Communications of the Association for Information Systems

Volume 34 Article 4

1-2014

A Systematic Mapping of Factors Affecting Accuracy of Software Development Effort Estimation

Dirk Basten
University of Cologne, basten@wiso.uni-koeln.de

Ali Sunyaev
Department of Information Systems, University of Cologne

Recommended Citation

Basten, Dirk and Sunyaev, Ali (2014) "A Systematic Mapping of Factors Affecting Accuracy of Software Development Effort Estimation," *Communications of the Association for Information Systems*: Vol. 34 , Article 4.
DOI: 10.17705/1CAIS.03404
Available at: https://aisel.aisnet.org/cais/vol34/iss1/4

Communications of the Association for Information Systems

CAIS

A Systematic Mapping of Factors Affecting Accuracy of Software Development Effort Estimation

Dirk Basten

Department of Information Systems and Systems Development, University of Cologne

basten@wiso.uni-koeln.de

Ali Sunyaev

Department of Information Systems, University of Cologne

Abstract:

Software projects often do not meet their scheduling and budgeting targets. Inaccurate estimates are often responsible for this mismatch. This study investigates extant research on factors that affect accuracy of software development effort estimation. The purpose is to synthesize existing knowledge, propose directions for future research, and improve estimation accuracy in practice. A systematic mapping study (a comprehensive review of existing research) is conducted to identify such factors and their impact on estimation accuracy. Thirty-two factors assigned to four categories (estimation process, estimator's characteristics, project to be estimated, and external context) are identified in a variety of research studies. Although the significant impact of several factors has been shown, results are limited by the lack of insight into the extent of these impacts. Our results imply a shift in research focus and design to gather more in-depth insights. Moreover, our results emphasize the need to argue for specific design decisions to enable a better understanding of possible influences of the study design on the credibility of the results. For software developers, our results provide a useful map to check the assumptions that undergird their estimates, to build comprehensive experience databases, and to adequately staff design projects.

Keywords: information system projects, effort estimation, systematic mapping, software development

Volume 34, Article 4, pp. 51-86, January 2014

Communications of the Association for Information Systems

I. INTRODUCTION

Accurate estimation of software development effort is of high importance to both software professionals and researchers. In fact, effort estimation is one of the fundamental activities of conceptualizing projects [Jurison, 1999], and its importance is evident in many software development projects. For instance, effort estimates are used for project planning, project staffing, project progress monitoring, project success assessing, and the evaluation of developers [Moløkken-Østvold et al., 2004; Basten and Mellis, 2011]. Whereas overestimated effort can lead to omitting beneficial projects and thereby to lost business opportunities, underestimated effort may lead to budget and schedule overruns and, thus, result in significant losses [Grimstad, Jørgensen, and Moløkken-Østvold, 2006; Kappelman, McKeeman, and Zhang, 2006; Mukhopadhyay, Vicinanza, and Prietula, 1992]. The high importance of effort estimation in practice is also demonstrated in a survey among more than 1000 IT professionals [CompTIA, 2007]. The participants perceive two of the three major reasons for project failure to be associated with poor effort estimation. As Jørgensen and Shepperd [2007] showed in their review, effort estimation of software projects also received great attention in the scientific literature over the last decades.

In order to increase accuracy of effort estimation, this study investigated various factors affecting this accuracy in literature [Jørgensen and Grimstad, 2008; Lederer and Prasad, 1995]. Previous research analyzed differences in accuracy of effort estimation depending on the estimation technique applied [Li and Ruhe, 2008; Li, Xie, and Goh, 2009; Xia, Ho, and Capretz, 2008; Moløkken-Østvold et al., 2004]. When considering different estimation techniques, the most general question is whether to rely on formal models or expert judgment [Jørgensen, Boehm, and Rifkin, 2009]. Formal models have been developed since the 1960s and 1970s [Jørgensen et al., 2009], but empirical findings indicate that they are seldom applied [Jørgensen, 2004a]. The focus on expert estimation can be explained by the finding that this technique can be more easily applied and that, in general, no significant difference exists in the level of accuracy when comparing formal models and expert judgment. The comprehensive review by Jørgensen showed that “there is no substantial evidence supporting the superiority of model estimates over expert estimates” [2004a, p. 55].

Due to the lack of substantial evidence for the superiority of model or expert estimates, we neglect the choice of the estimation technique and focus on the variety of further factors that directly influence accuracy of software development effort estimation. Examples include neutrality and relevance of information [Jørgensen and Grimstad, 2008], request format [Jørgensen and Halkjelsvik, 2010], historical data [McDonald, 2005], and estimator experience [Morgenshtern et al., 2007]. Although some of these factors are related to characteristics of an estimator and thus primarily apply to expert estimation, they are of general interest as they affect accuracy of effort estimation irrespective of which technique is used. This is due to the fact that formal models are not entirely objective and essential input to such models is based on expert judgment [Jørgensen et al., 2009; Grimstad and Jørgensen, 2007].

Researchers dealt with a multitude of such factors, but did so in varying ways. While studies in the 1990s predominantly aimed to identify factor lists as comprehensively as possible [van Genuchten, 1991; Lederer and Prasad, 1995; Gray, MacDonell, and Shepperd, 1999], more recent studies seem to evaluate the impact of a single or a few factors on accuracy of software development effort estimation [Jørgensen and Grimstad, 2008; Grimstad and Jørgensen, 2007; Connolly and Dean, 1997]. Consequently, a wide range of results exists concerning various factors, and related strength of evidence differs for the various researched factors depending on the quality of the studies. Per our literature review (cf. Section III) and, therefore, to the best of our knowledge, an overview of factors affecting accuracy of software development effort estimation and the corresponding evidence does not exist. However, it will advance both the theoretical and practical understanding in the field of effort estimation. Along with a comprehensive assessment of influences on estimation accuracy, the overview is beneficial, as it analyzes previously applied research approaches to guide future studies on effort estimation accuracy toward a structured and rigorous research stream. Consequently, we pose our main research question (RQ) as follows:

What evidence exists regarding the factors that affect the accuracy of software development effort estimation?

To answer this research question, we conducted a systematic mapping [Kitchenham, 2007] of factors that affect accuracy of software development effort estimation. A systematic mapping study is defined as “a broad review of primary studies in a specific topic area that aims to identify what evidence is available on the topic” [Kitchenham, 2007, p. vii] and “allows the evidence in a domain to be plotted at a high level of granularity” [Kitchenham, 2007, p.

5]. In systematic mappings, therefore, analyses of the identified studies are more likely to be on an aggregated level—that is, to provide totals and summaries [Kitchenham, 2007]. Moreover, it is typical to pose broader and multiple research questions. Accordingly, in this study we divide our main research questions into five broader questions that we will address.

As a first step, we present an overview of factors researched to affect accuracy of software development effort estimation:

RQ1: Which factors were identified to affect accuracy of software development effort estimation?

Studies identifying such factors may be based on different research methods (e.g., experiments, expert interviews) and may be conducted in software development as well as other research areas (e.g., other forecasting disciplines). We aim to gather together these various studies to provide an overview of all potentially relevant factors and their definitions. The following research questions refer to the factors identified by answering RQ1.

RQ2: Which research methods were applied to identify these factors?

Although this analysis of applied research methods provides insights into the general research approach, we are moreover interested in the quality of conducting and reporting the research.

RQ3: Which level of quality do the identified studies provide?

Furthermore and partially based on this quality assessment, we aim to gather knowledge about the provided strength of evidence (which can be used for statements about the generalizability of the results).

RQ4: Which strength of evidence do the identified studies provide?

For a better understanding of the different factors, we assess to which degree the factors impact estimation accuracy.

RQ5: To which degree do the factors affect accuracy of software development effort estimation?

In order to satisfactorily answer these questions and to ensure high quality in our systematic mapping, we conduct a systematic literature review of fifteen leading software and information system journals, focusing on empirical studies investigating software development effort estimation. We solely include factors associated with the estimation process itself. Factors that lead to deviations between estimated and actual effort after an estimation is completed (e.g., management removal of padding, change requests) are not included in our review, as these cannot be regarded in an estimation process and, thus, should be handled separately.

With our systematic mapping, we contribute to both research and practice. Predominantly, researchers obtain an overview of what factors were studied and how. This overview helps to direct future studies on effort estimation and, therefore, to establish strength of evidence in this field of research. We assess the way in which research on this topic is conducted so far. We then provide guidelines for closing research gaps and improving the quality of future studies. Practitioners are likely to find this overview of factors helpful for controlling risks in the project estimating and planning phases. They might also alter their method of data collection. During all project phases, it is not only essential to continuously collect data, but also to consider all relevant aspects—that is, all factors that are likely to affect the accuracy of software development effort estimation. Based on such data, it is possible to analyze the impact of factors and thereby improve estimation accuracy.

The remainder of our article is structured as follows. We present the theoretical background relevant to the study (Section II). We then describe the systematic mapping and the analysis of the articles (Section III). After presenting the results, we discuss the findings (Section IV), and finally provide implications for future research and project management (Section V).

II. THEORETICAL BACKGROUND

In order to ensure construct validity as recommended by Gibbert et al. [2008], we first elaborate on the key concepts and terms used in the study. Along these lines, we motivate the synonymous usage of the terms *software* and *information system* in our context; address the lack of a clear definition of *effort estimate* in research; and differentiate the concepts of *size*, *effort*, *cost*, and *schedule*. We then address previous research on factors influencing estimation accuracy.

Key Concepts and Terms

Whereas software "is defined as computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system" [Linberg, 1999, p. 179], an information system "can be defined technically as a set of interrelated components that collect (or retrieve), process, store, and distribute information to support decision making and control in an organization" [Laudon and Laudon, 2009, p. 46]. Obviously, a clear distinction is made between software and information systems, as software is one of the interrelated components in information systems. However, in the context of project management, no differentiation seems to be made between these two terms from the research perspective—many authors use management of software projects synonymously with management of information system projects (e.g., Jurison, 1999; Lederer and Prasad, 1993). Moreover, textbooks on *information system* project management usually refer to a *software* development effort when describing estimation and planning processes (e.g., Avison and Torkzadeh, 2009; Cadle and Yeates, 2008; McManus and Wood-Harper, 2003). This equivalence of the terms *software* and *information systems* in the project management context may result from information system projects where labor is seen as the major cost driver [Allen, 1982; Lederer et al., 1990; Shepperd and Schofield, 1997]. Labor, in this context, should include all activities associated with software requirement analysis, design, coding, testing, and rollout, as well as documentation, configuration management, business or quality assurance, and project management [McManus and Wood-Harper, 2003, p. 206]. Other costs (e.g., for hardware) are usually neglected. Furthermore, characteristics associated with information system development processes also apply to software development processes [Avison and Torkzadeh, 2009]. In our context of effort estimation in projects, therefore, we do not distinguish between software and information system projects.

In general, an effort is defined as "a vigorous or determined attempt" and an estimate is "an approximate calculation or judgment of the value, number, quantity, or extent of something" (Oxford Dictionary). However, a lack of a common understanding of the term *effort estimate* exists in the context of software projects [Grimstad et al., 2006] despite extensive research over the last decades (cf. the review in Jørgensen and Shepperd, 2007). According to Jørgensen [2007], typical interpretations of effort estimates include planned effort, budgeted effort, most likely use of effort (median value), and the effort with a 50 percent probability of not being exceeded (median value). Jørgensen [2007] addresses this shortcoming and the problems of measuring estimation accuracy without a clear definition. However, research shows that even if a clear definition is shared between researchers and study participants, no difference in accuracy rates exists compared to previous studies [Basten and Mellis, 2011]. Although we concur with the usefulness of establishing an agreed-upon definition of the concept effort estimate in general, we do not attempt to provide one in this article. First, applying a single definition would lead to an exclusion of studies for which we cannot be sure of the definition applied. Second and more importantly, it is unlikely that the presence of a clear and agreed-upon definition creates an impact on the relevance of a factor (e.g., lying, irrelevant and misleading information).

