A Systematic Review of Software Development Cost Estimation Studies

Magne Jørgensen, Simula Research Laboratory
Martin Shepperd, Brunel University

Abstract: This paper aims to provide a basis for the improvement of software estimation research through a systematic review of previous work. The review identifies 304 software cost estimation papers in 76 journals and classifies the papers according to research topic, estimation approach, research approach, study context and data set. Based on the review, we provide recommendations for future software cost estimation research: 1) Increase the breadth of the search for relevant studies, 2) Search manually for relevant papers within a carefully selected set of journals when completeness is essential, 3) Conduct more research on basic software cost estimation topics, 4) Conduct more studies of software cost estimation in real-life settings, 5) Conduct more studies on estimation methods commonly used by the software industry, and, 6) Conduct fewer studies that evaluate methods based on arbitrarily chosen data sets.

Keywords: systematic review, software cost estimation, software effort estimation, software cost prediction, software effort prediction, research methods

1 Introduction

This paper reviews journal articles on software development cost[1] estimation with the goal of supporting and directing future estimation research. Our review differs from previous reviews [1-3] with respect to the following elements:

[1] The main cost driver in software development projects is typically the effort and we, in line with the majority of other researchers in this field, use the terms cost and effort interchangeably in this paper.

- *Different goal*. While the main goal of this review is to direct and support future estimation research, the other reviews principally aim at introducing software practitioners or novice estimation researchers to the variety of formal estimation models. This difference in goal leads to different focuses. For example, our review focuses on research method and does not include a comprehensive description of the different estimation methods.
- *More comprehensive and systematic review*. We base the analysis on a systematic search of journal papers, which led to the identification of 304 journal papers. The review in [1] is based on about 130, that in [2] on about 65, and that in [3] on about 40 journal and conference papers. None of the previous reviews describe a systematic selection process or state clear criteria for inclusion or exclusion.
- *Classification of studies*. We classify the software development estimation papers with respect to estimation topics, estimation approach, research approach, study context and data set. We found no classification, other than that with respect to estimation method in the other reviews.

The remaining part of this paper is organized as follows: Section 2 describes the review process. Section 3 analyzes and discusses properties of the existing estimation research. Section 4 summarizes the main recommendations for future research on software cost estimation.

2 Review Process

2.1 Inclusion Criteria

The main criterion for including a journal paper in our review is that the paper describes *research on software development effort or cost estimation*. Papers related to estimation of software size, assessment of software complexity, or identification of factors correlated with software development effort, are only included if the main purpose of the studies are to improve software development effort or cost estimation. We exclude pure discussion/opinion papers.

Papers based on the same data set but with different focus are included as different papers. Fortunately, the number of such cases was small and would not lead to

important changes in the outcome of our analysis. We decided, therefore, not to exclude any papers for that reason. However, when conducting a review of a particular phenomenon, e.g., the robustness of a particular finding, we would recommend a clearer distinction between study and paper and a more careful inclusion process.

2.2 Identification of Papers

The search for papers was based on an issue-by-issue, manual reading of titles and abstracts of *all* published papers, starting with volume 1, in more than 100 potentially relevant, peer-reviewed journals with papers written in English. These journals were identified through reading reference lists of cost estimation papers, searching on the internet for previously not identified, relevant journals, and using our own previous experience. Both authors constructed a list of potentially relevant journals independently. These two lists were merged together. In spite of the high number of identified journals, it is possible that there are, e.g., national or company specific, journals with software cost estimation papers that we have missed.

Papers that were potential candidates for inclusion in our review were read more thoroughly to decide whether to include them or not. In total, 304 relevant papers were found in 76 of these journals. The 76 journals and 304 papers are listed in, respectively, Appendices 1 and 2. The search was completed in April 2004.

2.3 Classification of Papers

For the purpose of our analysis the authors of this paper classified the papers according to the properties and categories listed in Table 1. The categories are explained in Appendix 3.

Table 1: Classification of Papers

Property	**Categories**
Research topic	Estimation method, production function, calibration of models, size measures, organizational issues, uncertainty assessments, measures of estimation performance, data set properties, other
Estimation	Regression, analogy, expert judgment, work break-down, function

approach	points, classification and regression trees, simulation, neural network, theory, Bayesian, combination of estimates, other, not relevant
Research approach	Theory, survey, experiment, case study, development of estimation method, history-based evaluation, own experience, real-life evaluation, review, simulation, other
Study context	Students and/or student projects, professionals and/or industrial software projects, not relevant
Data sets	Reference to data set applied

The categories were based on categories applied in previous estimation studies, e.g., [1, 2, 4-6] and adapted to the needs of our analysis. The classification schema was developed for the purpose of our review and is not intended to be a general-purpose classification of software effort estimation studies. Note that most of the categories are nonexclusive, e.g., a paper may focus on more than one estimation approach and apply more than one research approach.

The initial classification was performed by the first author of this paper. The robustness of the classification schema and process was evaluated by testing a random sample of 30 papers (about 10% of the total). This classification test was performed by the second author of this paper. The classification test showed that several of the initial descriptions of categories were vague. In fact, there were disagreements over 59 out of the 150 classifications, i.e. 39%. Most of the disagreements were due to recurring issues, e.g., different interpretations regarding how much review of previous studies a paper should include to be classified as a review paper. Fortunately, fewer than 3% of the initial classifications were clearly inconsistent with the category descriptions. In other words, the main problem was the clarity of the descriptions and not so much the initial classification itself. We therefore decided that the initial classification had a degree of accuracy sufficiently high for the purpose of this paper, given that we: 1) clarified the descriptions that led to disagreements (12 of the category descriptions were clarified), and 2) reclassified the papers that belonged to the problematic categories (109 papers were reread and considered for change of classification; 21 reclassifications were made in total). This clarification and reclassification was then completed by the first author.

In spite of this effort to improve the reliability of the classification, it is likely that several of our classifications are subject to discussion, that the descriptions could be improved further, and, that some of the papers are classified incorrectly. However, we believe that on the whole, the current classification is of sufficiently high quality to serve the purpose of our analysis. Furthermore, BESTweb (www.simula.no/BESTweb), the underlying bibliographic database, is publicly available and other researchers are free to reanalyze the data and draw their own conclusions.

2.4 Analysis

The classification of research papers provided a general picture of the characteristics of the software estimation research. This general picture served as a starting point for deeper investigation of findings that seemed, from the authors' perspectives, to suggest important shortcomings in estimation research and possibilities for improvement.

2.5 Threats to Validity

The main threats to validity of our review we have identified are these:

Publication bias: The exclusion of conference papers and reports is based mainly on practical concerns, including work-load, e.g., the problems of identifying all relevant conferences and the amount of analysis needed to handle the fact that many journal papers are improvements of previously published conference papers. This exclusion of conference papers would have been difficult to defend if we had studied a particular estimation phenomenon, e.g., whether the estimation method COCOMO or Function Points have been shown to be more accurate. In that type of study, all relevant papers should be identified and reviewed, regardless of type of source. However, we are interested mainly in properties of the research into software cost estimation. The main bias of our inclusion of journal papers is simply one towards high quality papers. We analyzed informally a selection of software cost estimation papers (about 50) published at conferences and found that the research topics, methods, study designs and study contexts of these papers were similar to those of the journal papers. This suggests that our analyzes and recommendations would not be very different if we had included conference papers. Another potential publication bias is that which might result from *not* publishing

estimation research that (i) has non-significant or company-confidential results, or (ii) that was conducted on topics that do not fit into the common software engineering journals, or (iii) generated results that did not yield the desired outcome. The size and effect of the potential publication biases would be interesting to study, but would require a study design different from ours.

Vested interests of the authors: Both authors of this paper are active researchers on software cost estimation. One of the authors mainly publishes papers on expert estimation and the other on formal, mainly analogy-based, estimation models. We are not aware of biases we may have had when categorizing the papers, but the reader should be aware of the possible impact of our own interests on the analyzes. In particular, it is possible that the recommendations we make are affected strongly by our interests and opinions.

Unfamiliarity with other fields: Clearly, estimation is a topic that is relevant to many fields; thus, it is possible that we have overlooked essential work and relevant journals published in another discipline, e.g., civil engineering.

3 Research Questions, Results, Discussions and Recommendations

Based on what we believed were interesting issues to analyze we posed the ten research questions described in Table 2. The underlying motivation for all questions was our goal of improvement of the software cost estimation research.

Table 2: Research Questions

Research Question	Main Motivation
RQ1: Which journals include papers on software cost estimation?	Support cost estimation researchers with a list of journals with potentially relevant papers.
RQ2: How easy is it to identify relevant software cost estimation journal papers?	Identify possible shortcomings of internet and library-based searches to identify cost estimation papers.
RQ3: To what extent are software cost	Identify the extent to which software cost

estimation researchers aware of the breadth of potential estimation study sources?	estimation researchers identify related work.
RQ4: Which journal is the dominant software cost estimation journal? To what extent does this journal have research topic biases?	Identify the most important software cost estimation journal and the extent to which this journal reflects the totality of software cost estimation research.
RQ5: How many researchers are there who have a long term interest in software cost estimation? To what extent do the interests of these researchers affect the distribution of research topics?	Assess the vulnerability of software cost estimation research; for example, there being few researchers on particular topics may increase vulnerability.
RQ6: What are the most investigated software cost estimation research topics and how has this changed over time?	Background for recommendations on change of research focus (RQ8).
RQ7: What are the most investigated estimation methods and how has this changed over time?	Background for recommendations on change of research focus (RQ8).
RQ8: Is there a need for change of research focus?	Provide recommendations, if necessary, on change of research focus.
RQ9: What are the most frequently applied research methods, and in what study context? How has this changed over time?	Background for recommendations on change of use of research methods (RQ10).
RQ10: What are the main shortcomings regarding use of research methods?	Provide recommendations, if necessary, on change of use of research methods.

3.1 Relevant Software Cost Estimation Research Journals (RQ1)

We found papers on software cost estimation in as many as 76 journals, i.e., the total number of journals with such papers is high, or at least higher than we expected. The 10 journals with five or more papers on software cost estimation are displayed in Table 3, together with the corresponding number, proportion, and cumulative proportions of papers. These 10 journals include two thirds of all identified journal papers on software cost estimation. Reading the 10 most relevant journals *only,* means, however, that important research results may be missed.

Table 3: Most Important Software Cost Estimation Journals

Rank	Journal	Number	Proportion	Cumulative Proportion
1	IEEE Transactions on Software Engineering	51	17%	17%
2	Information and Software Technology	47	15%	32%
3	Journal of Systems and Software	42	14%	46%
4	Empirical Software Engineering	12	4%	50%
5	IEEE Software	11	4%	54%
6	Communications of the ACM	9	3%	57%
7	Software Quality Journal	9	3%	60%
8	American Programmer	6	2%	62%
9	Information and Management	6	2%	64%
10	Journal of Software Maintenance and Evolution	5	2%	66%

3.2 Identification of Relevant Software Cost Estimation Research Journal Papers (RQ2)

Our search for estimation papers was based on a manual issue-by-issue search of about 100 journals. This is, we believe, an accurate method of identifying relevant research papers, given that the people conducting the search possess sufficient expertise. It does, however, require much effort and, if possible, it should be replaced with more automated search and identification methods. The main tool for this is the use of digital libraries. To indicate the power of the digital libraries we conducted the following evaluation:

1) The search term: *"software cost estimation" OR "software effort estimation"* was applied in the digital research libraries Google Scholars (scholar.google.com) and Inspec. Wider searches would obviously lead to more complete searches. The number of "false alarms" would, however also increase strongly and the benefit of automatic search may easily disappear. The search "software" AND ("cost" OR "effort") AND "estimation", for example, led to the identification of about 278 000 records in Google Scholar. A similar example using Inspec is presented later in this section.

