The survey also tries to obtain the respondents interpretation of ‘estimate’. Is it interpreted as the most likely effort, the price-to-win, or something else? In order to obtain knowledge of these, and other central aspects of software estimation, personal interviews with both top management and the project managers are employed.

Acknowledgement

This research was funded by the Research Council of Norway under the project INCO.

References

1. Standish, G., *The Chaos Report*. 1994, The Standish Group.
2. Jenkins, A.M., J.D. Naumann, and J.C. Wetherbe, *Empirical Investigation of Systems Development Practices and Results.* Information & Management, 1984. **7**: p. 73-82.
3. McAulay, K. *Information Systems Development and the Changing Role of MIS in the Organisation*. in *First New Zealand MIS Management Conference*. 1987. Wellington.
4. Purvis, M.K., S.G. MacDonell, and J. Westland, *Software Metrics in New Zealand: Recent Trends.* New Zealand Journal of Computing, 1994. **5**(2): p. 13-21.
5. Phan, D., D. Vogel, and Nunamaker, *The Search for Perfect Project Management*, in *Computerworld*. 1988. p. 95-100.
6. Phan, D., *Information Systems Project Management: an Integrated Resource Planning Perspective Model*, in *Department of Management and Information Systems*. 1990, Arizona: Tucson.
7. 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.

8. Heemstra, F.J. and R.J. Kusters, *Function point analysis: Evaluation of a software cost estimation model.* European Journal of Information Systems, 1991. **1**(4): p. 223-237.

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

10. Lederer, A.L. and J. Prasad, *Nine management guidelines for better cost estimating.* Communications of the ACM, 1992. **35**(2): p. 51-59.

11. Lederer, A.L. and J. Prasad, *Information systems software cost estimating: a current assessment.* Journal of Information Technology, 1993(8): p. 22-33.

12. Lederer, A.L. and J. Prasad, *Causes of Inaccurate Software Development Cost Estimates.* Journal of Systems and Software, 1995(31): p. 125-134.

13. Lederer, A.L. and J. Prasad, *The Validation of a Political Model of Information Systems Development Cost Estimating.* Computer-Personnel, 1991. **13**(2): p. 47-57.

14. Bergeron, F. and J.-Y. St-Arnaud, *Estimation of Information Systems Development Efforts: A Pilot Study*. Information & Management, 1992. **22**: p. 239-254.

15. 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.

16. 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.

17. 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.

18. Cozby, P.C., *Methods in behavioral research*. 5th ed. 1993, Mountain View: Mayfield Publishing Company.

19. Maxwell, K.D. and P. Forselius, *Benchmarking Software Development Productivity.* IEEE Software, 2000. **17**: p. 80-88.

20. Boehm, B., *Software Engineering Economics*. 1981, Englewood Cliffs, NJ: Prentice-Hall.

21. Kitchenham, B., et al., *A case study of maintenance estimation accuracy.* To appear in: Journal of Systems and Software, 2002.

22. Höst, M. and C. Wohlin. *An Experimental Study of Individual Subjective Effort Estimations and Combinations of the Estimates*. in *Proceedings the 20th International Conference onSoftware Engineering*. 1998. Kyoto, Japan.

23. Jørgensen, M. and D.I.K. Sjøberg, *The impact of customer expectation on software development effort estimates.* To appear in Journal of Project Management, 2004. **22**(4).

24. Jørgensen, M. *An empirical evaluation of the MkII FPA estimation model*. in *Norwegian Informatics Conference*. 1997. Voss, Norway: Tapir, Oslo.

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

26. Hihn, J. and H. Habib-Agahi. *Cost estimation of software intensive projects: A survey of current practices*. in *International Conference on Software Engineering*. 1991.

27. Jørgensen, M., *A Review of Studies on Expert Estimation of Software Development Effort.* To appear in Journal of Systems and Software, 2004 (an early version of this paper can be downloaded from: http://www.simula.no/publication_one.php?publication_id=444).