Regarding effort estimates in software projects, *size*, *cost*, and *schedule* are closely related concepts that need to be considered. Size denotes the relative extent of something (e.g., a software), cost refers to the payment of a specified sum of money (e.g., to develop the software), and a schedule is the plan for carrying out a process or procedure (e.g., the software development) (Oxford Dictionary). Following guidelines for project estimating and planning [McManus and Wood-Harper, 2003, pp. 206–215], the key determinant for effort, cost, and schedule is software size, usually measured in source lines of codes or function points. The majority of research articles (cf. the review in Jørgensen and Shepperd, 2007) use the terms *effort* and *cost* in an interchangeable way; however, we will use the expression *effort* exclusively to denote software development expenditure (e.g., measured in person-days) since *cost* (e.g., measured in labor rates) is calculated based on effort estimates. Estimated effort needs to be converted to cost depending on, for instance, the number of team members involved and interdependencies between the various project tasks. The same applies to schedule estimation, which is aimed to determine "the length of time needed to complete the project and determine when major milestones and reviews will occur. Before this process can take place, the functional work breakdown structures (WBS), size and cost estimates must have been completed" [McManus and Wood-Harper, 2003, pp. 214–215]. Whereas the indices of size and effort are estimated, the indices of cost and schedule are calculated or planned based on estimates. Despite the relatedness, therefore, we emphasize that none of the above-mentioned concepts should be equated. Consequently, it is unreasonable to argue that the factors impacting estimation or planning accuracy for one of the four different indices (size, effort, cost, schedule) will also apply to the other three. In this study, therefore, we consider only those factors that affect accuracy of software development effort estimates.

Previous Research on Effort Estimation Accuracy Factors

An analysis of previous research shows that different types of classification schemes exist; they emphasize important research areas (e.g., van Genuchten, 1991; Lederer and Prasad, 1995). We believe that three main reasons exist that explain why these classification schemes differ. First, the studies have different foci and do not address the same set of factors. Consequently, categories classifying the factors are likely to differ as well. Second, some studies do not refer to accuracy factors only; they also involve factors that lead to differences between estimated and actual effort that are beyond actual estimation process (e.g., management removal of padding; cf. Lederer et al., 1990). Finally, it is probably not possible to build a scheme that is suitable in every context. Hence, the derived classification scheme depends on the purpose for which a classification is performed. We believe that a suitable approach to derive categories is to apply the principle of causation according to which "the relation between cause and effect is considered to be a necessary outcome" [Regopoulos, 1966, p. C136]. Following this principle, we are able to directly specify how to cope with the causes of inaccuracies. Following this approach, several main areas were emphasized by previous researchers, as explained in the following.

When considering development of effort estimates for software projects, process factors build the first category to be regarded. Factors related to this category are inherent in the process and can be controlled by project management. Examples include accountability [Lederer and Prasad, 2000] and request format [Jørgensen and Halkjelsvik, 2010]. Moreover, empirical studies indicate that accurate estimation depends on project and estimator characteristics [Henrion, Fischer, and Mullin, 1993; Hora, Dodd, and Hora, 1993], which might also be controlled by project management. Whereas project characteristics primarily cover aspects such as neutrality and relevance of information [Jørgensen and Grimstad, 2008] and task complexity [Gray et al., 1999], estimator characteristics refer to estimation experience [Morgenshtern, Raz, and Dov, 2007] and reference to the project [Jørgensen and Moløkken-Østvold, 2004]. The latter category is especially important as identifying "such characteristics … may be a first step to support software organizations' selection of software engineers who are less likely to provide strongly optimistic predictions" [Jørgensen, Faugli, and Gruschke, 2007, p. 1472]. Finally, several external influences are operative beyond estimation processes and, therefore, are not under the control of project management. Examples include client maturity and contract type [Moløkken-Østvold and Jørgensen, 2005a].

Irrespective of the different categories, research is applied in different ways. However, two major study types prevail. On the one hand, studies identify factors associated with estimation accuracy [Moløkken-Østvold and Jørgensen, 2005a; Jørgensen and Moløkken-Østvold, 2004; van Genuchten, 1991]. On the other hand, researchers focus on selected factors and evaluate their impact on estimation accuracy [Jørgensen and Grimstad, 2008; Connolly and Dean, 1997; Grimstad and Jørgensen, 2007]. Furthermore, the applied research methods span a broad range, including, among others, quantitative surveys [Lederer and Prasad, 1995], expert interviews [Moløkken-Østvold and Jørgensen, 2005a], case studies [Lederer et al., 1990], and controlled experiments [Grimstad and Jørgensen, 2007]. The choice of one of these research methods is limited if researchers aim to ensure applicability of their results to real situations. Whereas controlled experiments provide a high degree of evidence, realism of the results is at least questionable, as experiment participants might behave differently in real life.

While a great body of research addresses factors that affect accuracy of software development effort estimation, an overview of these factors and the corresponding studies does not exist. We believe that such an overview will advance the theoretical and practical understanding in the field of effort estimation. Along with a comprehensive assessment of influences on estimation accuracy, such an overview would be beneficial, given that it would analyze previously applied research approaches to guide future studies on effort estimation accuracy toward an even more structured and rigorous research stream. An overview of factors affecting accuracy of software development effort estimation is likewise advantageous for researchers and practitioners, as it helps both parties to be aware of what factors were studied and in what manner. Thus, researchers benefit from a map to guide future studies on effort estimation. The mapping helps to focus on factors identified as being potentially relevant but not evaluated yet. Furthermore, the mapping will show contradicting results concerning single factors and provide a better understanding of the extent to which factors affect accuracy of software development effort estimation—in other words, the combined analysis of various studies concerning single factors. As we also consider research methods applied and quality and strength of evidence provided by the studies, we close the gap of a missing systematic mapping of the according subjects. Directing future studies based on these assessments will contribute to raising the strength of evidence in this field of research. We also will move the literature forward by providing guidelines on how to design and report effort estimation studies. Practitioners might use the overview of factors to control risks in project estimating and planning phases. Moreover, it is essential to continuously collect data about various surroundings of an estimation process to build an adequate experience database. Finally, as we also categorize the various factors, our mapping can be used to improve estimation accuracy in research and practice step by step.

III. SYSTEMATIC MAPPING

To answer our research questions, we conducted a systematic mapping study [Kitchenham, 2007] following guidelines on how to conduct systematic mapping studies in software engineering [Petersen, Feldt, Mujtaba, and Mattsson, 2008]. The three necessary steps are: (1) the identification of relevant articles, (2) the development of a classification, and (3) the mapping of the articles.

Data Sources and Search Process

In general, when systematically searching the literature, we see two distinct approaches. First, an automated search uses digital libraries to search for germane keywords. The second option is a manual issue-by-issue search of the relevant journals over a given period. Due to the multitude of key terms used in the context of software effort estimation studies, it would not be promising to rely solely on an automated search [Jørgensen and Shepperd, 2007]. As it is not workable to search all potentially relevant publications manually, we adapted the manual issue-by-issue search process as recommended by Webster and Watson [2002]. Starting from a base set of articles identified in leading software journals over more than a decade (1999 to the end of 2010), we worked backward by reviewing sources/works cited in these base articles. Then we worked forward by using digital libraries to identify works that cite these base articles.

We based our primary issue-by-issue search on the following journals:

- *Communications of the Association for Information Systems*
- *Empirical Software Engineering*
- *European Journal of Information Systems*
- *IEEE Transactions on Software Engineering*
- *IEEE Software*
- *Information and Management*
- *Information and Software Technology*
- *Information Systems Journal*
- *Information Systems Research*
- *International Journal of Project Management*
- *Journal of Information Technology*
- *Journal of Management Information Systems*
- *Journal of Systems and Software*
- *Management Information Systems Quarterly*
- *Management Science*

We chose those journals based on two criteria. First, we aimed to include the leading journals in the IS and software areas. Thus, we used the journal ranking of the Association for Information Systems (AIS) (http://ais.affiniscape.com/displaycommon.cfm?an=1&subarticlenbr=432) as a starting point. Second, we prioritized journals with references to software effort estimation. The review by Jørgensen and Shepperd [2007] was particularly helpful for this part of our identification process, because it provided a quantity of journals with a focus on software effort estimation. All journals selected for our review are highly ranked in the AIS ranking and/or provide a multitude of relevant articles about this specific topic.

We used the following steps for our search (cf. also Figure 1). To minimize the risk of missing potentially relevant articles, the search was carried out by two researchers independently. The two lists of relevant articles were merged. If only one of the researchers voted to include an article, both researchers read the article in its entirety to decide on its inclusion or exclusion. Our primary review contained all articles (13,882) of the fifteen above-mentioned journals published since 1999. The initial search resulted in a consolidated list of eighteen articles from six journals *(Empirical Software Engineering, IEEE Transactions on Software Engineering, IEEE Software, Information and Software Technology, International Journal of Project Management,* and *Journal of Systems and Software*). Two researchers primarily read the articles' titles and abstracts. In cases of disagreement, the researchers read the articles in-depth and then jointly decided on inclusion or exclusion.

The fact that nine journals did not contribute to the results may be due to the diversity of aspects concerning software effort estimates. Factors influencing accuracy of such estimation are only a small part of research on this subject. For example, other studies deal with the comparison of estimation methods [Finnie, Wittig, and Desharnais, 1997; Kemerer, 1987; Moløkken-Østvold et al., 2004] or the improvement of estimation methods [Burgess and Lefley, 2001; Matson, Barrett, and Mellichamp, 1994; Xia et al., 2008].

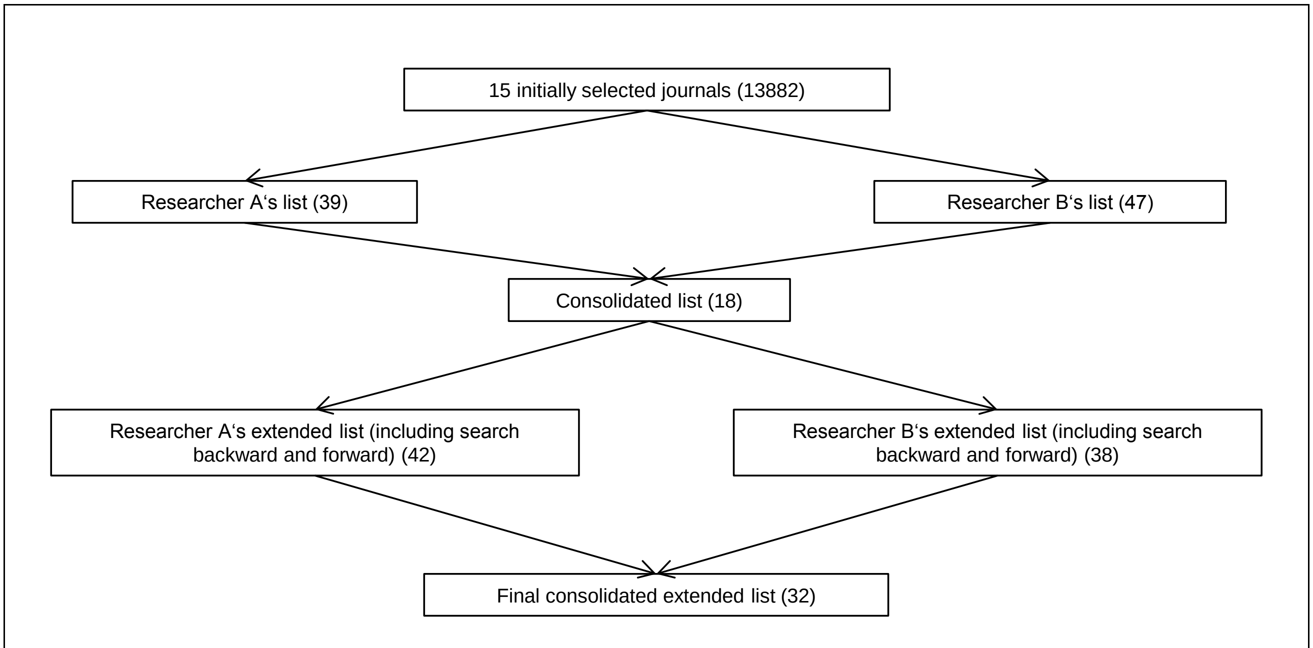

Figure 1. Search and Selection Process for Relevant Scientific Articles

Afterwards, we reviewed the references of those eighteen articles with regard to the above criteria. This backward search resulted in eleven more articles. Then, the researchers performed the forward search using Web of Science (http://www.isiknowledge.com) and Google Scholar (http://scholar.google.com). This procedure led to another three articles. In total, we identified thirty-two articles. Twenty-six of them come from nine journals; the remaining six articles were published at various software conferences. Twice two articles from the same authors were similar, as they reported results from the same study. In those cases, the earlier version of the article was excluded, but only if the research approach and the identified factors were the same and the articles differed simply in the way the study was described.