2) The papers identified by using the above searches were compared with the set of papers from our manual search.

The main conclusion from this simple test was that the using the search facilities of digital libraries to search for common software cost estimation terms is not sufficient for the identification of all relevant software cost estimation research. The search in Google Scholars (October 2005) resulted in 998 records. However, only 92 out of the 304 of the journal papers were identified, i.e., a coverage rate of about 30% only. The search in Inspec identified 763 journal papers. As expected, Inspec performed better and identified 177 of the 304 papers, i.e., a coverage rate of about 60%. The joint set of Google Scholar and Inspec led to the identification of 204 of the 304 papers, i.e., a coverage rate of almost 70%. Nevertheless, even the use of both libraries missed a substantial part of relevant papers.

A closer examination of the titles and abstracts of the journal papers not identified by Google Scholar or Inspec suggests that the most typical reasons for non-identification in our test were:

- A variety of substitutes for the term "estimation", e.g., "prediction", "scheduling", "cost modeling", "use of costing models", "application of cost estimation techniques", "calculating the cost", and, "use of estimation technology".
- A variety of terms used instead of "software", e.g., "system", "maintenance", "project", and, "task".
- Use of more specific terms derived from particular estimation methods, e.g., "function points", instead of more general estimation terms.
- Studies dealing with specific estimation topics, e.g., studies on the use of accuracy measures or cost interval predictions, may not use the general terms.

In many cases, a software cost estimation researcher will use more specific terms when searching for relevant papers. This may reduce the effect of the last two of the above reasons and higher coverage rates can be expected. However, there remains the problem that a number of synonyms are used for the terms "estimating" and "software". A paper written by one of this paper's authors titled: *Experience with the accuracy of software maintenance task effort prediction models* [7], illustrates this problem. The paper compares different formal estimation models. Assume that a researcher wants to summarize related work on regression-based software cost estimation models. Our paper is clearly relevant for this summary, but turns out to be difficult to identify since we use "prediction" instead of "estimation" and "maintenance task" instead of, e.g., "software development". A wider search, e.g., the search ("software" AND ("prediction OR "estimation")) in Inspec identified the paper. The set of identified records was, however, as high as 11303. This is too large a set to be meaningful for an identification of relevant papers. In addition, if we had used variants of the terms, e.g., "predict" instead of "prediction" we would not have identified the paper in spite of a very wide search. It is evident that searches in digital libraries that are sufficiently wide to identify relevant software cost estimation research can easily lead to higher work-load as purely manual search processes.

Manual searches do not guarantee completeness, either. It is easy to make errors and to miss relevant journals. To illustrate the point, when we conducted this evaluation of Google Scholar and Inspec we found three previously journals with potentially

relevant papers that we had not identified during our manual search. These journals were country- and company-specific IT journals.

In short, the current situation, with a lack of standardized terminology, may require a manual search of titles and abstracts in a carefully selected set of journals to ensure the proper identification of relevant research on software cost estimation. In the longer term, moving to a more standard scheme of classification for software cost estimation, or a more standardized use of keywords, should be an important goal for digital libraries and researchers on software cost estimation.

3.3 Researcher Awareness of Relevant Journals (RQ3)

We were interested in the degree to which software estimation researchers were aware of, and systematically searched for, related research in more than a small set of journals. An indication of this awareness was derived through a random selection of 30 software cost estimation journal papers (about 10% of the total). The reference lists of each of these papers were examined. From this examination we found that:

- The typical (median) software cost estimation study relates its work to and/or builds on cost estimation studies found in only three different journals. We examined the topics of the papers and found several previously published, seemingly relevant, papers on the same research topic that were not referred to. This indicates that many research papers on software cost estimation are based on information derived from a narrow search for relevant papers.
- The most referenced journal, with respect to related cost estimation work, was IEEE Transactions on Software Engineering (IEEE TSE). Estimation papers from this journal were referred to in as many as 70% of the papers. Relative to the relative number of cost estimation papers in each journal, there were surprisingly many references to Communications of the ACM; fully 30% of the papers made references to at least one of the journal's nine papers on software cost estimation. There were, relative to the number of cost estimation papers available, few references to papers in Information and Software Technology (20%), and Software Quality Journal (3%).
- Papers published in Information System (IS) journals, e.g., Information and Management, mainly contained references to other IS journals; while papers

published in Software Engineering (SE) journals mainly contained references to other SE journals. The only exception here was IEEE TSE. This journal was referred to frequently by both communities. Clearly, to communicate software cost estimation results to other researchers from both communities, results should be published in IEEE TSE.

- Few papers referred to estimation results outside the software community, e.g., to studies in forecasting, human decision making and project management. The main proportion of references to sources outside the software community seems to be to literature on statistics.
- We made a separate test on references to two journals outside the software engineering field: the International Journal of Forecasting and the International Journal of Project Management. The former includes many relevant results on how to predict future outcomes (forecasting) and contains papers and results that are highly relevant for many papers on software cost estimation. The latter is the major journal on project management and provides results related to the project context of software cost estimation as well as project cost estimation results. Out of the 30 journal papers, only one referred to the International Journal of Project Management and none to the International Journal of Forecasting! Both journals can be accessed by using digital libraries, e.g., Inspec and Google Scholar, but may be hard to identify due to different use of terminology.

The above evidence suggests that the authors of many software cost estimation papers use criteria that are too narrow when searching for papers that are relevant for describing the context and background of the reported study, as is found, for example, in the "related work" section of a paper. However, the most important issue is whether papers on software cost estimation miss important prior results that would improve the design, analysis or interpretation of the study. Our impression, based on the review presented in this paper and previous experience, is that many papers (but far from all) refer to a substantial subset of relevant results published in major software engineering journals. The major problem is the identification and integration of results from journals in other domains, as exemplified by the lack of reference to papers in the International Journal of Forecasting and the Journal of Project Management.

An example of the incomplete summarisation of related work can be found in one of the authors' own papers [8]. That paper claims that "*A number of researchers have used this type of approach* [estimation by analogy] *with generally quite encouraging results*". This claim was based on references to six studies (out of 20 relevant studies). Of the six studies, three were conducted by one of the authors. Including all relevant studies would have led to less optimism. Out of the 20 studies, nine were in favour of estimation by analogy, four were inconclusive, and seven were in favour of regression-based estimation. An incomplete identification of relevant studies may, consciously or unconsciously, lead to conclusions that are biased towards the researcher's own vested interests [9].

3.4 Most Important Software Cost Estimation Journal (RQ4)

IEEE TSE was found to be the dominant software cost estimation research journal when considering both the number of studies (see Table 3) and citations made by other researchers (see analysis in Section 3.3). It is, therefore, an interesting question whether IEEE TSE has publication biases, e.g., whether it favors certain topics and rejects papers on other topics. A strong publication bias in IEEE TSE could, for example, have the unfortunate consequence of directing software cost estimation researchers' focus towards the topics most easily accepted by the journal. To analyze this we compared the distribution of research topics, estimation approaches and research methods of the IEEE TSE with the total set of estimation papers.

We found that the distributions of IEEE TSE cost estimation topics, estimation methods, and research methods were similar to the corresponding distributions of the total set of papers. IEEE TSE papers had a somewhat stronger focus on “Function Point”-based estimation methods and less focus on “Expert judgment”, but even here the difference was small. Moreover, there may be a time effect, since this topic was more popular in the 1980s and 1990s whilst not all other journals, e.g. Empirical Software Engineering, were publishing during this time. This suggests that the IEEE TSE software cost estimation papers reflect the total set of software cost estimation papers reasonably well. Notice that we have only studied high level types of publication bias and do not exclude the possibility that there are other types of difference regarding journal

publication, for example, differences in the formalism usedwhen describing estimation methods.

3.5 Researchers with a Long-Term Interest in Software Cost Estimation (RQ5)

There seem to be few researchers with a long-term focus on software cost estimation research. We found, for example, only 13 researchers with more than five journal papers on software cost estimation. Most of these researchers are still active; nine of them, for example, published journal papers on software cost estimation in the period 2000-2004.

The potential importance of researchers with a long-term focus on software cost estimation can be illustrated through an analysis of the papers covering the topics "measures of estimation performance" and "data set properties". These topics are basic research on cost estimation necessary for meaningful analyzes and evaluations of estimation methods. The topics require more than average experience and maturity in software cost estimation research. Not surprisingly, almost two thirds of the papers on these topics were authored (or co-authored) by the nine active, long-term focus researchers identified above.

The current number of active researchers with a long-term focus on software cost estimation is low compared to the number of research topics and estimation approaches and there could be a strong research bias towards these researchers' interests, e.g., through the active researchers' positions as supervisors, project leaders and from ideas generated from their research. Fortunately, these researchers seem to cover a wide spectrum of research topics, estimation approaches, and research methods: they publish papers on most topics, on most estimation approaches, and apply most types of research method. Nevertheless, the fact that there are so few active researchers with a long-term focus on software cost estimation both slows the progress of research on several topics and increases the likelihood of error. Several cost estimation topics have been the long-term focus of only one active researcher or research group.

A consequence of the low number of active researchers with a long-term focus on software cost estimation is that a substantial proportion of the research papers have

authors (and reviewers) with a shorter-term focus on software cost estimation, e.g., as part of a Ph.D. The body of knowledge within software cost estimation and related areas is very large and without supporting environments we cannot expect fresh researchers to deliver high-quality research. We recommend that research groups with an interest in software cost estimation should address this challenge, e.g. by developing and using supporting tools such as BESTweb and by co-operation with relevant estimation research expertise in software engineering and other disciplines.

3.6 Distribution of Research Topics (RQ6)

Table 4 shows the distribution of topics for three periods and in total.

Table 4: Research Topics (one paper may discuss more than one topic)

Period	Em	Pf	Cm	Sm	Oi	Un	Ep	Ds	Ot	# papers
-1989	30 (73%)	8 (20%)	3 (7%)	5 (12%)	9 (22%)	2 (5%)	2 (5%)	0 (0%)	0 (0%)	41
1990-1999	96 (59%)	7 (4%)	13 (8%)	39 (24%)	25 (15%)	10 (6%)	8 (5%)	1 (1%)	3 (2%)	163
2000-2004	58 (58%)	3 (3%)	4 (4%)	16 (16%)	14 (16%)	13 (13%)	6 (6%)	2 (2%)	1 (1%)	100
Total	184 (61%)	18 (6%)	20 (7%)	60 (20%)	48 (16%)	25 (8%)	16 (5%)	3 (1%)	4 (1%)	304
The abbreviations used are: Estimation method = Em, Production function = Pf, Calibration of models = Cm, Size measures = Sm, Organizational issues = Oi, Uncertainty assessments = Un, Measures of estimation performance = Ep, Data set properties = Ds, Other = Ot.										

The distribution in Table 4 suggests that:

- The most common research topic, with 61% of the papers, is the introduction and evaluation of estimation methods (Em).
- The distribution of topics over time is quite stable, with a few exceptions. Papers on size measures (mainly function points) seem to have had their heyday in the 1990s,

papers on the production functions were more common before 1990, and there has been an increase in focus on the uncertainty of effort estimates since 2000.

3.7 Distribution of Estimation Methods (RQ7)

Table 5 shows the distribution of papers on different estimation approaches per period and in total.

Table 5: Estimation Approaches (one paper may discuss more than one approach)

Period	Rg	An	Ej	Wb	Fp	Ct	Si	Nn	Th	By	Cb	Ot
-1989	21 (51%)	1 (2%)	3 (7%)	3 (7%)	7 (17%)	0 (0%)	2 (5%)	0 (0%)	20 (49%)	0 (0%)	0 (0%)	2 (5%)
1990-1999	76 (47%)	15 (9%)	22 (13%)	5 (3%)	47 (29%)	5 (3%)	4 (2%)	11 (7%)	14 (9%)	1 (1%)	3 (2%)	7 (4%)
2000-2004	51 (51%)	15 (15%)	21 (21%)	4 (4%)	14 (14%)	9 (9%)	4 (4%)	11 (11%)	5 (5%)	6 (6%)	2 (2%)	16 (16%)
Total	48 (49%)	1 (10%)	6 (15%)	2 (4%)	8 (22%)	4 (5%)	0 (3%)	2 (7%)	9 (13%)	7 (2%)	5 (2%)	5 (8%)

The abbreviations used are: Rg = Regression, An = Analogy, Ej = Expert judgment, Wb = Work break-down, Fp = Function Point, Ct = Classification and regression trees, Si = Simulation, Nn = Neural network, Th = Theory, By = Bayesian, Cb = Combination of estimates, Ot = Other

The distribution in Table 5 suggests that:

- Regression-based estimation approaches dominate. Notice that regression-based estimation approaches include most common parametric estimation models, e.g., the COCOMO model. Roughly half of all estimation papers try to build, improve or compare with regression model-based estimation methods.
- The proportion of papers on analogy and expert judgment-based estimation models is increasing, but still relatively low.

- The heyday of research on the function point-based estimation approaches may have been in the 1990s. In that period, almost one third of the papers tried to improve upon, or make comparisons with, function point-based estimation methods.
- Theory-based estimation approaches, e.g., the SLIM model and Halstead's software science, were initially very popular, but research interest has faded considerably since 1990.
- The diversity of estimation approaches is very high and increasing, especially in the period 2000-2004. This increase is illustrated by the increased proportion of "other approaches" (16% in 2000-2004). "Other approaches" include, for example, the use of lexical analysis of requirement specifications, genetic programming, linear programming, economic production models, soft computing, fuzzy logic modelling, and bootstrap-based analogy cost estimation.

3.8 Is There a Need for Change of Research Focus? (RQ8)

It is, of course, impossible to conduct an objective comparison of the actual distributions in Tables 4 and 5 with the optimal distributions. This would, among other things, require that there was consensus about how the optimal distribution should look. The analysis in this section is, we admit, quite subjective and influenced by the authors' beliefs and opinions.

Our impression, based on the observations of journal publications and other experience as software cost estimation researchers, is that certain changes should be made in research topic and estimation approach focus. In particular, we would like the following changes to occur:

- *More studies on basic estimation issues*: There are many unsolved problems regarding how to compare and evaluate estimation methods, as pointed out in several studies [10-13]. Basic estimation issues that need to be addressed better include the following: how to evaluate estimation method performance, the relation between goodness of estimation approach and dataset properties, how to select between estimation methods, and the mental processes involved in expert estimation of

development costs. As we see it, these issues are frequently prerequisites for the meaningful development, evaluation and implementation of estimation methods.

- *Fewer studies that evaluate methods based on arbitrarily chosen data sets*: Our review suggested that the proposal of a new or improved estimation method is typically followed by an evaluation of the method on an arbitrarily chosen data set. This type of evaluation does not provide much information about other contexts where we should expect use of the method to improve the estimation performance, i.e., when to use it or why we should expect improved estimation performance. The value of this type of evaluation is, at best, limited. There is clearly a need for research on how to evaluate estimation methods and how to make recommendations for use in real-life contexts. This requires more basic research on estimation.
- *More studies on support rather than replacement of expert judgment-based estimation processes*: The estimation approaches most commonly applied by the software industry are, according to several surveys, based on expert judgment (see [14] for a review of the surveys). Currently, expert judgment-based approaches are discussed in relatively few papers (about 15 %). However, this rather contrasts with the fact that the performance of more formal estimation techniques has been somewhat erratic to date. The relative lack of focus upon these expert judgment-based approaches suggests, we believe, that most researchers either do not co-operate closely with the software industry or that they believe it is better to focus on replacement, rather than improvement, of approaches currently employed in industry.

3.9 Research Approach (RQ9)

Table 6 shows the distribution of papers applying different research approaches, per period and in total.

Table 4: Research Approaches

Period	Th	Sv	Ex	Ca	Dm	Hv	Ox	Rl	Re	Si	Ot
-1989	6 (15%)	1 (2%)	1 (2%)	1 (2%)	18 (44%)	22 (54%)	4 (10%)	0 (0%)	5 (12%)	2 (5%)	0 (0%)
1990-1999	6 (4%)	20 (12%)	9 (6%)	3 (2%)	73 (45%)	90 (55%)	9 (6%)	8 (6%)	17 (10%)	8 (5%)	0 (0%)

2000-2004	6 (6%)	6 (6%)	9 (9%)	4 (4%)	50 (50%)	54 (54%)	5 (5%)	3 (3%)	4 (4%)	7 (7%)	2 (2%)
Total	18 (6%)	27 (9%)	19 (6%)	8 (3%)	141 (46%)	166 (55%)	18 (6%)	11 (4%)	26 (9%)	17 (6%)	2 (1%)
The abbreviations used are: Th = Theory, Sv = Survey, Ex = Experiment, Ca = Case study, Dm = Development of estimation method, Hv = History-based evaluation, Ox = Own experience, Rl = Real-life evaluation, Re = Review, Si = Simulation, Ot = Other.											

Table 6 suggests that:

- There are few case studies that focus upon the actual effort estimation process published in the journals, i.e., in-depth studies of software effort estimation processes and outcomes. Overall, we identified only eight such estimation case studies. Even in the IS-journals, which may have a stronger tradition for case studies, estimation case studies seem to be rare. There may be much to learn from well-described and well-understood real-life cases. It may, therefore, be unfortunate that so few estimation case studies are reported in journals.
- Most papers evaluate estimation methods by employing historical data; few evaluations are completed in real-life estimation situations. Of particular concern is that the already small proportion of real-life evaluations of estimation methods seems to be decreasing.

In short, there seems to be a lack of in-depth studies on the actual use of estimation methods and real-life evaluations. We investigated the papers that proposed a new estimation method or evaluated an existing, i.e., those with topic = Em. We found that most of these papers (more than 60%) were of the research approach Hv, i.e., they evaluated an estimation method by applying historical data. This is an understandable choice of evaluation method, given the extra noise added with more realism. However, there is a problem when there are few or no groups that conduct research on the actual use and effect of an estimation method in real-life settings. We believe this is the case for software cost estimation methods. We analyzed all the papers with topic *Em* and research approach *Rl*. We could find no study that had an in-depth data collection and analysis of how estimation methods were actually applied! Even for well known models, such as the COCOMO-model, this type of study was missing. This means, among other things, that

we cannot rule out the possibility that the actual use of formal estimation models is typically a means to disguise expert estimates [15]. In our opinion, this lack of in-depth, real-life evaluation of the use and impact of estimation methods is an important shortcoming of the previous and more recent research on software cost estimation. We cannot claim, based on real-life evidence published in journal papers, that any of our estimation methods yield any benefit in terms of increased accuracy.

3.10 Study Context and Use of Data Sets (RQ10)

Table 7 shows the distribution of papers applying different study contexts per period and in total.

Table 7: Study Contexts

Period	St	Pr	Nr
-1989	1 (2%)	24 (59%)	16 (39%)
1990-1999	9 (6%)	123 (75%)	29 (18%)
2000-2004	11 (11%)	70 (70%)	22 (22%)
Total	21 (7%)	217 (71%)	67 (22%)
The abbreviations used are: St = Students and/or student projects, Pr = Professionals and/or industrial software projects, Nr = Not relevant			

Table 7 suggests that the study context is usually professionals and/or professional projects. This increases the realism of the studies. However, as reported in Section 3.8, this does not necessarily mean that problems with realism are absent. In particular, the dominant use of historical data instead of real-life evaluation results in a decrease in realism. The "Not relevant" category includes papers where there are no empirical studies, e.g., the development of an estimation method without empirical evaluation or analytical comparison of properties of estimation methods.

Evaluation of estimation methods based on historical data sets obviously requires that these data sets be representative for current or future projects; or, at least, that their analysis should be a useful means of evaluating the strengths and benefits of an estimation method. We analyzed the use of historical datasets in the journal papers and

found that there are good reasons to claim that the *availability* of a data set is more indicative for its use than its representativeness or other properties.

The *Boehm-Cocomo* dataset [4], for example, was published in 1981 and is readily available. However, it is based on software development projects that may be quite different from the most recent development projects, e.g., the technology used might well be different. In spite of that, the data set has been used in as many as 12 journal papers to evaluate estimation methods, calibration of methods and estimation model size measures since 1995.

If a data set is not representative it may still be useful for evaluating properties of an estimation method relative to properties of datasets. Unfortunately, we found that very few papers (including most of our own) paid any attention at all to properties of the data set. The evaluations typically pick one or more easily available data sets and leave it to the reader to decide the degree to which it is possible to generalize the results to other contexts and other sets of projects. This is hardly a research method that leads to a systematic aggregation of results.

4 Summary

This paper reviews software cost estimation papers published in journals. Based on the review we provide recommendations for future research as follows:

- **Increase the breadth of the search for relevant studies**: Software cost estimation research studies seem, on average, to be based on searches for relevant previous studies that use too few sources. If the goal is to find the latest research on relevant software cost estimation, it is *not* sufficient to search searches in digital libraries or to conduct manual searches of the most important software engineering journals. We identified as many as 76 journals with studies on software cost estimation, several software cost estimation papers that would have been hard to find through searches in digital libraries, and many journals from other domains that contained relevant results.
- **Search manually for relevant papers in a carefully selected set of journals when completeness is essential**: There is a lack of standardized use of terms pertaining to software cost estimation. We believe that such a lack makes it is easy to miss

important papers when relying on automatic searches in digital libraries. The search term: "software cost estimation" OR "software effort estimation", for example, did not identify more than 60% of the papers on software cost estimation identified by our manual search. We are aware of the practical limitations related to the use of manual search process, e.g., the required search effort. In cases where completeness is not essential, a combination of manual search of the most relevant journals and use of digital libraries to cover the other journals may be sufficient.