Inclusion and Exclusion Criteria

We used the following inclusion and exclusion criteria, which we defined and refined through a pilot selection of studies (cf. Figure 1). We selected the studies by analyzing the titles, abstracts, or full text of the papers (in cases where title and abstract were not sufficient for a decision).

Inclusion criteria:

- Peer-reviewed empirical publications from electronic databases
- Publications identifying factors as potentially important (e.g., opinion surveys among software experts)
- Publications analyzing the impact of factors on accuracy (e.g., experiments) of effort estimation
- In case of duplicate publications, we included only the most complete and newest study.

Exclusion criteria (in general publications without relevance to our research questions):

- Publications dealing with estimation methods, their development, or their improvement
- Publications concerned with the uncertainty assessment of effort estimation
- Publications related to estimation of software size, the assessment of software complexity, or the identification of factors associated with software development effort
- Publications related to the development process itself that inhibit the adherence-to-schedule or similar estimation-related topics
- Non-English papers
- Technical reports, books, discussions, and opinion papers

Primary Studies

Thirty-two identified studies were of interest for our mapping study. The relevant articles were published between 1990 and 2010 (cf. Appendix A for the distribution of the publication fora). We provide an overview of the identified publications in Appendix B. In that overview, we use the Goal Question Metric (GQM) template to characterize the studies according to the criteria of object of study, purpose, quality focus, viewpoint, and context [Briand, Differding, and Rombach, 1996]. Except for the purpose, all information was directly extracted from the studies. The purpose may refer to characterization (aims at assessing the current state), evaluation (aims at comparing and assessing efficiency/effectiveness of processes), prediction (aims at identifying relationships between various factors and using these relationships to predict relevant external attributes of processes), or control and change (aims at identifying causal relationships that influence the state of processes).

Data Collection

To ensure the completeness of our analysis and for reasons of validity, the data was collected by two researchers independently, according to the following proposition. We analyzed the selected publications regarding the factors that were shown to likely impact accuracy of software development effort estimation. We included a factor if it was either explicitly put in the focus of the research (e.g., "To what extent do lessons-learned sessions […] lead to improved accuracy of effort estimation […]?"; cf. Jørgensen and Gruschke, 2009, p. 369) or listed as an influence that may affect accuracy of software development effort estimation (cf., for instance, Table 2 in Lederer and Prasad, 1995, p. 129). Factors were included in our results only if they directly affected accuracy of effort estimation. Factors that caused deviations between estimated and actual effort (e.g., management removal of padding) were not included. The two factor lists were merged and the related data assembled. In addition to the factors denotation, we collected data about the following:

- Factor's name and (if provided) its definition
- Factor bias—whether a factor's occurrence leads to (perceived) overestimation or underestimation
- The extent to which factor biases effort estimation (if applicable)
- The research method(s) used to identify and/or analyze the factor
- All relevant information to answer the questions included in the quality assessment

In some cases, different factors' definitions were coexistent or overlapping. As it hurts clarity to list the same factor twice with different names, we consolidated those factors. In the case of overlapping factors, we combined them to one super-ordinate factor. Such cases are explicitly mentioned along with an argumentation for our decisions in the related sections of Section IV.

Classification Scheme

Petersen et al. [2008] state that the three dimensions that can be used to classify publications are (1) research focus, (2) type of contribution, and (3) research type.

With regard to the research focus (1), we categorized the factors following the content analysis by Jankowicz [2004, pp. 148–149]. We applied the bootstrapping technique to aggregate the identified factors to categories:

- Starting with a single factor, factors are stepwise added to existing categories.
- If a factor does not fit into the existing categories, a new category is created.
- Such a new creation may lead to redefinitions of existing categories (combining or breaking up categories, with their items reallocated accordingly).
- The procedure ends when all factors are classified.

Overall, we categorized the factors in four categories already mentioned in Section II (estimation process, estimator's characteristics, project to be estimated, external context) that are described in the following:

- Factors related to the *estimation process* describe different ways to design the process of developing estimates and, therefore, are not related to characteristics of the estimator herself/himself or the project being estimated. These factors are about single projects (e.g., estimation strategy, request format) as well as cross-project issues (e.g., historical data, learning).
- *Estimator's characteristics* are factors that are associated with the estimator herself/himself, that is, his/her knowledge, skills, behavior, and reference to the project. These factors are inherent to the estimator and, therefore, cannot be controlled by the design of the estimation process.

- The factors belonging to the *project to be estimated* are features that are inherent in the development task (e.g., newness of technology, task complexity) or that surround the estimation process during the course of the project (e.g., lying, neutrality and relevance of information). Neither the design of the estimation process nor the estimator can influence these characteristics.
- The *external context* comprises factors that deal with the surroundings of the estimation that are related to the client (e.g., client priority) or general project characteristics that are determined by the client (e.g., bidding process) or in collaboration with the client (e.g., contract type). Thus, these factors are independent from the estimation process and the project to be estimated.

In our context, contributions (2) refer to the way the factors are researched. Therefore, we distinguish between two contribution types only: factors that are proposed to potentially affect accuracy of software development effort estimation, and factors that are evaluated with regard to their effects on accuracy of software development effort estimation.

Considering the research type (3), we analyzed the research methods that were used in the primary studies to identify or evaluate the according accuracy factors. As this dimension is based on our findings, we provide more detailed information in the relevant subsection of our results (cf. Section IV).

Quality Assessment

We assessed the quality of each selected study (cf. RQ3). We answered questions concerning different aspects of the study design and reporting. Appendix C shows the according questions that were used to conduct the quality assessment checks of the selected studies with regard to their methods used and the quality of reporting. We used three possible responses and their values on a three-point scale (Yes = 1; To some extent = 0.5; No = 0). The overall score—which provides the quality assessment—is the sum of the scores for all questions answered for each study. We also accounted for the cases in which studies provided limited information about the aspects under consideration.

We used the instrument for the quality assessment that was used for systematic reviews before. Ali, Babar, Chen, and Stol [2010] adapted the instrument, which is applicable to quality assessment of empirical studies in software engineering in general [Ali et al., 2010; Dybå, Dingsøyr, and Hanssen, 2007] from Dybå and Dingsøyr [2008a].

IV. RESULTS

In this section, we present our results and answer the research questions raised (cf. Section I). While the first subsection (Research Focus) provides an overview and descriptions of the identified factors (cf. RQ1), the following subsection (Research Type) is concerned with RQ2, RQ3, RQ4, and RQ5, which are all related to research methods applied and related quality and strength of evidence of the according primary studies.

Research Focus

Referring to our first research question (RQ1), we identified thirty-two different factors that impact accuracy of software effort estimation. Not all factors apply to every context (e.g., customer expectations may not exist in every case). In the following subsections (one for each category; cf. Section III), we provide the factors' descriptions and provide a short overview of our empirical findings. If no additional information is given, both types of bias (underestimation and overestimation) are possible. Table 2 through Table 5 provide an overview of the categories and their according factors. These concept matrices [Webster and Watson, 2002] indicate for each category which publications contribute to the factors' research results. "O" indicates that a factor was proposed to be important and "X" indicates that a factor was evaluated in the corresponding study. Accordingly, the frequency of a factor being referred to is not a measure of its importance, but should rather be seen as a useful guideline to identify all the identified studies that address a specific factor.

Estimation Process

Factors related to the estimation process describe different ways to design the process of developing estimates. Therefore, these factors are not related to characteristics of the estimator himself/herself or the project being estimated. Table 1 provides an overview of the factors presented in this section.

Table 1: Overview of Estimation Process Factors and the Identified Primary Studies

	Accountability	Assessment of Developer Skills/Assessment of Task Complexity	Careful Examination of the Estimate	Estimate Alteration	Estimation Development	Estimation Management	Estimation Strategy	Historical Data	Initial Risk Analysis	Learning	Request Format
Connolly and Dean, 1997							X				
Furulund and Moløkken-Østvold, 2007		O						X			
Jørgensen, 2004b		O									
Jørgensen, 2004c							X	X			
Jørgensen, 2010									X		
Jørgensen et al., 2007								X			
Jørgensen and Gruschke, 2009										X	
Jørgensen and Halkjelsvik, 2010											X
Jørgensen and Moløkken-Østvold, 2004		O						O		O	
Lederer et al., 1990		O			O			O			
Lederer and Prasad, 1995	X	X	X		X			X		X	
Lederer and Prasad, 2000	X										
Magazinović and Pernstål, 2008						X		X			
McDonald, 2005								X			
Moløkken-Østvold and Jørgensen, 2005a					O						
Moløkken-Østvold and Jørgensen, 2005b										O	
Morgenshtern et al., 2007						X					
Prechelt and Unger, 2000										X	
Rombach et al., 2008										X	
Subramanian and Breslawski, 1995				X							
van Genuchten, 1991		O									
Wesslén, 2000										X	

"O"' indicates that a factor was proposed to be important in the corresponding study.
"X" indicates that a factor was evaluated in the corresponding study.

Accountability. Accountability is defined as holding software experts responsible for their work. Accountability refers to the inclusion of estimation accuracy in the job reviews of estimators, developers, and managers. If estimation accuracy was more frequently part of an evaluation process, estimators possibly would experience more pressure to be accurate (reduction of overestimation and underestimation). Accountability practice is one of the direct indicators in a model of the estimation error [Lederer and Prasad, 2000]. As part of performance reviews, meeting estimates is also apparent in an earlier article of the authors [Lederer and Prasad, 1995]. According to the participants, this factor is only one of the less important causes of inaccurate estimation.

On the one hand, the results indicate that estimators' accountability might help improve the accuracy of estimates. On the other hand, further aspects must be considered. Jørgensen and Sjøberg [2001] showed that effort estimates have an impact on project work. Projects with a high priority on costs tend to adjust the work to fit the estimate. Such time pressure may lower the product's quality [Austin, 2001]. Professionals estimating their own work might also extend their estimate to fulfill the expectations in a definite manner later on. Accountability may also lead to adjustments so that the perceived accuracy is higher. One review [Lerner and Tetlock, 1999] shows that accountability in general has been widely researched. One of their findings referred to the difficulty in coping with the wide range of effects caused by accountability.

Assessment of Developer Skills/Assessment of Task Complexity. We decided to combine these separate factors because the inaccurate assessments of both developer skills and task complexity are interrelated. Inaccurate estimates could result from either highly skilled developers overestimating simple tasks (overestimated effort) or inexperienced team members who are overstrained because of high complexity (underestimated effort). Although this factor is mentioned in a variety of articles, its importance is questionable, as the results represent only the participants' perceptions. Project managers mention underestimated complexity as most important cause for inaccurate estimates [Furulund and Moløkken-Østvold, 2007]. Compared to task complexity, the assessment of developer skills is more often mentioned as a cause for inaccurate estimates. A high number of inexperienced team members [van Genuchten, 1991] is mentioned, as well as the fact that a "developer worked much faster than expected" [Jørgensen, 2004b, p. 307]. Software projects experience reports showed that tasks were simpler or developers more skilled than expected [Jørgensen and Moløkken-Østvold, 2004]. Two articles reflect the problem of accurately assessing the team members' characteristics [Lederer et al., 1990; Lederer and Prasad, 1995].

Careful Examination of the Estimate. This factor denotes the process of carefully examining an estimate once developed. Only one article [Lederer and Prasad, 1995] describes this influence. Whereas the regression analysis of survey data shows a strong and highly significant correlation with estimation accuracy, the participants believe this factor to be of minor importance. It seems natural that the IS managers (94 percent of the participants) did not rate themselves as one of the reasons for inaccurate estimates, as the IS department's management is usually responsible for a lack of careful examination. The credibility of this finding might be limited as, for example, a study with estimators in different positions might lead to different results.

Estimate Alteration. Alteration of estimates is defined, in this case, as delaying as well as changing estimates. Adaptions of estimates can lead to improvements in their accuracies as the estimates then can be developed based on more information. The impact of estimate alterations on estimation accuracy is researched in one of the investigated studies [Subramanian and Breslawski, 1995]. The results illustrate the significant impact of alterations on estimation accuracy. The study's results show that changing initial estimates may lead to higher accuracy. Therefore, it makes no difference if alterations are made because of management-given time constraints or the experience of the project manager. In addition, estimation accuracy is improved by delaying the final estimate until a thorough analysis is conducted. Although no reason exists to doubt these results, it must be questioned why estimate alterations due to time constraints should lead to more accurate estimates. If estimates are lowered because of management constraints, it would be reasonable to believe that their accuracy may get even worse because the same project would need to be implemented in less time. In contrast, alterations made because of the manager's experience and delaying estimates actually seem to lead to improvements in accuracy.