- **Conduct more research on basic software cost estimation topics**: We argue that there is insufficient focus on basic software cost estimation topics, e.g., approaches for evaluating methods of software cost estimation. The review describes evidence that connects this with the small number of researchers that have a long-term focus on software cost estimation. In our opinion, a high number of estimation method evaluations (including the authors' own) are based on poor evaluation methods and measures. To make the results from estimation methods more valuable, we should put much more research effort into developing better ways of evaluating estimation methods and of measuring the accuracy of estimations.
- **Conduct more studies of software cost estimation methods in real-life situations**: The proportion of estimation studies where estimation methods are studied or evaluated in real-life situations is low. We could not, for example, find a single study on how software companies actually use formal estimation models. Our knowledge of the performance of formal estimation models is, therefore, limited to laboratory settings (this does not change by using real-life historical data) or real-life settings where we do not know how the models are used.
- **Conduct more studies on estimation methods commonly used by the software industry**: In spite of the fact that formal estimation models have existed for many years, the dominant estimation method is based on expert judgment. Further, available evidence does not suggest that the estimation accuracy improves with use of formal estimation models. Despite these factors, current research on, e.g., expert estimation is sparse and we believe that it deserves more research effort. This includes more research on topics such as the combination of formal models and expert judgment, structuring of the expert-judgment estimation processes, use of

checklists and work-breakdown structures, and understanding of the cognitive steps involved in expert estimation of software costs.

- **Conduct fewer studies that evaluate methods based on arbitrarily chosen data sets**: Currently, a typical evaluation of an estimation method is based on an arbitrarily chosen dataset where the representativeness and other properties are not analyzed or much discussed. Even worse, quite a few papers are based on data sets that are clearly too old to be representative for more recent or future projects. We recommend that researchers on software cost estimation change their focus from the availability of project data sets to understanding the relationship between project characteristics (data set properties) and estimation methods.

References

1. Briand, L.C. and I. Wieczorek, *Resource estimation in software engineering*, in *Encyclopedia of software engineering*, J.J. Marcinak, Editor. 2002, John Wiley & Sons: New York. p. 1160-1196.
2. Walkerden, F. and D.R. Jeffery, *Software cost estimation: A review of models, process, and practice.* Advances in Computers, 1997. **44**: p. 59-125.
3. Boehm, B., C. Abts, and S. Chulani, *Software development cost estimation approaches - A survey.* Annals of Software Engineering, 2000. **10**: p. 177-205.
4. Boehm, B.W., *Software engineering economics*. 1981, New Jersey: Prentice-Hall. XXVII, 767 s.
5. Fairley, R.E. *Recent advances in software estimation techniques*, in *International Conference on Software Engineering*. 1992. Melbourne, Australia: ACM, New York, NY, USA: p. 382-391.
6. Heemstra, F.J., *Software cost estimation.* Information and Software Technology, 1992. **34**(10): p. 627-639.

7. Jørgensen, M., *Experience with the accuracy of software maintenance task effort prediction models.* IEEE Transactions on Software Engineering, 1995. **21**(8): p. 674-681.
8. MacDonell, S.G. and M.J. Shepperd, *Combining techniques to optimize effort predictions in software project management.* Journal of Systems and Software, 2003. **66**(2): p. 91-98.
9. Mair, C. and M. Sheppard. *The consistency of empirical comparisons of regression and analogy-based software project cost prediction*. Forthcoming *in proceedings of ISESE*. 2005. Australia: p.
10. Foss, T., et al., *A simulation study of the model evaluation criterion MMRE.* IEEE Transactions on Software Engineering, 2003. **29**(11): p. 985-995.
11. Shepperd, M., M. Cartwright, and G. Kadoda, *On building prediction systems for software engineers.* Empirical Software Engineering, 2000. **5**(3): p. 175-182.
12. Lo, -.B.-W.-N. and Xiangzhu-Gao, *Assessing software cost estimation models: criteria for accuracy, consistency and regression.* Australian Journal of Information Systems, 1997. **5**(1): p. 30-44.
13. Mair, C. and M. Shepperd. *The Consistency of Empirical Comparisons of Regression and Analogy-based Software Project Cost Prediction*. Forthcoming *in proceedings of the conference ISESE*. 2005. Noosa Heads, Australia: p.
14. Moløkken, K. and M. Jørgensen. *A review of software surveys on software effort estimation*, in *International Symposium on Empirical Software Engineering*. 2003. Rome, Italy: Simula Res. Lab. Lysaker Norway: p. 223-230.
15. Jørgensen, M. and T. Gruschke. *Industrial Use of Formal Software Cost Estimation Models: Expert Estimation in Disguise?* in *EASE*. 2005. Keele, UK: p. 1-7.

APPENDIX 1: List of Included Papers

(Numbers of papers identified in each paper in (). The journals with more than 5 papers in italic)

ACM Transactions on Computer Personnel (1)
Ada User Journal (1)
Advances in Computers (3)
Advances in Information Systems (1)
American Programmer / Cutter IT Journal (6) – Rank 7
Annals of Software Engineering (3)
Applied Computing Review (1)
Australian Journal of Information Systems (3)
Automated Software Engineering (2)
Communications of the ACM (9)
Computers & Operations Research (2)
Computing and Control Engineering Journal (1)
Concurrent Engineering: Research and Applications (1)
Datamation (1)
Embedded Systems Programming (1)
Empirical Software Engineering (12) – Rank 4
Engineering Economist (1)
Engineering Intelligent Systems for Electrical Engineering and Communications (1)
European Journal of Information Systems (3)
Expert Systems (1)
Expert Systems with Applications (2)
GEC Journal of Research (1)
Human Factors (1)
IBM Systems Journal (1)
ICL Technical Journal (1)
IEE Proceedings Software (4)

IEE Proceedings Software Engineering (2)

IEEE Aerospace and Electronic Systems Magazine (1)

IEEE Computer (1)

IEEE Multimedia (1)

IEEE Software (11) – Rank 5

IEEE Transactions on Computers (1)

IEEE Transactions on Software Engineering (51) - Rank 1

IEEE Transactions on Systems, Man and Cybernetic (3)

IIE Transactions (1)

Industrial Management & Data Systems (2)

Information and Management (6) – Rank 6

Information and Software Technology (47) – Rank 2

Information Resources Management Journal (1)

Information Strategy: The Executive's Journal (1)

Information Systems Journal (1)

Information Systems Management (1)

Information Systems Research (1)

Information Technology & Management (1)

International Journal of Project Management (4)

International Journal of Software Engineering and Knowledge Engineering (2)

International Journal of Systems Science (2)

Journal of Computer and Software Engineering (1)

Journal of Computer Information Systems (1)

Journal of Defense Software Engineering (1)

Journal of End User Computing (1)

Journal of Experimental and Theoretical Artificial Intelligence (1)

Journal of Information Technology (3)

Journal of Management Information Systems (3)

Journal of Parametrics (2)

Journal of Software Maintenance and Evolution: Research and Practice (5)

Journal of Systems and Software (42) – Rank 3

Journal of Systems Management (1)

Management Science (4)

MIS Quarterly (4)

New Review of Hypermedia and Multimedia (1)

Pakistan Journal of Information and Technology (1)

Programming and Computer Software (1)

R.F.-Design (1)

Scandinavian Journal of Information Systems (1)

SIGPLAN Notices (1)

Software - Practice and Experience (1)

Software Engineering Journal (4)

Software Engineering Notes (4)

Software Quality Journal (9) – Rank 6

Software World (1)

Technometrics (1)

Texas Instruments Technical Journal (1)

The Australian Computer Journal (1)

Transactions of the Information Processing Society of Japan (1)

Vitro Technical Journal (1)

APPENDIX 2: Journal Papers on Software Cost Estimation

(Search completed April 2004. For a more updated list, that includes more recently published journal papers, see www.simula.no\BESTweb)

References

Abdel-Hamid, T. (1990). "Investigating the cost/schedule trade-off in software development." IEEE Software **7**(1): 97-105.

Abdel-Hamid, T. K. (1993). "Adapting, correcting, and perfecting software estimates: a maintenance metaphor." IEEE Computer **26**(3): 20-29.

Abdel-Hamid, T. K. and S. E. Madnik (1983). "The dynamics of software project scheduling." Communications of the ACM **26**(5): 340-346.

Abdel-Hamid, T. K. and S. E. Madnik (1986). "Impact of schedule estimation on software project behavior." IEEE Software **3**(4): 70-75.

Abdel-Hamid, T. K., K. Sengupta, et al. (1993). "Software project control: An experimental investigation of judgment with fallible information." IEEE Transactions on Software Engineering **19**(6): 603-612.

Abdel-Hamid, T. K., K. Sengupta, et al. (1999). "The impact of goals on software project management: An experimental investigation." MIS Quarterly **23**(4): 531-555.

Abran, A., M. Maya, et al. (1997). "Adapting function points to real-time software." American Programmer **10**(11): 32-43.

Abran, A. and P. N. Robillard (1994). "Function points: A study of their measurement processes and scale transformations." Journal of Systems and Software **25**: 171-184.

Abran, A. and P. N. Robillard (1996). "Function points analysis: an empirical study of its measurement processes." IEEE Transactions on Software Engineering **22**(12): 895-910.

Abran, A., I. Silva, et al. (2002). "Field studies using functional size measurement in building estimation models for software maintenance." Journal of Software Maintenance and Evolution: Research and practice **14**(1): 31-64.

Adrangi, B. and W. Harrison (1987). "Effort estimation in a system development project." Journal of Systems Management **36**(8): 21-23.

Aguilar-Ruiz, J. S., I. Ramos, et al. (2001). "An evolutionary approach to estimating software development projects." Information and Software Technology **43**(14): 875-882.

Ahn, Y., J. Suh, et al. (2003). "The software maintenance project effort estimation model based on function points." Journal of Software Maintenance and Evolution: Research and Practice **15**(2): 71-85.

Albrecht, A. J. and J. E. Gaffney Jr (1983). "Software function, source lines of code, and development effort prediction: A software science validation." IEEE Transactions on Software Engineering **9**(6): 639-648.

Angelis, L. and I. Stamelos (2000). "A simulation tool for efficient analogy based cost estimation." Empirical Software Engineering **5**(1): 35-68.

Antoniol, G., R. Fiutem, et al. (2003). "Object-oriented function points: an empirical validation." Empirical Software Engineering **8**(3): 225-254.

Antoniol, G., C. Lokan, et al. (1999). "A function point-like measure for object-oriented software." Empirical Software Engineering **4**(3): 263-287.

Arifoglu, A. (1993). "A methodology for software cost estimation." Software Engineering Notes **18**(2): 96-105.

Balcom, O. (1998). "Estimating program complexity." Embedded Systems Programming **11**(3): 96-100.

Banker, R. D., H. Chang, et al. (1994). "Evidence on economies of scale in software development." Information and Software Technology **36**(5): 275-282.

Banker, R. D., S. M. Datar, et al. (1993). "Software complexity and maintenance costs." Communications of the ACM **36**(11): 81-94.

Banker, R. D., R. J. Kauffman, et al. (1994). "Automating output size and reuse metrics in a repository-based computer-aided software engineering (CASE) environment." IEEE Transactions on Software Engineering **20**(3): 169-187.

Banker, R. D. and S. A. Slaughter (1997). "A field study of scale economies in software maintenance." Management Science **43**(12): 1709-1725.

Barki, H., S. Rivard, et al. (2001). "An integrative contingency model of software project risk management." Journal of Management Information Systems **17**(4): 37-69.

Barry, E. J., T. Mukhopadhyay, et al. (2002). "Software project duration and effort: an empirical study." Information Technology & Management **3**(1-2): 113-136.

Basili, V. R. and J. Beane (1981). "Can the Parr curve help with manpower distribution and resource estimation problems?" Journal of Systems and Software **2**(1): 59-69.

Basili, V. R. and K. Freburger (1981). "Programming measurement and estimation in the software engineering laboratory." Journal of Systems and Software **2**(1): 47-57.

Behrens, C. A. (1983). "Measuring the productivity of computer systems development activities with function points." IEEE Transactions on Software Engineering **9**(6): 648-652.

Benediktsson, O. and D. Dalcher (2003). "Effort estimation in incremental software development." IEE Proceedings Software Engineering **150**(6): 351-357.

Benediktsson, O., D. Dalcher, et al. (2003). "COCOMO-Based Effort Estimation for Iterative and Incremental Software Development." Software Quality Journal. **11**(4): 265-281.

Benyahia, H. (1996). "Costs and productivity estimation in computer engineering economics." Engineering Economist **41**(3): 229-241.

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

Betteridge, R. (1992). "Successful experience of using function points to estimate project costs early in the life-cycle." Information and Software Technology **34**(10): 655-658.

Bielak, J. (2000). "Improving size estimates using historical data." IEEE Software **17**(6): 27-35.

Boehm, B., C. Abts, et al. (2000). "Software development cost estimation approaches - A survey." Annals of Software Engineering **10**: 177-205.

Boehm, B., B. Clark, et al. (1996). "The COCOMO 2.0 software cost estimation model: A status report." American Programmer **9**(7): 2-17.

Boehm, B. and K. Sullivan (1999). "Software economics: Status and prospects." Information and Software Technology **41**(14): 937-946.

Boehm, B. W. (1984). "Software engineering economics." IEEE Transactions on Software Engineering **10**(1): 4-21.

Boehm, B. W., B. Clark, et al. (1995). "Cost models for future software life cycle processes: COCOMO 2.0." Annals of Software Engineering **1**: 57-94.

Boehm, B. W. and P. N. Papaccio (1988). "Understanding and controlling software costs." IEEE Transactions on Software Engineering **14**(10): 1462-1477.

Boehm, B. W. and R. W. Wolverton (1980). "Software cost modeling: Some lessons learned." Journal of Systems and Software **1**: 195-201.

Bollinger, T. B. and S. L. Pfleeger (1990). "Economics of reuse: Issues and alternatives." Information and Software Technology **32**(10): 643-652.

Bowden, P., M. Hargreaves, et al. (2000). "Estimation support by lexical analysis of requirements documents." Journal of Systems and Software **51**(2): 87-98.

Bradley, M. and R. Dawson (1999). "Whole life cost: The future trend in software development." Software Quality Journal **8**(2): 121-131.

Briand, L. C., V. R. Basili, et al. (1992). "A pattern recognition approach for software engineering data analysis." IEEE Transactions on Software Engineering **18**(11): 931-942.

Briand, L. C. and J. Wüst (2001). "Modeling development effort in object-oriented systems using design properties." IEEE Transactions on Software Engineering **27**(11): 963-986.

Burgess, C. J. and M. Lefley (2001). "Can genetic programming improve software effort estimation? A comparative evaluation." Information and Software Technology **43**(14): 863-873.

Card, D. N. and W. W. Agresti (1987). "Resolving the software science anomaly." Journal of Systems and Software **7**(1): 29-35.

Carmel, E. (1995). "Time-to-completion factors in packaged software development." Information and Software Technology **37**(9): 515-520.

Carson, -. C. (1996). "Using the TQC problem-solving process to develop an improved estimation technique." Texas Instruments Technical Journal **13**(6): 101-106.

Chan, T., S. L. Chung, et al. (1996). "An economic model to estimate software rewriting and replacement times." IEEE Transactions on Software Engineering **22**(8): 580 - 598.

Chang, C. K., M. J. Christensen, et al. (2001). "Genetic algorithms for project management." Annals of Software Engineering **11**: 107-139.

Chatzoglou, P. D. and L. A. Macaulay (1996). "A review of existing models for project planning and estimation and the need for a new approach." International Journal of Project Management **14**(3): 173-183.

Chatzoglou, P. D. and L. A. Macaulay (1998). "A rule-based approach to developing software development prediction models." Automated Software Engineering **5**(2): 211-243.

Chrysler, E. (1978). "Some basic determinants of computer programming productivity." Communications of the ACM **21**(6): 472-483.

Chulani, S., B. Boehm, et al. (1999). "Bayesian analysis of empirical software engineering cost models." IEEE Transactions on Software Engineering **25**(4): 573-583.

Cockcroft, S. (1996). "Estimating CASE development size from outline specifications." Information and Software Technology **38**(6): 391-399.

Cole, A. (1995). "Runaway projects: Cause and effects." Software World **26**(3): 3-5.

Connolly, T. and D. Dean (1997). "Decomposed versus holistic estimates of effort required for software writing tasks." Management Science **43**(7): 1029-1045.

Constantine, G. and W. J. Vitaliano (2001). "Structured estimating - A new approach to a critical project parameter." R.F.-Design **24**(5): 60-68.

Conte, S. D., H. E. Dunsmore, et al. (1985). "Software effort estimation and productivity." Advances in Computers **24**: 1-60.

Costagliola, G., F. Ferrucci, et al. (2000). "A metric for the size estimation of object-oriented graphical user interfaces." International Journal of Software Engineering and Knowledge Engineering **10**(5): 581-603.

Cuelenaere, A. M. E., M. J. I. M. Genuchten, et al. (1987). "Calibrating a software cost estimation model: Why and how." Information and Software Technology **29**(10): 558-567.

Czuchra, W. (1999). "Optimizing budget spendings for software implementation and testing." Computers & Operations Research **26**(7): 731-747.

Day, R. (1999). "Software project estimation: experiences from a large scale object-oriented Ada project." Ada User Journal **20**(2): 154-161.

de Souza Lima Júnior, O., P. P. M. Farias, et al. (2003). "Fuzzy Modeling for Function Points Analysis." Software Quality Journal. **11**(2): 149-166.

Dolado, J. J. (1997). "A study of the relationships among Albrecht and Mark II Function Points, lines of code 4GL and effort." Journal of Systems and Software **37**(2): 161-172.

Dolado, J. J. (2000). "A validation of the component-based method for software size estimation." IEEE Transactions on Software Engineering **26**(10): 1006-1021.

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

Dutta, S., L. N. v. Wassenhove, et al. (1998). "Benchmarking European software management practices." Communications of the ACM **41**(6): 77-86.

Ebrahimi, N. B. (1999). "How to improve the calibration of cost models." IEEE Transactions on Software Engineering **25**(1): 136-140.

Edwards, J. S. and T. T. Moores (1994). "A conflict between the use of estimating and planning tools in the management of information systems." European Journal of Information Systems **3**(2): 139-147.

Evanco, W. M. (1995). "Modeling the effort to correct faults." Journal of Systems and Software **29**(1): 75 - 84.

Ferens, D. and D. Christensen (1999). "Calibrating software cost models to department of defense databases: a review of ten studies." Journal of Parametrics **XIV**(1): 33-51.

Ferens, D. V. (1984). "Software support cost models: quo vadis?" Journal of Parametrics **4**(4): 64-99.

Ferens, D. V. (1996). "Software cost estimation in the DoD environment." American Programmer **9**(7): 28-34.

Ferens, D. V. (1999). "The conundrum of software estimation models." IEEE Aerospace and Electronic Systems Magazine **14**(3): 23-29.

Finnie, G. R., G. E. Wittig, et al. (1997a). "A comparison of software effort estimation techniques: Using function points with neural networks, case-based reasoning and regression models." Journal of Systems and Software **39**(3): 281-289.

Finnie, G. R., G. E. Wittig, et al. (1997b). "Reassessing function points." Australian Journal of Information Systems **4**(2): 39-45.

Fleck, R. A., Jr. (1998). "Managing programmer resources in a maintenance environment with function points." Industrial Management + Data Systems **98**(2): 63-70.

Foss, T., E. Stensrud, et al. (2003). "A simulation study of the model evaluation criterion MMRE." IEEE Transactions on Software Engineering **29**(11): 985-995.

Francalanci, C. (2001). "Predicting the implementation effort of ERP projects: empirical evidence on SAP/R3." Journal of Information Technology **16**(1): 33-48.

Genuchten, M. v. and H. Koolen (1991). "On the use of software cost models." Information and Management **21**: 37-44.

Gilb, T. (1986). "Estimating software attributes: some unconventional points of view." Software Engineering Notes **11**(1): 49-59.

Gonzales, M. and S. Paravastu (2000). "Correction on the software science length estimator skewness for 'C' language programs." Empirical Software Engineering **5**(2): 155-159.

Goodman, P. A. (1992). "Application of cost-estimation techniques: Industrial perspective." Information and Software Technology **34**(6): 379-382.

Granju-Alvarez, J. C. and M. J. Barranco-Garcia (1997). "A method for estimating maintenance cost in a software project: A case study." Journal of Software Maintenance and Evolution: Research and Practice **9**(3): 161-175.

Graves, T. L. and A. Mockus (2001). "Identifying productivity drivers by modeling work units using partial data." Technometrics **43**(2): 168-179.

Gray, A. and S. MacDonell (1999). "Software metrics data analysis-exploring the relative performance of some commonly used modeling techniques." Empirical Software Engineering **4**: 297-316.

Gray, A. R. and S. G. MacDonell (1997). "A comparison of techniques for developing predictive models of software metrics." Information and Software Technology **39**(6): 425-437.

Gulezian, R. (1991). "Reformulating and calibrating COCOMO." Journal of Systems and Software **16**(3): 235-242.

Hakuta, M., F. Tone, et al. (1997). "A software size estimation model and its evaluation." Journal of Systems and Software **37**(3): 253-263.

Hall, B., G. Orr, et al. (2001). "A technique for function block counting." Journal of Systems and Software **57**(3): 217-220.

Halstead, M. H. (1979). "Advances in software science." Advances in Computers **18**: 119-172.

Harrison, W. and B. Adrangi (1987). "The role of programming language in estimating software development costs." Journal of Management Information Systems **3**(3): 101-110.

Hastings, T. E. and A. S. M. Sajeev (2001). "A vector-based approach to software size measurement and effort estimation." IEEE Transactions on Software Engineering **27**(4): 337-350.

Haynes, P. and B. Henderson-Sellers (1996). "Cost estimation of OO projects: empirical observations, practical applications." American Programmer **9**(7): 35-41.

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

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

Heiat, A. (2002). "Comparison of artificial neural network and regression models for estimating software development effort." Information and Software Technology **44**(15): 911-922.

Heiat, A. and N. Heiat (1997). "A model for estimating efforts required for developing small-scale business applications." Journal of Systems and Software **39**(1): 7-14.

Helander, M. E., Z. Ming, et al. (1998). "Planning models for software reliability and cost." IEEE Transactions on Software Engineering **24**(6): 420-434.

Hill, J., L. C. Thomas, et al. (2000). "Experts' estimates of task durations in software development projects." International Journal of Project Management **18**(1): 13-21.

Horgan, G., S. Khaddaj, et al. (1998). "Construction of an FPA-type metric for early lifecycle estimation." Information and Software Technology **40**(8): 409-415.

Hu, Q. (1997). "Evaluating alternative software production functions." IEEE Transactions on Software Engineering **23**(6): 379-387.

Hu, Q., R. T. Plant, et al. (1998). "Software cost estimation using economic production models." Journal of Management Information Systems **15**(1): 143-163.

Huang, S. J. (2001). "Early project estimation in the formal communication protocol development." Information and Management **38**(7): 449-458.