Estimation Development. In terms of inaccurate estimates, software professionals identify the lack of an adequate methodology or guideline as a potential cause [Lederer et al., 1990; Lederer and Prasad, 1995]. Software professionals refer to objective guidelines on how to assess a developing task's complexity. Without those guidelines, estimators might allow too much tolerance to determine the effort needed. Another study mentions a good process of estimation development as a reason for accurate estimates [Moløkken-Østvold and Jørgensen, 2005a]. It is difficult to analyze this result any further, as no information is given about what makes an estimation process good compared to others. Therefore, it must be assumed that the managers in this case refer to an estimation development, which is subjectively perceived as frictionless.

Estimation Management. Estimation management comprises motivation and commitment to success [Morgenshtern et al., 2007]. The technical process of developing an estimate may influence not only accuracy, but also managerial decisions. Those include customer and IT unit controlling, the frequency of reporting, and performance and risk assessments. The results by Morgenshtern et al. [2007] indicate a relation between higher usage of such management processes and a smaller estimation error. It must be questioned, however, whether the results are representative for software projects in general. The two types of projects (high vs. low management control) might be seen as comparable, because they are all internally settled in the same governmental IT organization and were completed within the same time period. Although differences in the degree of management control were present, we expected the differences to be small enough to ensure comparability.

Estimation Strategy. In general, estimators can choose whether to develop estimates top-down or bottom-up. Top-down means that the overall project's effort is estimated and the effort is subsequently distributed to single activities. If single tasks are estimated and their individual efforts aggregated to the total effort, the estimation strategy is called bottom-up. In one study, different estimation teams estimated the effort of projects following both a top-down and a bottom-up approach [Jørgensen, 2004c]. The results did not suggest a general rule as to when to use each of the strategies or the extent of the influence on estimation accuracy. Jørgensen [2004c] recommended that top-down estimation should be used only if estimators have knowledge about similar, previously completed projects. Otherwise, the possible analogies (based on knowledge about similar, already completed projects, where the actual effort is known) that are crucial for accuracy in this approach will be hard to find. For this reason, the more effort-intensive bottom-up approach is advisable if estimators have no experience with or access to previous similar projects.

Two student experiments [Connolly and Dean, 1997] showed differences in estimation accuracy when using a top-down (holistic) or a bottom-up (decomposed) strategy. One of the experiments showed no difference, whereas the second one showed more accurate estimates for bottom-up strategies. Additionally, although the participants were advanced in their studies, they cannot be equated with experienced estimators.

Historical Data. Historical data refers to experiences with previous similar estimation tasks. In this context, the use of checklists and experience databases can help make the estimation process more transparent and can avoid overlooking tasks. In general, this can lead to more accurate estimates. This is reflected by the personal perspective of several estimation experts [Jørgensen and Moløkken-Østvold, 2004]. The importance of using experience data and checklists also was explicitly researched and confirmed [Furulund and Moløkken-Østvold, 2007]. Additionally, not using a common template for estimates affects the accuracies of effort estimates [Magazinović and Pernstål, 2008]. Jørgensen's [2004c] analysis of different outcomes depending on the estimation strategy (top-down vs. bottom-up) also showed interesting results concerning the use of such analogies. The two teams that actually used analogies seem to benefit from their knowledge, as their estimates were the most accurate. The importance of using historical data is also indicated in further studies [Lederer et al., 1990; Lederer and Prasad, 1995]. McDonald [2005] demonstrated that estimation teams with experience regarding similar estimation tasks provided more realistic estimates. Jørgensen et al. [2007] also found a relation between the ability to recall the previous use of effort and the optimism of estimators.

These eight studies indicate the significance of historical data in the accuracy of effort estimates. First, professionals point out that estimates will be less accurate when no historical data is available. Second, the studies within professional contexts indicate that the usage of relevant experience is strongly connected to accuracy of the estimates. Third, more realistic estimates are provided by students experienced with similar estimation tasks.

Initial Risk Analysis. Jørgensen [2010] analyzed whether the number of identified risks influences effort estimation. The results show that there are contexts in which estimates may be more optimistic if more effort is spent on the process of risk analysis. Therefore, the feeling of risk may affect accuracy of effort estimates. The time for identifying risks should be controlled carefully. The study proposes a second insight. Jørgensen [2010] suggests a four-step approach to combining risk identification and effort estimation processes. Beginning with the (1) estimation of the optimal effort, steps (2) and (3) are concerned with exposing risks and assessing their influence on the deviation from the optimal effort to adopt the effort estimate. Additionally, (4) effort for not-unforeseen risks should be added.

Learning. Learning effects are supposed to be highly relevant, as it is necessary to assess what went wrong on previous estimation tasks in order to improve estimators' skills. This is reflected in a study by Jørgensen and Moløkken-Østvold [2004] that shows systematic feedback is the highest demand of the participating software experts. Nevertheless, estimators may think that accuracy is not important if they do not receive appropriate feedback [Moløkken-Østvold and Jørgensen, 2005b]. Often, only the estimation accuracy of the overall project is evaluated. Such an approach does not enable any learning for individual estimators.

Jørgensen and Gruschke [2009] analyzed the variation in estimation accuracy through comparing estimates by groups of estimators with and without lessons-learned sessions. The variation in estimation accuracy was assessed through comparing estimates by groups of estimators with and without lessons-learned sessions. Those sessions included, among others, feedback on estimated and actual effort, estimation errors, and a thirty-minute learning session on perceived reasons for (in)accuracy and perceived realism in estimation accuracy. The fact that no significant difference between those groups turned up may be ascribed to the circumstance that estimators worked under. They learned individually, with no chance to ask for support from colleagues.

By using the disciplines of the Personal Software Process (PSP), accuracy improvements could be gained. PSP is a course and framework to support engineers in controlling and improving their performance (for more information, see Humphrey, 2004). In general, the positive effect of the PSP is apparent in several studies [Prechelt and Unger, 2000; Wesslén, 2000; Rombach et al., 2008].

Depending on the type of data collection, forms for PSP evaluations can be defective [Johnson and Disney, 1999]. The study showed that data collected by students manually on paper forms contain a great number of errors. However, these errors can be decreased by using tools that calculate the derived PSP measures based on engineers' raw data [Rombach et al., 2008]. As shown above, many software professionals demand learning. Nonetheless, the results are not distinct.

Request Format. One study analyzes whether the way of requesting effort estimates affects these estimates [Jørgensen and Halkjelsvik, 2010]. Although from a strictly rational viewpoint, it should make no difference, the study assumes that the way of phrasing an effort estimation request influences the outcome and thus the accuracy of the estimate. The traditional approach ("How much effort is required to complete X?") is compared to an alternative approach ("How much can be completed in Y work-hours?"). The alternative approach may apply especially in incremental development or in case of prototyping if a subsystem needs to be implemented until a specific point in time. The effect was stronger in cases of short time-frames, that is, the stated work-hours in the alternative approach. The effect vanished in cases where work hours were closer to the median of the traditional effort estimate. However, as the study's results [Jørgensen and Halkjelsvik, 2010] show that the alternative approach provides lower, that is, more optimistic estimates compared to the traditional approach, the use of the alternative approach should be avoided.

Estimator's Characteristics

Estimator's characteristics are factors that are associated with the estimator himself/herself, that is, his/her knowledge, skills, behavior, and reference to the project. These characteristics are illustrated in Table 2 and described in the following.

Table 2: Overview of Estimator Characteristic Factors and the Identified Primary Studies

	Diligence	Estimator's Experience	Estimator's General Degree of Optimism	Estimator's Reference to the Project	Estimator's Role	Inconsistency of Estimates
Furulund and Moløkken-Østvold, 2007		O				
Grimstad and Jørgensen, 2007						X
Grimstad, Jørgensen, Moløkken-Østvold, 2005						
Jørgensen, 2004b		O		X	X	
Jørgensen et al., 2007		X	X			
Jørgensen and Moløkken, 2003				X		
Jørgensen and Moløkken-Østvold, 2004				X	X	
Lederer and Prasad, 1995	X			X		
Magazinović and Pernstål, 2008	X	X				
McDonald, 2005		X				
Moløkken-Østvold and Jørgensen, 2005a		O				
Moløkken-Østvold and Jørgensen, 2005b		X				
Morgenshtern et al., 2007		X				
Subramanian and Breslawski, 1995		O				
van Genuchten, 1991		O				

"O" indicates that a factor was proposed to be important in the corresponding study.
"X" indicates that a factor was evaluated in the corresponding study.

Diligence. No definition of diligence was found in the studies in which this factor is mentioned. The Oxford Dictionary states that diligence is "careful and persistent work or effort." As such, we see this factor as one of the characteristics of the estimator. Lack of diligence by system analysts and programmers is mentioned as a reason for inaccurate estimation [Lederer and Prasad, 1995]. Experts in a second study mention inadequate time spent on estimation as a problem in estimation accuracy [Magazinović and Pernstål, 2008]. In cases of lack of diligence, the estimate accuracy may suffer as a cause of inaccurate requirement assessment by the analysts and a less valuable development by the programmers.

Diligent work may in this context be a necessary condition for accurate estimation. As in many other contexts, high outcome quality may not be feasible in cases where high time pressure leads to shortcuts and negligence of diligent quality assurance [Costello, 1984; Austin, 2001].

Estimator's Experience. The estimators' experiences seem to be an important factor. According to project managers, lack of technical skills is one important reason for estimation errors [Furulund and Moløkken-Østvold, 2007; van Genuchten, 1991]. The studies' participants perhaps assumed that lack of estimators' technical skills leads to inaccurate estimates, as it is more difficult to estimate software tasks you do not understand. This phenomenon is also indicated in further studies [Magazinović and Pernstål, 2008; Moløkken-Østvold and Jørgensen, 2005b;

Jørgensen et al., 2007]. But experience is not limited to technical skills. It also includes managerial experience [Subramanian and Breslawski, 1995]. Such experience covers fundamental knowledge about software development in different areas such as staffing, planning, and production. A further study showed that the number of previously managed projects and the degree of application experience are highly correlated with estimation error [Morgenshtern et al., 2007]. Consequently, the number of projects seems to be more important than the number of years of experience. This seems reasonable because a single long-term project might not be as high a contribution to an estimator's experience as several short-term projects.

Two of Jørgensen's studies postulate high skills in general [Jørgensen, 2004b; Moløkken-Østvold and Jørgensen, 2005a]. Besides the regression model, his survey on development tasks provides reasons for accurate estimates for the analyzed projects [Jørgensen, 2004b]. Such skill-related reasons were, for example, "good knowledge and flexibility concerning how to implement requirements," "much experience with similar tasks," and "long experience with this type of task" [Jørgensen, 2004b, pp. 311–312].

McDonald [2005] found support for a general demand for highly skilled estimators. It is shown that average team experience is positively correlated with high estimates (in that study, higher estimates were the more realistic ones). Overall, broad demand for highly skilled estimators (in general, as well as in technical and managerial dimensions) could be found.

Estimator's General Degree of Optimism. Among others, one study relates an estimator's general degree of optimism to the optimism in software effort estimates [Jørgensen et al., 2007]. The study is based on previous findings that more accurate and less optimistic effort estimates are to some extent linked to estimator's previous experience in the context of the estimation task.

The researchers [Jørgensen et al., 2007] used several measures for optimism (explanatory style, life orientation, and self-assessed optimism). First, the results show that optimism on previous estimation tasks is possibly the best indicator for optimism on future tasks. As such information may not be available, the need for an alternative way to assess an estimator's degree of optimism is apparent. Second, the study indicates that the estimators' self-assessments are the best indicator for their general level of optimism. Thus, estimation accuracy might be improved by selecting estimators with a lower-than-average optimism self-assessment.