Hughes, R. T. (1996). "Expert judgement as an estimating method." Information and Software Technology **38**(2): 67-75.

Hughes, R. T., A. Cunliffe, et al. (1998). "Evaluating software development effort model-building techniques for application in a real-time telecommunications environment." IEE Proceedings Software **145**(1): 29-33.

Humphrey, W. S. (1996). "The PSP and personal project estimating." American Programmer **9**(6): 2-15.

Humphrey, W. S. and N. D. Singpuwally (1991). "Predicting (individual) software productivity." IEEE Transactions on Software Engineering **17**(2): 196-207.

Höst, M. and C. Wohlin (1997). "A subjective effort estimation experiment." Information and Software Technology **39**(11): 755-762.

Idri, A., T. M. Khoshgoftaar, et al. (2002). "Investigating soft computing in case-based reasoning for software cost estimation." Engineering Intelligent Systems for Electrical Engineereing and Communications **10**(3): 147-157.

Jeffery, D. R. (1987). "Time-sensitive cost models in the commercial MIS environment." IEEE Transactions on Software Engineering **13**(7): 852-859.

Jeffery, D. R. and G. Low (1990). "Calibrating estimation tools for software development." Software Engineering Journal **5**(4): 215-221.

Jeffery, D. R., G. C. Low, et al. (1993). "A comparison of function point counting techniques." IEEE Transactions on Software Engineering **19**(5): 529-532.

Jeffery, D. R., M. Ruhe, et al. (2000). "A comparative study of two software development cost modeling techniques using multi-organizational and company-specific data." Information and Software Technology **42**(14): 1009-1016.

Jeffery, R. and L. Graham (1997). "Function points and their use." The Australian Computer Journal **29**(4): 148 - 156.

Jeffery, R. and J. Stathis (1996). "Function point sizing: Structure, validity and applicability." Empirical Software Engineering **1**(1): 11-30.

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

Jenson, R. L. and J. W. Bartley (1991). "Parametric estimation of programming effort: An object-oriented model." Journal of Systems and Software **15**(2): 107-114.

Johnson, P. M., C. A. Moore, et al. (2000). "Empirically guided software effort guesstimation." IEEE Software **17**(6): 51-56.

Jongmoon, B., B. Boehm, et al. (2002). "Disaggregating and calibrating the CASE tool variable in COCOMO II." IEEE Transactions on Software Engineering **28**(11): 1009-1022.

Jun, E. S. and J. K. Lee (2001). "Quasi-optimal case-selective neural network model for software effort estimation." Expert Systems with Applications **21**(1): 1-14.

Jung, H.-W., Y.-W. Ahn, et al. (2000). "Optimal schedule and effort considering the penalty cost of schedule delay in software development." International Journal of Systems Science **31**(1): 91-95.

Jørgensen, M. (1995). "Experience with the accuracy of software maintenance task effort prediction models." IEEE Transactions on Software Engineering **21**(8): 674-681.

Jørgensen, M. (2003). "How much does a vacation cost? or What is a software cost estimate?" Software Engineering Notes **28**(6): 5-5.

Jørgensen, M. (2004a). "Realism in assessment of effort estimation uncertainty: It matters how you ask." IEEE Transactions on Software Engineering **30**(4): 209-217.

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

Jørgensen, M. (2004c). "Top-down and bottom-up expert estimation of software development effort." Information and Software Technology **46**(1): 3-16.

Jørgensen, M., U. Indahl, et al. (2003). "Software effort estimation by analogy and "regression toward the mean"." Journal of Systems and Software **68**(3): 253-262.

Jørgensen, M. and D. I. K. Sjøberg (2001a). "Impact of effort estimates on software project work." Information and Software Technology **43**(15): 939-948.

Jørgensen, M. and D. I. K. Sjøberg (2001b). "Software process improvement and human judgement heuristics." Scandinavian Journal of Information Systems **13**: 99-121.

Jørgensen, M. and D. I. K. Sjøberg (2003). "An effort prediction interval approach based on the empirical distribution of previous estimation accuracy." Information and Software Technology **45**(3): 123-136.

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

Jørgensen, M., K. H. Teigen, et al. (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.

Kalipsiz, Y. and S. Kurnaz (2002). "Adapting function point analysis to object oriented systems." Pakistan Journal of Information and Technology **1**(1): 5-11.

Kamatar, J. and W. Hayes (2000). "An experience report on the personal software process." IEEE Software **17**(6): 85-89.

Kaplan, H. T. (1991). "The Ada COCOMO cost estimating model and VASTT development estimates vs. actuals." Vitro Technical Journal **9**(1): 48-60.

Kappelman, L. A., D. Fent, et al. (1998). "Calculating the cost of year-2000 compliance." Communications of the ACM **41**(2): 30-39.

Keil, M., L. Wallace, et al. (2000). "An investigation of risk perception and risk propensity on the decision to continue a software development project." Journal of Systems and Software **53**(2): 145-157.

Keller-McNulty, S., M. S. McNulty, et al. (1991). "Stochastic models for software science." Journal of Systems and Software **16**(1): 59-68.

Kemerer, C. F. (1987). "An empirical validation of software cost estimation models." Communications of the ACM **30**(5): 416-429.

Kemerer, C. F. (1993). "Reliability of function points measurement: A field experiment." Communications of the ACM **36**(2): 85-97.

Kemerer, C. F. and B. S. Porter (1992). "Improving the reliability of function point measurement: an empirical study." IEEE Transactions on Software Engineering **18**(11): 1011-1024.

Kirsopp, C. and M. Sheppard (2002). "Making inferences with small numbers of training sets." IEE Proceedings Software Engineering **149**(5): 123-130.

Kitchenham, B. (1992). "Empirical studies of assumptions that underlie software cost-estimation models." Information and Software Technology **34**(4): 211-218.

Kitchenham, B. (1998). "A procedure for analyzing unbalanced datasets." IEEE Transactions on Software Engineering **24**(4): 278-301.

Kitchenham, B. and S. Linkman (1997). "Estimates, uncertainty, and risk." IEEE Software **14**(3): 69-74.

Kitchenham, B., S. L. Pfleeger, et al. (2002). "An empirical study of maintenance and development estimation accuracy." Journal of Systems and Software **64**(1): 57-77.

Kitchenham, B. and N. R. Taylor (1985). "Software project development cost estimation." Journal of Systems and Software **5**(4): 267-278.

Kitchenham, B. A. (2002). "The question of scale economies in software—why cannot researchers agree?" Information and Software Technology **44**(1): 13-24.

Kitchenham, B. A., L. M. Pickard, et al. (2003). "Modeling software bidding risks." IEEE Transactions on Software Engineering **29**(6): 542-554.

Kitchenham, B. A., L. M. Pickard, et al. (2001). "What accuracy statistics really measure." IEE Proceedings Software **148**(3): 81-85.

Kitchenham, B. A. and N. R. Taylor (1984). "Software cost models." ICL Technical Journal **4**(1): 73-102.

Koch, S. and G. Schneider (2002). "Effort, co-operation and co-ordination in an open source software project: GNOME." Information Systems Journal **12**(1): 27-42.

Kusters, R. J., M. J. I. M. Genuchten, et al. (1990). "Are software cost-estimation models accurate?" Information and Software Technology **32**(3): 187-190.

Kusumoto, S., O. Mizuno, et al. (2001). "Software project simulator for effective process improvement." Transactions of the Information Processing Society of Japan **42**(3): 396-408.

Känsälä, K. (1997). "Integrating risk assessment with cost estimation." IEEE Software **14**(3): 61-67.

Lanza, R. B. (2003). "Getting to realistic estimates and project plans: a Monte Carlo approach." Information Strategy: The Executive's Journal **19**(4): 26-34.

Laranjeira, L. A. (1990). "Software size estimation of object-oriented systems." IEEE Transactions on Software Engineering **16**(5): 510-522.

Larsen, K. R. T. and P. A. Bloniarz (2000). "A cost and performance model for web service investment." Communications of the ACM **43**(2): 109-116.

Lassez, J.-L., D. van der Knijff, et al. (1981). "A critical examination of software science." Journal of Systems and Software **2**(2): 105-112.

Lavazza, L. and G. Valetto (2000). "Requirements-based estimation of change costs." Empirical Software Engineering **5**(3): 229-243.

Lederer, A. L., R. Mirani, et al. (1990). "Information system cost estimating: a management perspective." MIS Quarterly **14**(2): 159-176.

Lederer, A. L. and J. Prasad (1991). "The validation of a political model of information systems development cost estimating." ACM Transactions on Computer Personnel **13**(2): 47-57.

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

Lederer, A. L. and J. Prasad (1993a). "Information systems software cost estimating: a current assessment." Journal of Information Technology **8**(1): 22-33.

Lederer, A. L. and J. Prasad (1993b). "Systems development and cost estimating. Challenges and guidelines." Information Systems Management **10**(4): 37-41.

Lederer, A. L. and J. Prasad (1995a). "Causes of inaccurate software development cost estimates." Journal of Systems and Software **31**(2): 125-134.

Lederer, A. L. and J. Prasad (1995b). "Perceptual congruence and systems development cost estimation." Information Resources Management Journal **8**(4): 16-27.

Lederer, A. L. and J. Prasad (1998). "A causal model for software cost estimating error." IEEE Transactions on Software Engineering **24**(2): 137-148.

Lederer, A. L. and J. Prasad (2000). "Software management and cost estimating error." Journal of Systems and Software **50**(1): 33-42.

Lee, A., C. Chun Hung, et al. (1998). "Software development cost estimation: integrating neural network with cluster analysis." Information and Management **34**(1): 1-9.

Lee, H. (1993). "A structured methodology for software development effort prediction using the analytic hierarchy process." Journal of Systems and Software **21**(2): 179-186.

Leung, H. K. N. (2002). "Estimating maintenance effort by analogy." Empirical Software Engineering **7**(2): 157 - 175.

Lewis, J. P. (2001). "Limits to software estimation." Software Engineering Notes **26**(4): 54-59.

Lind, M. R. and J. M. Sulek (1998). "Undersizing software systems: third versus fourth generation software development." European Journal of Information Systems **7**(4): 261-268.

Lind, M. R. and J. M. Sulek (2000). "A methodology for forecasting knowledge work projects." Computers & Operations Research **27**(11-12): 1153-1169.

Lo, -. B.-W.-N. and Xiangzhu-Gao (1997). "Assessing software cost estimation models: criteria for accuracy, consistency and regression." Australian Journal of Information Systems **5**(1): 30-44.

Lokan, C. J. (1996). "Early size prediction for C and PASCAL programs." Journal of Systems and Software **32**(1): 65-72.

Lokan, C. J. (2000). "An empirical analysis of function point adjustment factors." Information and Software Technology **42**(9): 649-659.

Lokan, C. J. and S. D. Conte (1995). "Size prediction for Pascal programs." Journal of Computer and Software Engineering **3**(4): 377-394.

Londeix, B. (1995). "Deploying realistic estimation (field situation analysis)." Information and Software Technology **37**(12): 655-670.

Low, G. C. and D. R. Jeffery (1990). "Function points in the estimation and evaluation of the software process." IEEE Transactions on Software Engineering **16**(1): 64-71.

MacDonell, S. G. (1994). "Comparative review of functional complexity assessment methods for effort estimation." Software Engineering Journal **9**(3): 107-116.

MacDonell, S. G. (1997). "Establishing relationships between specification size and software process effort in CASE environments." Information and Software Technology **39**(1): 35-45.

MacDonell, S. G. (2003). "Software source code sizing using fuzzy logic modeling." Information and Software Technology **45**(7): 389-404.

MacDonell, S. G., T. Fletcher, et al. (1999). "Industry practices in project management for multimedia information systems." International Journal of Software Engineering and Knowledge Engineering **9**(6): 801-815.

MacDonell, S. G. and M. J. Shepperd (2003). "Combining techniques to optimize effort predictions in software project management." Journal of Systems and Software **66**(2): 91-98.

Madachy, R. J. (1995). "Knowledge-based risk assessment and cost estimation." Automated Software Engineering **2**(3): 219-230.

Madachy, R. J. (1997). "Heuristic risk assessment using cost factors." IEEE Software **14**(3): 51-59.

Mair, C., G. Kadoda, et al. (2000). "An investigation of machine learning based prediction systems." Journal of Systems and Software **53**(1): 23-29.

Marouane, R. and A. Mili (1989). "Economics of software project management in Tunisia: basic TUCOMO." Information and Software Technology **31**(5): 251-257.

Marshall, I. M., S. Price, et al. (1997). "Code-based analysis of the development effort of a large scale courseware project." Information and Software Technology **39**(8): 541-549.

Marshall, I. M., W. B. Samson, et al. (1994). "Predicting the development effort of multimedia courseware." Information and Software Technology **36**(5): 251-258.

Marwane, R. and A. Mili (1991). "Building tailor-made software cost model: Intermediate TUCOMO." Information and Software Technology **33**(3): 232-238.

Matson, J. E., B. E. Barrett, et al. (1994). "Software development cost estimation using function points." IEEE Transactions on Software Engineering **20**(4): 275-287.

Matson, J. E. and J. M. Mellichamp (1993). "An object-oriented tool for function point analysis." Expert Systems **10**(1): 3-14.

Maxwell, -. K.-D., -. L. Van-Wassenhove, et al. (1996). "Software development productivity of European space, military, and industrial applications." IEEE Transactions on Software Engineering **22**(10): 706-718.

Maxwell, -. K., -. L. van-Wassenhove, et al. (1999). "Performance evaluation of general and company specific models in software development effort estimation." Management Science **45**(6): 787-803.

McCulla, P. (1989). "The estimating process (IT project management)." International Journal of Project Management **7**(1): 36-38.

Mendes, E., S. Counsell, et al. (2002). "Web hypermedia cost estimation: further assessment and comparison of cost estimation modelling techniques." New Review of Hypermedia and Multimedia **8**: 199-229.

Mendes, E., N. Mosley, et al. (2001). "Web metrics - estimating design and authoring effort." IEEE Multimedia **8**(1): 50-57.

Mendes, E., N. Mosley, et al. (2002). "Comparison of Web size measures for predicting Web design and authoring effort." IEE Proceedings Software **149**(3): 86-92.

Mendes, E., I. Watson, et al. (2003). "A Comparative Study of Cost Estimation Models for Web Hypermedia Applications." Empirical Software Engineering **8**(2): 163-196.

Miranda, E. (2001). "Improving subjective estimates using paired comparisons." IEEE Software **18**(1): 87 - 91.

Misic, V. B. and D. N. Tesic (1998). "Estimation of effort and complexity: An object-oriented case study." Journal of Systems and Software **41**(2): 133-143.

Miyazaki, Y., A. Takanou, et al. (1991). "Method to estimate parameter values in software prediction models." Information and Software Technology **33**(3): 239-243.

Miyazaki, Y., M. Terakado, et al. (1994). "Robust regression for developing software estimation models." Journal of Systems and Software **27**(1): 3-16.

Mizuno, O., T. Kikuno, et al. (2000). "Statistical analysis of deviation of actual cost from estimated cost using actual project data." Information and Software Technology **42**(7): 465-473.

Mohanty, S. N. (1981). "Software cost estimation: Present and future." Software - Practice and Experience **11**(2): 103-121.

Moores, T. T. (2001). "Developing a software size model for rule-based systems: A case study." Expert Systems with Applications **21**(4): 229-237.

Morgan, M. J. (1994). "Controlling software development costs." Industrial Management & Data Systems **94**(1): 13-18.

Moser, S., B. Henderson-Sellers, et al. (1999). "Cost estimation based on business models." Journal of Systems and Software **49**(1): 33-42.

Moses, J. (2002). "Measuring Effort Estimation Uncertainty to Improve Client Confidence." Software Quality Journal **10**: 135-148.

Moses, J. and M. Farrow (2003). "A Procedure for Assessing the Influence of Problem Domain on Effort Estimation Consistency." Software Quality Journal. **11**(4): 283-300.

Mukhopadhyay, T. and S. Kekre (1992). "Software Effort Models for Early Estimation of Process Control Applications." IEEE Transactions on Software Engineering **18**(10): 915-924.

Mukhopadhyay, T., S. S. Vicinanza, et al. (1992). "Examining the feasibility of a case-based reasoning model for software effort estimation." MIS Quarterly **16**(2): 155-171.

Murali, C. S. and C. S. Sankar (1997). "Issues in estimating real-time data communications software projects." Information and Software Technology **39**(6): 399-402.

Musilek, P., W. Pedrycs, et al. (2000). "Software cost estimation with fuzzy models." Applied Computing Review **8**(2): 24-29.

Myrtveit, I. and E. Stensrud (1999). "A controlled experiment to assess the benefits of estimating with analogy and regression models." IEEE Transactions on Software Engineering **25**(4): 510-525.

Myrtveit, I., E. Stensrud, et al. (2001). "Analyzing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods." IEEE Transactions on Software Engineering **27**(11): 999-1013.

Navlakha, J. K. (1990). "Choosing a software cost estimation model for your organization: A case study." Information and Management **18**(5): 255-261.

Nesi, P. and T. Querci (1998). "Effort estimation and prediction of object-oriented systems." Journal of Systems and Software **42**(1): 89-102.

O'Brien, S. J. and D. A. Jones (1993). "Function points in SSADM." Software Quality Journal **2**(1): 1-11.

Ohlsson, N., C. Wohlin, et al. (1998). "A project effort estimation study." Information and Software Technology **40**(14): 831-839.

Ooi, G. and C. Soh (2003). "Developing an activity-based costing approach for system development and implementation." Advances in Information Systems **34**(3): 54-71.

Oriogun, P. K. (1999). "A survey of Boehm's work on the spiral models and COCOMO II - Towards software development process quality improvment." Software Quality Journal **8**: 53-62.

Orr, G. and T. E. Reeves (2000). "Function point counting: One program's experience." Journal of Systems and Software **53**(3): 239-244.

Parr, F. N. (1980). "An alternative to the Rayleigh curve model for software development effort." IEEE Transactions on Software Engineering **6**(3): 291-296.

Pendharkar, P. C. and G. H. Subramanian (2002). "Connectionist models for learning, discovering, and forecasting software effort: an empirical study." Journal of Computer Information Systems **43**(1): 7-14.

Pengelly, A. (1995). "Performance of effort estimating techniques in current development environments." Software Engineering Journal **10**(5): 162-170.

Pfleeger, S. L. (1991). "Model of software effort and productivity." Information and Software Technology **33**(3): 224-231.

Pfleeger, S. L. and T. B. Bollinger (1994). "The economics of reuse: New approaches to modelling and assessing cost." Information and Software Technology **36**(8): 475-484.

Pham, H. and H. Wang (2001). "A quasi-renewal process for software reliability and testing costs." IEEE Transactions on Systems, Man & Cybernetics, Part A Systems & Humans **31**(6): 623-631.

Pham, H. and X. Zhang (1999). "A software cost model with warranty and risk costs." IEEE Transactions on Computers **48**(1): 71-75.

Pickard, L., B. Kitchenham, et al. (2001). "Using simulated data sets to compare data analysis techniques used for software cost modelling." IEE Proceedings Software **148**(6): 165-174.

Pillai, K. and N. Sukumaran (1997). "A model for software development effort and cost estimation." IEEE Transactions on Software Engineering **23**(8): 485-497.

Prechelt, L. and B. Unger (2000). "An experiment measuring the effects of personal software process (PSP) training." IEEE Transactions on Software Engineering **27**(5): 465-472.

Prechelt, L., B. Unger, et al. (2003). "A controlled experiment on inheritance depth as a cost factor for code maintenance." Journal of Systems and Software **65**(2): 115-126.

Prietula, M. J., S. S. Vicinanza, et al. (1996). "Software-effort estimation with a case-based reasoner." Journal of Experimental and theoretical artificial intelligence **8**(3-4): 341-363.

Prokhorov, V. V. and D. V. Smirnov (2001). "A model of estimation and optimization of expenditures in multilevel fuzzy environments." Programming and Computer Software **27**(5): 252-259.

Putnam, L. H. (1978). "A general empirical solution to the macro software sizing and estimating problem." IEEE Transactions on Software Engineering **SE - 4**(4): 345-360.

Putnam, L. H. and A. Fitzsimmons (1979). "Estimating software costs." Datamation **25**(11): 171-178.

Putnam, L. H., D. T. Putnam, et al. (1996). "Adapting project estimation to advancing technologies." American Programmer **9**(6): 23-29.

Rask, R., P. Laamanen, et al. (1993). "Simulation and comparison of Albrecht's function point and DeMarco's function bang metrics in a CASE environment." IEEE Transactions on Software Engineering **19**(7): 661-671.

Ratcliff, B. and A. L. Rollo (1990). "Adapting function point analysis to Jackson system development." Software Engineering Journal **5**(1): 79-84.

Reifer, D. J. (1990). "ASSET-R: A function point sizing tool for scientific and real-time systems." Journal of Systems and Software **11**(3): 159-172.

Reifer, D. J. (2000). "Web development: estimating quick-to-market software." IEEE Software **17**(6): 57-64.

Ruiz, M., I. Ramos, et al. (2001). "A simplified model of software project dynamics." Journal of Systems and Software **59**(3): 299-309.

Rush, C. and R. Roy (2001). "Expert judgement in cost estimating: modelling the reasoning process." Concurrent Engineering: Research and Applications **9**(4): 271 - 284.

Rutherford, J. (1995). "Software estimating and metrics-a pragmatic approach." GEC Journal of Research **12**(2): 66-75.

Samson, B., D. Ellison, et al. (1997). "Software cost estimation using an Albus perceptron (CMAC)." Information and Software Technology **39**(1): 55-60.

Schneider, V. (1978). "Prediction of software effort and project duration-four new formulas." SIGPLAN Notices **13**(6): 49-59.

Schooff, R. M. and Y. Y. Haimes (1999). "Dynamic multistage software estimation." IEEE Transactions on Systems, Man, and Cybernetics **29**(2): 272-284.

Sengupta, K. and T. K. Abdel-Hamid (1996). "The impact of unreliable information on the management of software projects: A dynamic decision perspective." IEEE Transactions on Systems, Man, and Cybernetics **26**(2): 177-189.

Shepperd, M. and M. Cartwright (2001). "Predicting with sparse data." IEEE Transactions on Software Engineering **27**(11): 987-998.

Shepperd, M., M. Cartwright, et al. (2000). "On building prediction systems for software engineers." Empirical Software Engineering **5**(3): 175-182.

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

Shepperd, M. and C. Schofield (1997). "Estimating software project effort using analogies." IEEE Transactions on Software Engineering **23**(11): 736-743.