Estimator's Reference to the Project. It may be difficult to estimate the effort needed to complete a task if the estimator herself/himself does not conduct the actual development. Lederer and Prasad [1995] indicate the significance of a lack of participation in the estimation process. Although this factor is not highly ranked by the study's participants, the significant correlation with inaccuracy corroborates the importance of including analysts and developers in the estimation process. This seems reasonable, as the study's participants were mainly IS managers and not developers. Managers agreeing to this factor would criticize their decision to not include analysts and developers in the estimation process. Jørgensen and Moløkken-Østvold conducted two studies [Jørgensen and Moløkken, 2003; Jørgensen and Moløkken-Østvold, 2004] that also allude to this factor. Their results indicate that task effort, which was estimated by the developers themselves, was more consistent with the estimate or rather that estimates of projects provided by a person not involved in the development process and led to more inaccurate results.

Apart from that, this factor might be related to the accountability factor. Possibly, people are made responsible for estimation accuracy when they develop the estimated task themselves. Nevertheless, the authors are aware of cases where expert teams are appointed to projects just for the estimation process. In those cases, the estimators will be held responsible to a certain degree for accuracy, although they are not involved in the development process. Therefore, we separated these two factors.

Estimator's Role. Jørgensen [2004b] refers to this factor as the company role. The differentiation of project managers and software developers providing estimates is one factor in his linear regression model that is used to explain estimate deviances. The results suggested that project managers provide more accurate estimates. Jørgensen pointed out that, possibly, this can be ascribed to the fact that developers most often estimate only their own work, whereas managers are more experienced in an overall project's estimation. Another of Jørgensen's studies picks up this regression analysis with an extended dataset [Jørgensen and Moløkken-Østvold, 2004], showing similar results. Projects estimated by software developers tended to have more inaccurate estimates. This factor might be one of the more important ones, as it is part of the regression model, even with small sample sizes only.

Inconsistency of Estimates. The inconsistency of expert judgments is researched in a single study [Grimstad and Jørgensen, 2007]. It denotes the deviation between two effort estimates made by the same expert on the same task.

If the same task is not consistently estimated, at least one of the estimates is inaccurate. Thus, the bias can be assigned toward overestimation as well as underestimation of the effort. The experts that participated in this experiment did not receive any training between the two estimates. The results show that the degree of inconsistency averages 71 percent. Neither of the different estimation tasks was estimated consistently by the participants, nor was one of the experts consistent on all six tasks that were estimated twice. Three reasons explain why the degree of inconsistency might be even higher in reality than in the artificially-created environment of the experiment. First, this might result from the description of the estimation tasks in the experiment which did not include any misleading information and was, therefore, consistent-friendly. Second, the study showed that the degree of inconsistency increases with the size of the estimation task, and the tasks in this study were rather small. Third, the participants were highly educated and experienced as software consultants. Estimators with lower qualifications might perform even worse.

Project to Be Estimated

This section describes the factors related to the project itself. Such factors are features that are inherent in the development task (e.g., level of innovation, task complexity) or that surround the estimation process during the course of the project (e.g., lying, neutrality and relevance of information). Table 3 provides the according overview.

Table 3: Overview of Project to Be Estimated Factors and the Identified Primary Studies

	Changes in Personnel	Clear Project Goals	Customer Expectations	Development Process	Newness of Technology	Lying	Neutrality and Relevance of Information	Overlooked Tasks	Requirements Specification	Task Complexity
Aranda and Easterbrook, 2005			X							
Connolly and Dean, 1997										X
Furulund and Moløkken-Østvold, 2007					O			O	O	
Glass, Rost, and Matook, 2008						O				
Gray et al., 1999										X
Grimstad et al., 2005		X	X					X	O	O
Jørgensen, 2004b									O	
Jørgensen and Grimstad, 2008			X				X			
Jørgensen and Moløkken, 2003										X
Jørgensen and Moløkken-Østvold, 2004								O	O	
Jørgensen and Sjøberg, 2001			X							
Jørgensen and Sjøberg, 2004			X							
Lederer et al., 1990	O							O		
Lederer and Prasad, 1995	X	X						O		
Magazinović and Pernstål, 2008									X	
Moløkken-Østvold and Jørgensen, 2005a				X	O				O	O
Morgenshtern et al., 2007		X								
Subramanian and Breslawski, 1995					O					
van Genuchten, 1991								O	O	

"O" indicates that a factor was proposed to be important in the corresponding study.
"X" indicates that a factor was evaluated in the corresponding study.

Changes in Personnel. When estimates are developed, changes in personnel that might occur during a software development process must be taken into consideration. Changes that cannot be anticipated during the estimation process (e.g., people unexpectedly quitting their job or getting sick) must lead to re-estimations. However, changes known beforehand (e.g., experts leaving a project after a certain period because they are staffed on another project) must be accounted for when estimating the project in the first place. In general, the effort that is caused by changes in personnel might be underestimated. The case study by Lederer et al. [1990] showed this factor to be important. Nearly 25 percent of the software managers and professionals named it in the interviews. Although the study was conducted in only one IS department, the results should not be one-sided as the participants were members of different areas of that department. A second study corroborates the factors' possible influence on accuracy of estimates [Lederer and Prasad, 1995].

Clear Project Goals. This factor refers to the transparency and the active communication of project goals to the people involved in the project. People may be more highly motivated if the overall goals and their personal objectives are clearly visible. It seems reasonable to be more accurate if one is aware of the task's importance. This may also be true for estimators. Nevertheless, the findings of the three studies that address this factor are not distinct. While the questionnaire survey with IS managers and analysts by Lederer and Prasad [1995] corroborates the assumption above, two further studies show differing results. From the perspective of other experts [Grimstad et al., 2005] and the analysis of governmental projects [Morgenshtern et al., 2007], no significant correlation between an unclear project definition and the estimation error can be shown.

Customer Expectations. Customer expectations refer to a phenomenon called the anchor effect. The anchor effect is apparent in many different contexts when people are asked to provide estimates or guesses [Mussweiler and Strack, 2001; Tversky and Kahneman, 1974]. It refers to the finding that expert estimates tend to be influenced when knowledge is available concerning initial estimates based on available data [Armstrong, 2001], that is, customer expectations concerning a development effort. At the beginning of an estimation process, people tend to unconsciously anchor on any given starting point and only subsequently adjust this anchor until a plausible but not necessarily good estimate is reached. High anchors thus lead to significantly higher estimates in comparison to estimates in which a low anchor or no anchor at all is available [Aranda and Easterbrook, 2005]. Overall, five studies address this factor.

Grimstad et al. [2005] show that according to software professionals' perceptions, customer expectation is important. However, software professionals may easily blame unrealistic customer expectations for the inaccuracy of their estimates, because then the professionals would not be responsible themselves. In their study, Aranda and Easterbrook [2005] differentiate between more experienced and less experienced estimators as the participants partly experienced real-life estimation before.

They showed that even experienced estimators could not elude this bias when being aware of the anchor. A generalization is questionable, however, as the sample size was rather small and consisted mainly of students. Two further studies provide stronger empirical evidence that anchors tend to affect estimates [Jørgensen and Sjøberg, 2001, 2004]. Additionally, the impact of customer expectations is also apparent in an article on irrelevant and misleading information [Jørgensen and Grimstad, 2008]. The results show that even when estimators were told that client expectations should not be used as a starting point, the estimates were strongly affected by the anchors.

These five studies showed that inaccurate estimates could partly be ascribed to the existence of anchors. Four of these studies dealt with anchors under controlled conditions. Aranda and Easterbrook [2005] showed no difference between students and software experts, and even the results of the studies that were (partly) conducted with students can be seen as empirical evidence of the factor's influence. Even though estimators do not seem to be aware of the anchor effect [Jørgensen and Sjøberg, 2001], the factor's influence nonetheless exists even when estimators are made aware of an anchor and told to ignore it.

Development Process. One study deals with the impacts of different development approaches on effort overruns [Moløkken-Østvold and Jørgensen, 2005a]. In that study, flexible and sequential model types are compared. Projects with flexible development approaches tend to lead to lower overruns. A possible explanation for that is a higher number of estimate revisions due to cyclic development (compared to sequential development). Additionally, flexible approaches are correlated with more positive dialogues between the customer and the contractor. A necessary condition for this is the maturity of the client. More customer feedback leads to more accurate estimates. The researchers made sure that their sample of eighteen companies was representative.

Newness of Technology. This factor covers newness of technology (i.e., tools, languages, and hardware). In this context, newness does not necessarily imply that a technology is innovative. Newness also refers to situations in which estimators are unfamiliar with a technology. Little or no data about newness exists, but new technologies are always more complicated for estimation purposes. Therefore, it seems consequential that estimates for projects applying new technology are less accurate than estimates for projects with established technology. Without experience, estimates may be too high or too low, as estimators might overestimate or underestimate the concomitant complexity. Although this factor seems to be easy to investigate, as the assessment of technology newness should not be complicated, we found only three studies that refer to their participants' subjective perceptions about this factor [Furulund and Moløkken-Østvold, 2007; Subramanian and Breslawski, 1995; Moløkken-Østvold and Jørgensen, 2005a].

Lying. Although lying—meaning "intentionally distorting the truth"—is generally ubiquitous and not a new phenomenon [Ford, 1996], we identified only one study that deals with lying in the context of software projects [Glass et al., 2008]. Estimators' performances depend on information provided by project stakeholders. Due to lying stakeholders, estimates are negatively influenced by the false information provided. The study [Glass et al., 2008] is not about effort estimation in particular, but software projects in general. Nevertheless, most of the participants' jobs (69 percent) were linked to the estimation process. Eighty-six percent of all respondents admitted that they experienced lying in software projects they worked on. The most frequently mentioned kind of lying occurred in effort or schedule estimation (66 percent). According to 18 percent of the respondents, lying about estimation in general happened in 50 percent of the projects investigated in the study. Ten percent of the respondents reported having experienced lying about estimation on every project. It was most often management who lied and the developers who knew. As this factor may change according to the project context and the team members, we categorized this factor as a project characteristic, although lying is, in general, a factor that depends on people.

Neutrality and Relevance of Information. Next to customer expectations, Jørgensen and Grimstad [2008] conducted three more experiments to investigate the consequences of irrelevant and misleading information on effort estimation accuracy. They used variation in wording (e.g., minor vs. major changes), possible future opportunities, and irrelevant information. The results show that irrelevant and misleading information cause estimation errors. With irrelevant information, a project seems to be more complex, which leads to higher estimates. Future opportunities encourage the estimators to deliver lower estimates. A slight change in a task description can lead to significantly different results. The results provided a good reason to believe that such influences affect estimates.

Overlooked Tasks. Tasks that are necessary are not regarded during effort estimation. The authors of the six articles we found allude to this factor [Furulund and Moløkken-Østvold, 2007; Jørgensen and Moløkken-Østvold, 2004; van Genuchten, 1991; Lederer et al., 1990; Lederer and Prasad, 1995; Grimstad et al., 2005]. Although this variety of studies deal with this reason for discrepancies between estimated and actual effort, the results reflect only the participant's subjective views and information about the extent of this bias is not given.

Requirements Specifications. This factor refers to the care bestowed on the specification of system requirements. Poor requirements result in inaccurate estimates because imprecise descriptions lead to (possibly wrong) interpretations of the requirements, and thus, incorrect assessments by the estimator. The studies indicate that a weak or ambiguous requirement specification causes inaccurate estimates, whereas well-defined requirement specifications prevent projects from overruns [Furulund and Moløkken-Østvold, 2007; Grimstad et al., 2005; Jørgensen, 2004b; Moløkken-Østvold and Jørgensen, 2005a; van Genuchten, 1991; Magazinović and Pernstål, 2008]. This requirement is also reflected in a further study, as good knowledge on how to specify requirements can lead to more accurate estimates [Jørgensen and Moløkken-Østvold, 2004].

Task Complexity. Tasks can vary, especially in size and complexity. In general, it is assumed that small tasks are overestimated and large tasks are underestimated [Jørgensen and Moløkken, 2003]. Large development tasks may correlate with a high number of people involved [Moløkken-Østvold and Jørgensen, 2005a]. This might be a reason for inaccuracy that can be explained by the increased need for communication and more potential sources of errors. This differentiation between optimism for difficult tasks and pessimism for easy tasks is confirmed in one article [Connolly and Dean, 1997]. Further empirical findings show that this factor is more complex [Grimstad et al., 2005; Jørgensen and Moløkken, 2003; Gray et al., 1999]. The results differ and no general implication can be derived. Overall, the tasks under research were rather small.