Shoval, P. and O. Feldman (1997). "A combination of the Mk-II function points software estimation method with the ADISSA methodology for systems analysis and design." Information and Software Technology **39**(13): 855-865.

Shukla, K. K. (2000). "Neuro-genetic prediction of software development effort." Information and Software Technology **42**(10): 701-713.

Smith, R. K., J. E. Hale, et al. (2001). "An empirical study using task assignment patterns to improve the accuracy of software effort estimation." IEEE Transactions on Software Engineering **27**(3): 264-271.

Srinivasan, K. and D. Fisher (1995). "Machine learning approaches to estimating software development effort." IEEE Transactions on Software Engineering **21**(2): 126-137.

Stamelos, I. and L. Angelis (2001). "Managing uncertainty in project portfolio cost estimation." Information and Software Technology **43**(13): 759-768.

Stamelos, I., L. Angelis, et al. (2003). "On the use of Bayesian belief networks for the prediction of software productivity." Information and Software Technology **45**(1): 51-60.

Stensrud, -. E., -. T. Foss, et al. (2003). "A further empirical investigation of the relationship between MRE and project size." Empirical Software Engineering **8**(2): 139-161.

Stensrud, E. (2001). "Alternative approaches to effort prediction of ERP projects." Information and Software Technology **43**(7): 413-423.

Stewart, B. (2002). "Predicting project delivery rates using the Naive-Bayes classifier." Journal of Software Maintenance and Evolution: Research and Practice **14**(3): 161-179.

Strike, K., K. El-Emam, et al. (2001). "Software cost estimation with incomplete data." IEEE Transactions on Software Engineering **27**(10): 890-908.

Stutzke, R. D. (1996). "Software estimating technology: A survey." Journal of Defense Software Engineering (Crosstalk) **9**(5): 17-22.

Subramanian, A. and M. C. Lacity (1997). "Determinants of variability in function point estimates." Journal of End User Computing **9**(4): 19-28.

Subramanian, G. H. and S. Breslawski (1993). "Dimensionality reduction in software development effort estimation." Journal of Systems and Software **21**(2): 187-196.

Subramanian, G. H. and S. Breslawski (1995). "An empirical analysis of software effort estimate alterations." Journal of Systems and Software **31**(2): 135-141.

Symons, C. R. (1988). "Function Point Analysis: Difficulties and Improvements." IEEE Transactions on Software Engineering **14**(1): 2-11.

Taff, L. M., J. W. Borchering, et al. (1991). "Estimeetings: development estimates and a front-end process for a large project." IEEE Transactions on Software Engineering **17**(8): 839-849.

Tate, G. and J. M. Verner (1990). "Software sizing and costing models: a survey of empirical validation and comparison studies." Journal of Information Technology **5**(1): 12-26.

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

Uemure, T., S. Kusumoto, et al. (2001). "Function-point analysis using design specifications based on the Unified Modelling Language." Journal of Software Maintenance and Evolution: Research and Practice **13**(4): 223-243.

van der Poel, K. G. and S. R. Schach (1983). "A software metric for cost estimation and efficiency measurement in data processing system development." Journal of Systems and Software **3**(3): 187-191.

van Genuchten, M. (1991). "Why is software late? An empirical study of reasons for delay in software development." IEEE Transactions on Software Engineering **17**(6): 582-590.

Verner, J. and G. Tate (1987). "A model for software sizing." Journal of Systems and Software **7**(2): 173-177.

Verner, J. and G. Tate (1988). "Estimating size and effort in fourth-generation development." IEEE Software **5**(4): 15-23.

Verner, J. and G. Tate (1992). "A software size model." IEEE Transactions on Software Engineering **18**(4): 265-278.

Verner, J. M., S. P. Overmyer, et al. (1999). "In the 25 years since The Mythical Man-Month what have we learned about project management?" Information and Software Technology **41**(14): 1021-1026.

Vicinanza, S. S., T. Mukhopadhyay, et al. (1991). "Software effort estimation: An exploratory study of expert performance." Information Systems Research **2**(4): 243-262.

Vijayakumar, S. (1997). "Use of historical data in software cost estimation." Computing and Control Engineering Journal **8**(3): 113-119.

Walkerden, F. and D. R. Jeffery (1997). "Software cost estimation: A review of models, process, and practice." Advances in Computers **44**: 59-125.

Walkerden, F. and R. Jeffery (1999). "An empirical study of analogy-based software effort estimation." Empirical Software Engineering **4**(2): 135-158.

Walston, C. and C. Felix (1977). "A method of programming measurement and estimation." IBM Systems Journal **16**(1): 54-73.

Warburton, R. D. H. (1983). "Managing and predicting the costs of real-time software." IEEE Transactions on Software Engineering **9**(5): 562-569.

Watson, K. I. (1992). "COCOMO as a schedule prognosis and validation tool: A case study." Software Quality Journal **1**(4): 193-208.

Weinberg, G. M. and E. L. Schulman (1974). "Goals and performance in computer programming." Human Factors **16**(1): 70 - 77.

Whang, S. (1992). "Contracting for software development." Management Science **38**(3): 307 - 324.

Wittig, G. and G. Finnie (1997). "Estimating software development effort with connectionist models." Information and Software Technology **39**(7): 469-476.

Wittig, G. E. and G. R. Finnie (1994). "Using artificial neural networks and function points to estimate 4GL software development effort." Australian Journal of Information Systems **1**(2): 87-94.

Wolverton, R. W. (1974). "The cost of developing large-scale software." IEEE Transactions on Software Engineering **C-23**(6): 615-636.

Woodfield, S. N., V. Y. Shen, et al. (1981). "A study of several metrics for programming effort." Journal of Systems and Software **2**(2): 97-103.

Wrigley, C. D. and A. S. Dexter (1991). "A model for measuring information system size." MIS Quarterly **15**(2): 245-257.

Yau, C. and L.-Y. Gan (1995). "Comparing the top-down and bottom-up approaches of function point analysis: A case study." Software Quality Journal **4**(3): 175-187.

Yau, C. and T. Ho Leung (1998). "Modelling the probabilistic behaviour of function point analysis." Information and Software Technology **40**(2): 59-68.

Zhang, X. and H. Pham (1998a). "A software cost model with error removal times and risk costs." International Journal of Systems Science **29**(4): 435-442.

Zhang, X. and H. Pham (1998b). "A software cost model with warranty cost, error removal times and risk costs." IIE Transactions **30**(12): 1135-1142.

APPENDIX 3: Classification Descriptions

Category: Research topic

1. Estimation methods: Studies of methods producing effort estimates. This includes formal estimation models, expert estimation processes, decomposition-based estimation processes, and processes for combining different estimation methods and sources. It does not include estimation methods with a focus on size.
2. Production functions: Studies of properties of the "software production function". This includes studies on linear versus non-linear relationship between effort and size, and on the relationship between effort and schedule compression. Studies on production functions differ from studies on estimation methods in that they focus on selected properties of formal estimation models, not on complete estimation methods.
3. Calibration of models: Studies on calibration of estimation models, e.g., studies on local versus multi-organizational data or calibration of the COCOMO model to certain types of projects.
4. Size measures: Studies with a main focus on the validity and improvements of size measures that are important in estimation models, e.g., inter-rater validity of function points counting. Studies discussing size measures, but with a main focus on developing effort estimation methods, are not classified as Sm.
5. Organizational issues: Studies on estimation processes in a wide organizational context, e.g., surveys on estimation practice, reasons for cost overruns, impact of estimate on project work, and estimation in the general context of project management.
6. Effort uncertainty assessments: Studies on uncertainty of effort or size estimates, e.g., methods providing minimum-maximum intervals for effort.
7. Measures of estimation performance: Studies on evaluation and selection of estimation methods, e.g., studies on how to measure estimation accuracy or how to compare estimation methods.
8. Data set properties: Studies on how to analyze data sets for the purpose of estimation methods, e.g., data sets with missing data.
9. Other topics: Unclassified topics.

Category: Estimation approach

1. Regression: Studies on regression-based estimation methods, including most algorithmic models, e.g., studies on the COCOMO estimation model. Studies applying regression analysis for other purposes, e.g., to validate or test relationships, are *not* included.
2. Analogy: Studies on analogy- and case-based reasoning estimation methods, e.g., studies evaluating estimation by analogy on different data sets.
3. Expert judgment: Studies on expert judgment-based estimation methods. This includes structured processes for expert judgment, such as AHP and paired comparisons. Studies where expert judgment is input to a formal estimation method or where expert judgment based on output from other types of estimation approaches is *not* classified as Ej.
4. Work breakdown: Studies on work breakdown structure-based and other activity decomposition-based estimation methods.
5. Function Point: Studies on "function point"-based estimation methods, including studies on estimation methods based on feature points and use case points.
6. CART: Studies on estimation methods based on classification and regression trees.
7. Simulation: Studies based on simulation-based/derived models, e.g., system dynamics studies applying the Monte Carlo simulation.
8. Neural network: Studies on artificial neural network-based estimation methods.
9. Theory: Studies on theory-derived estimation models, e.g., the SLIM estimation model and estimation models based on "software science".
10. Bayesian: Bayesian or Markow-based estimation models.
11. Combination of estimates: Studies on the combination of estimates from difference sources, e.g., the combination of expert and formal model-based estimates.
12. Other: Estimation methods based on other techniques for estimation modelling, e.g., genetic programming, to derive estimation models.
13. Not relevant: Studies on estimation methods in general and studies where the approach to estimation is not relevant.

Category: Research approach

1. Theory: Non-empirical research approaches or theoretical evaluation of properties of estimation models. Most studies apply theories, but only studies that rely heavily on non-empirical research methods in their evaluation and development of estimation approach are included here.
2. Survey: Survey-based studies, e.g., questionnaire- and interview-based surveys of industry practice.
3. Experiment: Experiment-based studies.
4. Case study: Case-based studies, e.g. in-depth study of the estimation processes of one, or a very small number, of software projects.
5. Development of estimation method: Studies where new effort (or size) estimation models, processes or tools are developed. This includes new methods for combining estimates.
6. History-based evaluation: Studies evaluating estimation methods or other estimation-relevant relationships on previously completed software projects.
7. Own experience/lessons learned: Studies where the *only* reference is one's own experience, without any scientific documentation of the experience as a description of case studies, observations, experiments, etc..
8. Real-life evaluation: Studies evaluating estimation methods/expert estimation in real estimation situations.
9. Review: Studies that review other estimation papers / estimation methods. Studies where a review of other papers is not the main purpose of the paper, but is included mainly as a discussion of related work or as input to theory, or is included in discussions of results, are not classified as reviews.
10. Simulation: Simulation-based studies. This category relates to the research method, e.g., the evaluation method, *not* to the estimation method itself.
11. Other: Studies with other research approaches.

Category: Study contex*t*

1. Students: Studies where the subjects are students and/or student projects.

2. Professionals: Studies where the subjects are software professionals and/or industrial software projects.
3. Not relevant: Studies where the study context is not relevant.

Category: Data set

1. Own data, not included: Own data set applied to develop and/or evaluate estimation methods. The data set is not included in the paper.
2. Own data, included: Own data set, but the data set is not included in the paper.
3. Xx: Data sets borrowed from other studies, e.g., the data set Bc refers to the Barry Boehm, COCOMO 81 data set.
4. No data set: No data set applied