External Context

The external context comprises factors dealing with the surroundings of the estimation that are related to the client or general project characteristics that are determined by the client or in collaboration with the client. Table 4 shows the identified factors and the according primary studies.

Client Maturity. If the client does not possess the necessary knowledge and experience, it is more difficult for the contractor to deliver accurate estimates, as client maturity also comprises the users' ability to understand their own requirements. Therefore, client skills and the availability of capable decision makers and competent customers seem to be important [Lederer and Prasad, 1995; Moløkken-Østvold and Jørgensen, 2005a; Grimstad et al., 2005]. It may be true that client maturity impacts estimation accuracy. But it needs to be clarified that the results of the studies in this section reflect just contractors' subjective perceptions. It seems common to blame other people involved in the estimation process.

Table 4: Overview of External Context Factors and the Identified Primary Studies

	Client Maturity	Client Priority	Collaboration/ Communication	Contract Type	Project Priority
Grimstad et al., 2005	X		X		
Jørgensen, 2004b		X			
Jørgensen and Moløkken, 2003		X		X	
Lederer et al., 1990			O		
Lederer and Prasad, 1995	X		X		
Moløkken-Østvold and Furulund, 2007			X		
Moløkken-Østvold and Jørgensen, 2005a	O		O	O	O

"O" indicates that a factor was proposed to be important in the corresponding study.
"X" indicates that a factor was evaluated in the corresponding study.

Client Priority. Jørgensen [2004b] states that client priorities can be divided into three categories: time, cost, and quality. In Jørgensen's study, it is shown that estimation accuracy decreases when client priority is on time-to-delivery. This phenomenon can be explained by the behavior of the estimators and developers with regard to task priority. If the precedence is on time-of-delivery, the effort to reduce the probability that effort estimates are exceeded may be lower than otherwise. A second study [Jørgensen and Moløkken, 2003] confirms this finding. Jørgensen [2004b] conducted a regression analysis of forty-nine projects with regard to a variety of factors. Most of the factors are excluded from the regression model, as no significance could be shown. Client priority is shown to be significant even with this small sample size. Therefore, a strong correlation is indicated. The effect was shown only for the difference between priority on time and the priority on cost or quality. Further research is needed to analyze this triangle more deeply.

Collaboration/Communication. This factor refers to internal collaboration/communication, as well as the client-vendor collaboration/communication. Both parties need to collaborate well to enable accurate estimates. Lack of communication may lead to misunderstandings and, therefore, inaccurate estimates of tasks to be developed. Several questionnaire and interview studies [Grimstad et al., 2005; Moløkken-Østvold and Jørgensen, 2005a; Lederer et al., 1990; Lederer and Prasad, 1995] indicate the importance of good communication for preventing overruns, and lack of communication for causing overruns. Moløkken-Østvold and Furulund [2007] address customer collaboration in one of their articles. They compared projects having daily client communication with projects where communication occurred less often. In this study, daily client communication was associated with small overruns, whereas projects with less communication were significantly different and resulted in greater overruns. This result must be considered carefully, as other influences might be causing the difference in accuracy. For instance, the mean and median project size was larger for the daily communication group. While this is usually considered a reason for larger deviations between actual and estimated effort, the technical knowledge of this group was also higher. Therefore, the difference in accuracy in this study seems to be caused by a concurrence of several aspects.

Contract Type. The two possible extremes of this factor are fixed-price and time-and-material contracts. Although hybrid forms exist, fixed-price and time-and-material are the most common contracts. Research on this factor acts on the assumptions that fixed-price projects tend to be overestimated, as contractors are greatly interested in honoring the budget because they will lose money if the overruns occur. Surprisingly, the only study comparing projects with different contract types indicated that effort estimates and the projects' contract types are inversely related [Jørgensen and Moløkken, 2003]. Tasks that were paid per hour were less underestimated than fixed-price ones. Therefore, Jørgensen and Moløkken [2003] argue that the relation between contract type and estimation accuracy is more complex. Reasons for estimation accuracy given in the study showed that fixed-price estimates are adjusted to customer expectations. A possible conclusion might be that contractors tend to provide bids that will not pay off, rather than lose a contract. Only one of the interviewed software managers in a second study [Moløkken-Østvold and Jørgensen, 2005a] referred to a fixed-price contract as a reason for accurate estimates.

Project Priority. One study mentions project priority as influential on estimation accuracy [Moløkken-Østvold and Jørgensen, 2005a]. As the study's main goal is to identify the differences between sequential and flexible development approaches, the factors mentioned in the interviews were only a side effect and were not analyzed any further. For this reason, further research needs to be conducted to determine the factor's importance. The factor refers to the high priority of a project regarding the collaboration with the client. High priority was not defined in that study. It may be assumed that estimators are more thorough if the customer is of high importance to the employer. As a result, more effort in developing an estimate will be expended. This might ultimately lead to a more accurate assessment of the project's tasks.

Research Type

In this section, we provide the answers to the research questions (RQ) 2, 3, 4, and 5. First, we analyze which research methods were applied. We then present the results of our quality assessments. Finally, we address the provided strength of evidence and the extent to which the factors affect estimation accuracy.

Research Method

The identified primary studies can be divided into five categories according to research type: survey, experiment, case study, history-based evaluation, and real-life evaluation. We used Jørgensen and Shepperd's [2007] consolidated list of research methods employed in software effort estimation studies. Not all of their approaches are included in our study, due to our exclusion criteria for articles (cf. Section III). Assessing the limitations of research approaches without considering the context of the studies that use them is insufficient. We indicate only why a research approach may be problematic to assess the influence of factors affecting accuracy of the estimates.

- Surveys include questionnaire-based and interview-based studies. In the case of questionnaires, the data collection is controllable only to a certain degree (e.g., the respondents might misinterpret the questions). In most cases, such information represents only individual beliefs and can be used only as an indicator.
- Experiments are more suitable for determining causal relationships [Schutt, 2009]. By controlling the environment, researchers can gain insight to different designs of the estimation setting. However, a controlled environment limits the applicability of the study to real life.
- Case studies are in-depth studies of factors surrounding the estimation process. Although with such studies a deeper understanding (e.g., in comparison with surveys) is possible, a statistical generalization of the results is not without problems [Eisenhardt, 1989].
- History-based evaluations are based on previously completed projects. Even though those studies can rely on specific and possibly more objective project characteristics, the results may yield only limited value as only occasional clarification of the data can be given (e.g., by interviewing project managers).
- Real-life evaluations are suitable for delivering results as they are settled in realistic contexts. The difference in case studies is in the depth of the analysis. As no interference on behalf of the researchers during the evaluation process can be assumed, only very limited possibilities for causal analysis exist.

Considering RQ2, Figure 2 provides the general distribution of research methods applied. Predominantly, experiments are conducted (41 percent). Apart from that, surveys (28 percent) and real-life evaluations (18 percent) are often applied in effort estimation contexts. The remaining studies used case studies (8 percent) or history-based evaluations (5 percent). In some cases, researchers applied more than one research approach in their studies. Accordingly, the total number is higher than the number of primary studies (cf. the detailed overview in Appendix D).

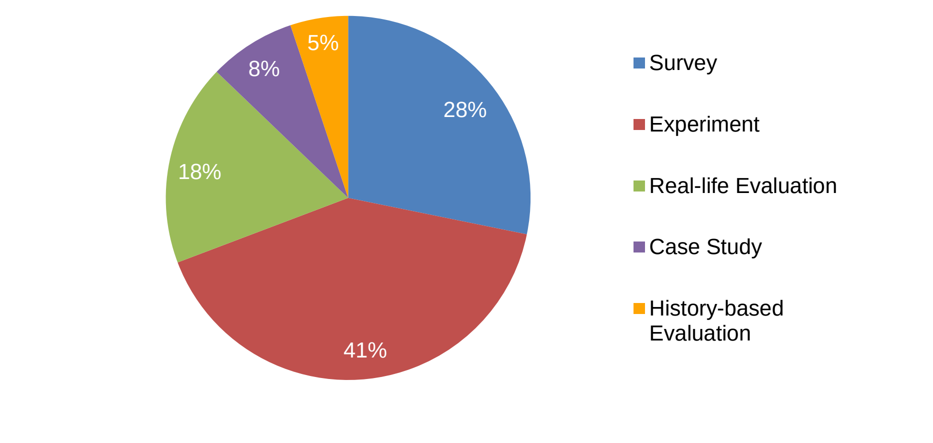

Figure 2. Primary Studies' Distribution over Research Method

For a more differentiated picture, we provide in Figure 3 a bubble plot of the identified publications concerning contribution type (proposal of a factor vs. evaluation of a factor) and research type (the research methods applied). The numbers concerning contribution type and research type extend the number of identified factors, as several factors were researched in different ways. Therefore, in many cases those factors contribute in both ways (i.e., proposal and evaluation) to the current body of knowledge. The numbers do not represent the importance of factors, that is, the strength of the impact on estimation accuracy, but rather provide an overview of which methods were applied for what purpose.

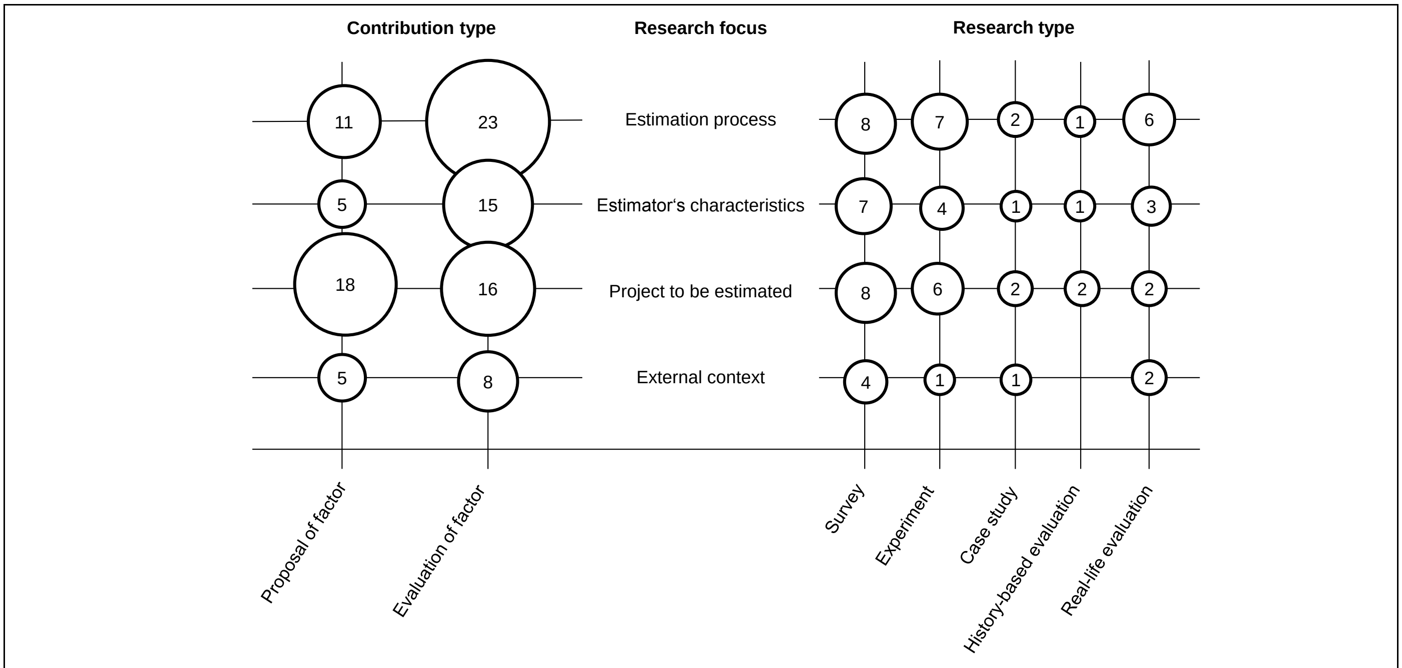

Figure 3. Factor Mapping According to Contribution Type, Research Focus, and Research Type

To analyze the timely development, we split the studies into two groups by separating the studies that were published before 2000 and since 2000 (cf. Appendix D). Obviously, research addressing factors that affect accuracy of software development effort estimation is increasing. While we identified six studies published between 1990 and 1999, twenty-six studies were published since 2000. Whereas other research approaches do not seem to differ, a noticeable increase occurred in using real-life evaluation (till 2000: one; since 2000: seven) and experiments (till 2000: four; since 2000: 13) (cf. Appendix D).

Quality Assessment

The detailed results of our quality assessments (cf. RQ3) are shown in Appendix E. These assessments are subjective and, therefore, are generalizable to only a limited degree. However, as two researchers independently answered the questions with a consistency rate above 85 percent, the inter-rater reliability exceeded the recommended threshold [Nunnally, 1978]. In cases of deviations, the assessments were discussed, along with an additional analysis of the corresponding article until agreement was reached. Overall, the average score concerning the quality assessments was 7.5 (11 being the highest possible score).

Table 5 provides an overview of the aggregated results. We ordered the questions in descending order according to the rating results. We clustered the questions according to their share of best answers (Yes = 1):

- Group 1: Questions with more than two thirds (> ⅔) of the studies ranked highest.
- Group 2: Questions with more than one third (> ⅓) but less than two thirds (≤ ⅔) of the studies ranked highest.
- Group 3: Questions with less than one third (≤ ⅓) of the studies ranked highest.

Considering the first group, the majority of the studies provides a rationale for undertaking the study (Q1), information about how data was collected (Q5), information about and justification of data analysis approaches (Q6), and a clear statement of the findings (Q8). Concerning Q6, the most common reason for 25 percent of the studies being rated 0.5 (to some extent) was that authors did not justify their method of data analysis. However, the general assessment shows that only slight improvements can be expected to be reached in future studies concerning these four questions.

Table 5: Aggregated Quality Assessment Results					
Group	Question	Yes	To some extent	No	Total
1	Q8	29 (91%)	3 (9%)	0 (0%)	32 (100%)
	Q1	27 (84%)	5 (16%)	0 (0%)	32 (100%)
	Q5	26 (81%)	6 (19%)	0 (0%)	32 (100%)
	Q6	22 (69%)	8 (25%)	2 (6%)	32 (100%)
2	Q3	19 (59%)	10 (31%)	3 (9%)	32 (100%)
	Q2	18 (56%)	13 (41%)	1 (3%)	32 (100%)
	Q11	18 (56%)	4 (13%)	10 (31%)	32 (100%)
	Q10	17 (53%)	8 (25%)	7 (22%)	32 (100%)
3	Q7	8 (25%)	21 (66%)	3 (9%)	32 (100%)
	Q4	4 (13%)	19 (59%)	9 (28%)	32 (100%)
	Q9	2 (6%)	1 (3%)	29 (91%)	32 (100%)
Question numbers refer to Appendix C.					

Analyzing questions in Group 2 yields two major findings. First, almost all studies provide a description of a study context (Q3, 91 percent) as well as a justification and description for a research design (Q2, 97 percent). However, in 31 percent and 41 percent of the studies, respectively, the authors could provide more insights into their approaches. Second, considering discussions of the credibility of the findings (Q10) and limitation of the study (Q11), the results are slightly worse. Although almost the same number of studies performs well concerning these criteria, 22 percent and 31 percent, respectively, provide no information at all.

The third group covers questions on whether researchers provide “sufficient” data to support the findings (Q7), whether researchers explain how the study sample (participants or cases) was identified and selected, the justification for a selection (Q4), and whether researchers critically examined their own role, potential bias, and influence during the formulation of research questions, sample recruitment, data collection, and analysis and selection of data for presentation (Q9). Only a few studies (9 percent) provide information about critically examining potential biases caused by researchers influencing the study design. Most studies (75 percent) would benefit from including more data in the analyses. Although the majority of studies provide information about how the study sample was identified and selected, a justification for corresponding approaches is only rarely provided.

Strength of Evidence

We are interested also in the strength of evidence of the primary studies (cf. RQ4). For our mapping study, we used the definitions from the Grading of Recommendations Assessment, Development and Evaluation (GRADE) working group that was used in software engineering research previously [Ali et al., 2010; Dybå and Dingsøyr, 2008a, Dybå and Dingsøyr, 2008]. Accordingly, the four different grades that follow are used to assess the strength of evidence (i.e., our confidence in the estimate of the effect that a factor is shown to produce on accuracy of software development effort estimation, as adopted from Atkins et al., 2004):

- Very low: Any estimate of effect is highly uncertain.
- Low: Further research is highly likely to impact on our confidence in the estimate of effect and is likely to change the estimate.
- Moderate: Further research is likely to impact moderately on our confidence in the estimate of effect and may change the estimate.
- High: Further research is highly unlikely to change our confidence in the estimate of effect.

To judge the strength of evidence, the following elements were used: study design, study quality, consistency, and directness. In this context, we also considered the studies’ quality assessments (cf. Appendix E; especially Q2, Q4, and Q9). Based on these assessments and the similarity of results across studies as well as the studies’ directness (extent to which the results are similar to contexts of interest), we assessed the strength of evidence for our identified primary studies. Again, the judgments were provided by two researchers independently giving an equally high inter-rater reliability (81 percent). The distribution of the studies’ GRADE assessments is shown in Figure 4.

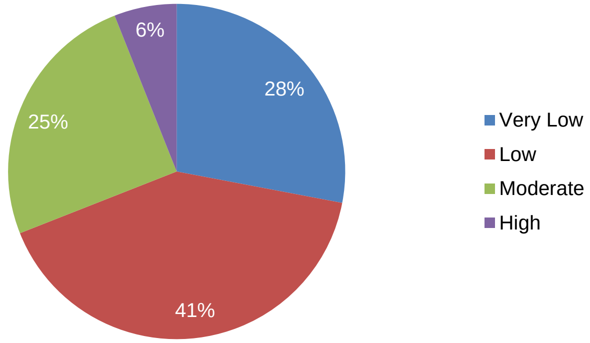

Figure 4. Primary Study Distribution over GRADE Assessment Categories

Despite the fact that many studies use experiments, only two studies were evaluated to provide a high strength of evidence. In most cases this can be attributed to the studies' context (very small sample sizes, single companies). Therefore, we would expect that most of the studies are likely to result in different findings when the factors under consideration are researched in different contexts.

Factors' Impact on Estimation Accuracy

In this subsection, we analyze the extent to which the identified factors affect accuracy of software development effort estimation (cf. RQ5). To determine such an extent, a factor needs to be evaluated (those that are marked by an "X" in Tables 2, 3, 4, and 5), and its causal relationship to estimation accuracy needs to be analyzed (e.g., based on a controlled experiment). However, studies that fulfill these criteria do not report on concrete extents of the analyzed impacts, but rather whether the differences are significant.

For instance, analysis of customer expectations [Jørgensen and Sjøberg, 2004, p. 320] showed that the "median estimated effort of the software professionals in the LOW group was 77 work-hours, and in the HIGH group 632 work-hours, i.e., more than eight times as high!" The attempt to determine this impact may be misleading as the concrete impact is likely to be influenced by the height of the expectation, the real effort, and the experience of the estimator, and thus cannot be assessed in general. Even though the results clearly indicate that anchors such as customer expectations should be avoided in the estimation, it is not feasible to determine the extent to which an anchor affects estimation accuracy. The same applies to research on the effect of irrelevant and misleading information [Jørgensen and Grimstad, 2008], initial risk analyses [Jørgensen, 2010], task complexity [Connolly and Dean, 1997], estimation experience [McDonald, 2005], and estimators' general degree of optimism [Jørgensen et al., 2007].

In general, studies that apply experiments in their analyses of software development effort estimation predominantly aim to investigate whether a specific factor influences estimation accuracy (e.g., "Does a change from the traditional request 'How much effort is required to complete X?' to the alternative 'How much can be completed in Y work-hours?' affect effort estimates?"; cf. Jørgensen and Halkjelsvik, 2010, p. 30) and not to what extent. At this point, we see the need to distinguish between factors that can be controlled or avoided by project management and factors that are out of the control of project management and thus need to be regarded in estimation processes. In the case of the former, it is obviously sufficient to know whether a significant impact exists on estimation accuracy in order to derive implications for future projects. In the case of the latter, the extent of the impact must be known in order to consider the factor adequately in, for instance, formal estimation models. As previous research mainly focused on factors that can be controlled (e.g., request format, use of experienced estimators) or avoided (e.g., anchors, irrelevant information) by project management, the objective of future studies should be to analyze the extent of the impact of a specific factor which cannot be avoided (e.g., client maturity, lying, lack of historical data).

V. DISCUSSION

Our systematic mapping identified thirty-two factors (RQ1) that potentially affect accuracy of software development effort estimates. These factors were researched in articles published between 1990 and 2010. Although we identified four categories of research focus (estimation process, estimator's characteristics, project to be estimated, and external context; cf. Section IV), the majority of the factors (twenty-one out of thirty-two, cf. Tables 2 and 4) are related to the estimation process and characteristics of the projects to be estimated. External context and estimators themselves need to be further analyzed, especially as, in most cases, estimates are based exclusively on experts' judgments.

The results show that software professionals are aware of the problem surrounding the estimation process and might know the causes of estimation inaccuracy. However, many studies do not distinguish between reasons for inaccurate estimates (cf. the factors in Section IV) and causes for deviations between estimated and actual effort (e.g., change requests). While both types of causes lead to deviations from project plans, only the former provide the opportunity to make estimates more realistic. Causes such as change requests need to be handled by project management, as these instances typically must be considered in re-estimates.

The thirty-two identified factors that potentially influence accuracy of software development effort estimation can be seen as a cause for inaccurate estimates and, thus, deviations between estimated and actual effort in software projects. As in most cases, effort estimates are based on expert judgments, and the factors that refer to estimator characteristics seem to be important. Learning and feedback mechanisms should help estimators improve their abilities/skills and might at least reduce the inconsistency of expert judgments. Apart from that, estimators should be given sufficient time to deliver their estimates. Diligence and careful examination of estimates are factors that, from our perspective, should receive attention in the future. This might also reduce the number of tasks that are overlooked at the beginning of a project.

Considering RQ2, we found that surveys and experiments are the research methods that are most often applied in the identified studies (cf. Figure 3). While (controlled) experiments are helpful in uncovering causal relations, surveys most often explore participants' subjective perceptions. Surveys and experiments show two counterparts. While surveys are relatively easy to conduct and helpful in eliciting expert knowledge, experiments are difficult to conduct under real-life conditions (e.g., finding participants) and cover the behavior of participants. As a consequence, experiments are in many cases conducted with students instead of software professionals. This results in a threat to the external validity of the results. Further contrasting these two methods, an important point to discuss is the matter of causality. Based on surveys, it is not feasible to derive causal relations, but correlations are more feasible. Experiments may provide insights about the causality of factors, but likely under artificial conditions. Surveys, on the one hand, may cover a variety of factors. Experiments, on the other hand, seem to cover only one or a few factors. In the future, case studies could provide deeper insights into the understanding of inaccurate estimates.

Our assessments of the studies' quality (cf. RQ3) showed an average quality level of 7.5 (11 being the highest possible score). Further, the results indicate that basic information concerning study design (e.g., motivation, clear statements of the findings) is provided in most cases. However, quality could be increased especially by discussing the credibility of the findings and limitations of the study design. We believe that neglecting such information can be explained by the affinity of researchers to focus on the positive aspects of their study. Finally, we encourage researchers to more critically examine their own role and the potential resulting bias when designing their studies. For instance, many of the studies we analyzed would benefit from providing a justification for a specific approach of identifying and selecting their samples.

When considering the strength of evidence of the identified studies (cf. RQ4), it seems that it is rather complicated to provide a high level. Our assessments led to only 6 percent of the studies being assigned to this category. Although a broad range of studies exist that apply (controlled) experiments and thus causal relations, settings of such studies (e.g., small sample size, student participants) lead to a limitation of our confidence in the estimate of effect (i.e., the effect that a factor is shown to produce on accuracy of software development effort estimation). Whereas it is unlikely to conduct experiments with larger sample sizes, a replication of existing experiments might help to strengthen the evidence concerning the factors' influence on estimation accuracy.

With regard to our last research question (cf. RQ5), we found that in many cases a single factor is researched and, additionally, the participants provide reasons for accurate and inaccurate estimates. These factors are referred to without (proper) definitions and without information about the factors' biases. Thus, the corresponding studies provide only limited information about the extent to which estimates are influenced by the factors. However, the variety of studies applying experiments show a causal influence of selected factors on estimation accuracy. The results are nonetheless limited to special contexts, as the majority of the experiments are conducted in single companies or industries with a small sample size, or with students. Moreover, it seems rather difficult to analyze the extent to which factors influence estimation accuracy. Most of the studies apply a pragmatic approach and focus on factors which can be avoided by project management by designing the estimation process accordingly (e.g., request format, use of experienced estimators, avoidance of anchors and irrelevant information). While the results are undoubtedly valuable for research and practice concerning effort estimation, the discipline would benefit from empirical models that provide more insight into the extent to which effort estimation is influenced by single factors or a combination of several factors.

In general, the differentiation of what is in the focus of the estimation is interesting as well. In many studies, no definition or characterization of what constitutes a task or project is made clear. Accordingly, we need to assume that diverging types of tasks were analyzed. At this point, we are not able to consider the corresponding consequences.

The findings in our mapping study may be partly controversial in terms of how to optimally design the effort estimation process—especially the decision about whether an estimator or a manager should develop the estimate. This aspect is the result of studies conducted in different contexts and through the application of different research methods. Although it would be more helpful for practitioners to provide a distinct rule for the design of the effort estimation process, in this systematic mapping we are able to present only the results provided by previous research studies. At this point, the relevant conclusion is for researchers; this aspect clearly calls for a deeper investigation.

Although we investigated factors with an impact on the accuracy of software development effort estimates irrespective of the estimation technique applied, the factors' majority seems to be linked to the judgment of experts—the most predominantly applied technique [Basten and Mellis, 2011]. This result can be attributed to the exclusion of studies dealing with the improvement of single estimation techniques. Furthermore, formal estimation techniques aim to account for cost drivers that affect the effort, for example, in relation to source lines of codes or function points [Boehm et al., 1995]. Thus, accuracy factors seem to be more relevant for the human impact. Despite the focus on expert estimation, our results are of interest for studies on software development effort estimation in general, as expert judgment is also relevant for meaningful effort estimation models [Grimstad and Jørgensen, 2007].

The differentiation between factors affecting the effort of a software development project (e.g., seventeen cost drivers in CoCoMo II) and factors affecting accuracy of the software development effort estimation (thirty-two factors in our study) is an interesting aspect. The comparison of our identified factors with the cost drivers in CoCoMo II [Boehm, 2000] in Appendix F shows similarities. (These are not necessarily perfect matches, but, as our comparison aims to show coherences, we believe this mapping to be suitable.) The comparison also shows, for example, that the experience of team members is considered in more detail in the CoCoMo II model. Eight (of thirty-two) accuracy factors match ten (of seventeen) cost drivers. Ideally, future research on effort estimation will concentrate on the combined effect of effort and accuracy factors.

VI. THREATS TO VALIDITY

We analyze the validity of our mapping study with regard to reliability, construct validity, and internal as well as external validity (cf. also the mapping in Engström and Runeson, 2011).

Concerning the reliability of our study, we state the following. As we provide a protocol of our search, we believe that this search for accuracy factors is reproducible with the same results. We defined the timeframe, the journals under consideration, and the way the additional searches (both backward and forward) were conducted. The classification of the identified factors follows a structured approach (cf. Section III), but may lead to different schemes when researchers with different backgrounds conduct this approach. Although not all factors are covered with absolute certainty, we believe that our findings cover most essential research on software development effort estimation over the past decades. On the one hand, our results include several studies providing lists of factors associated with estimation inaccuracy. As such, our findings can be seen as representative concerning the relevant factors from the point of view of practitioners. On the other hand, our search process is based on an adequate approach. Following the recommendations by Webster and Watson [2002], we started with a set of highly relevant and high-quality outlets and subsequently applied searches backward and forward (cf. Section III).

As the field of effort estimation was studied for a lengthy period of time, we believe that the studies in which experts refer to potential accuracy factors know to what they are referring. Despite the lack of definitions in many cases, we believe that our interpretations of the factors are valid, as this part of our research was conducted by two researchers independently in order to secure construct validity.

Selection bias might be a problem. Due to our restriction to a manual issue-by-issue search, we reduced the publications in which we searched. We limited the number of journals and the timeframe in which we searched for relevant publications. As an overall search was not feasible, we limited this bias as far as possible by our search forward and backward.

Our results are limited to the contexts in which the analyzed studies were conducted. As seventeen of the thirty-two articles originate from studies of the Simula Research Laboratory in Norway, their results need to be validated in other contexts. Although these studies provided a comparatively high strength of evidence, their results might not apply to effort estimation in general. With a few exceptions (e.g., Australia and Israel), all studies were conducted in the United States and Europe, so that analyses in other geographical regions should be part of future research. In regard to internal validity, only negligible threats stain our quantitative results. As many studies provide only limited information about the factors' definitions and their effects, we needed to conjecture in several cases. At this point, other researchers might find slightly different conclusions when replicating our study design.

VII. IMPLICATIONS FOR RESEARCH AND PRACTICE

Implications for Research

To provide implications for research, we focus on various issues concerning study design, that is, the definition of key concepts, the perspective taken, research methods applied, and selected questions of the quality assessments.

The lack of a clear definition concerning the term effort estimate was previously addressed [Grimstad et al., 2006]. Similarly, many studies do not clearly define the factors under investigation. This problem predominantly applies to factors researched solely in studies aimed at identifying factor lists as comprehensively as possible (cf. Section I). Exemplary factors include diligence, clear project goals, historical data, and careful examination of the estimate. Without clear definitions, improvements in effort estimation are difficult to realize, as it is unclear how to regard factors that are understandable to a limited degree only. In an ongoing study, we experienced the beneficial result of involving software professionals during the initial phase of research studies (e.g., to define what effort estimate means) to secure a common and adequate understanding of a study's concepts. Otherwise, findings are of rather limited value for software professionals.

The clients' perspective should be examined to a larger extent as it is often blamed for inaccurate estimates. However, the client perspective was widely neglected in previous research. So far, only a few studies (i.e., Moløkken-Østvold and Furulund, 2007; Grimstad et al., 2005; Jørgensen and Sjøberg, 2004) dealt with customer impact on estimation accuracy and related insights into the factors that cover characteristics of the client seem to provide only limited insights. Moreover, and considering the characteristic of these studies in Appendix B, the customer's impact was assessed predominately from the contractors' point of view. Explicitly observing this impact from the client's perspective might thus lead to further insights. Furthermore, research addressed factors that are undoubtedly of high importance for effort estimation, but neglected factors that were previously proposed as being relevant. Examples include assessment of developer skills (e.g., van Genuchten, 1991), lying [Glass et al., 2008], and newness of technology [Furulund and Moløkken-Østvold, 2007; Subramanian and Breslawski, 1995; Moløkken-Østvold and Jørgensen, 2005a]. Choosing future study objectives, researchers should consider analyzing factors whose influence is yet unknown. A highly relevant change would be to develop approaches to assess the concrete extent to which factors affect accuracy of software development effort estimation.

Considering the distribution of research methods in Figure 2, case study research is a method that has been widely neglected in research concerning accuracy factors. Moreover, the timely distribution of research approaches in Appendix D shows that the three identified case studies occurred sporadically throughout the past decades. However, this research method is likely to provide in-depth insights into phenomena under investigation. Choosing this method might provide useful insights into the peculiarities of the estimation process. Specifically, it may be helpful to apply case study research to investigate the customer's role (cf. the previous implication) in developing estimates. Consequently, future research would benefit from in-depth insights concerning the customer's influence in the estimation process in addition to providing the requirements to be fulfilled.

Our quality assessments show an average score of 7.5 (11 being the highest; cf. Appendix E). A more detailed view on this result leads to the insight that most of the primary studies need to more extensively justify the applied study design (cf. Table 5). Researchers should argue for specific design decisions to enable a better understanding of possible influences of the study design on the credibility of the results. This predominantly applies to questions concerning potential biases and limitations (Q9 and Q11 in Table 5). The researcher's role in defining key concepts and designing studies may, for instance, lead to biases on behalf of the participants (e.g., adapted behavior in experiments). While we believe the average level of quality to be adequate, researchers should not miss describing the concrete setting of their studies, as this is a major threat to external validity and thus to the generalizability of the results.

Implications for Practice

We provide our implications for practice in a threefold way. First, we refer to the general application of the identified factor list. Second, we provide specific hints on how to improve effort estimation accuracy in practice. Finally, we encourage practitioners to more actively seek research collaborations by highlighting related benefits.

When estimates are delivered, they usually are subject to certain assumptions (e.g., requirements uncertainty, developer skills). These assumptions are considered to be risks at the beginning of the project and need to be controlled throughout the development process. The factors identified in this study help to consider the various risks throughout this process. The more factors or risks are controlled, the better the estimation and its control throughout the development process. In this context, the list of factors can also be used as a checklist for guidelines on effort estimation (e.g., Basten and Sunyaev, 2011). Systematic collection of experience data concerning development effort from ongoing projects helps to improve the estimation accuracy of future projects. The study by Furulund and Moløkken-Østvold [2007] shows that experience data can improve accuracy, but at the same time this study points out that many professionals lack relevant experience for a given context. Therefore, it is essential to continuously preserve project and estimation data. To collect experience data, our study provides a comprehensive list of factors that should be assessed along with data of the estimated and actual effort to derive useful models that can be applied to improve estimation accuracy. Software professionals should try to categorize software development tasks to better distinguish between different types of tasks. Based on such a categorization, it is easier to find analogies. In this context, justification for conducted estimates should be stored as well in order to ensure that analogies are not used misleadingly.

While the majority of research concerns accuracy factors related to the process and project, one of the most important factor groups is concerned with the characteristics of the estimator conducting the estimation. Therefore, decisions about estimators being assigned to projects should focus on estimator characteristics. Studies concerning such factors, for instance, show the relevance of considering estimators' experience in the concrete setting and the general level of optimism. The responsibility for providing estimates for software development effort should be assigned carefully. Relying solely on experienced software experts is not sufficient. We already mentioned that estimates are, in most cases, provided by experienced software experts. Although we showed that the estimators' qualification in terms of different kinds of experience is important, solely relying on experienced estimators may not be sufficient to receive accurate estimates. It was shown that even effort estimates delivered by experienced estimators were affected to a certain degree (e.g., customer expectations). The combination of expert knowledge and systematic developed estimates might help to improve estimation accuracy. From our point of view, it is less essential which estimation methods are combined, but more essential that estimates are not delivered based on a single method.

Software professionals are in the advantageous position of being able to collect experience data which is of immense relevance for effort estimation research. As software professionals in many cases do not have the time or resources for in-depth analyses of these data, they should seek collaboration with researchers. Based on our experience, such collaborations are usually perceived as burdensome by software professionals, but they are actually beneficial for both parties. As sole data collection is insufficient, software companies should seek support from research for data analysis. This would enable them to gather insights into the extent of factors influence on estimation accuracy based on their own experience. Finally, it would make a contribution to closing one of the open issues in this research stream.

VIII. CONCLUSION

Extensive research was conducted to investigate the surroundings of effort estimates that affect their accuracy. Although almost 70 percent of the identified articles dealt with three factors or fewer, only a few cases show a factor's causality toward accuracy. However, our findings (cf. Appendix D) indicate a shift from broad overview studies (mainly in the 1990s) toward controlled experiments concerning the impact of single factors (nowadays). While many studies provide a high level of quality (cf. Appendix E), the reporting of rationales for choosing a specific study design should in general be improved in future studies. Considering the identified surveys in particular, professionals seem to be aware of the various factors to a limited degree only. Researchers demonstrated the significant impact of a variety of factors on estimation accuracy (e.g., customer expectations, neutrality and relevance of information) that the participating professionals did not mention as being important. On the one hand, research provides useful insights into significant influences on estimation accuracy that enable software professionals to focus on neglected factors. On the other hand, future research should focus on assessing the extent of commonly mentioned factors, as this was rather neglected in the past (e.g., requirements uncertainty, developer skills).

REFERENCES

Editor's Note: The following reference list contains hyperlinks to World Wide Web pages. Readers who have the ability to access the Web directly from their word processor or are reading the article on the Web, can gain direct access to these linked references. Readers are warned, however, that:

1. These links existed as of the date of publication but are not guaranteed to be working thereafter.
2. The contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.
3. The author(s) of the Web pages, not AIS, is (are) responsible for the accuracy of their content.

4. The author(s) of this article, not AIS, is (are) responsible for the accuracy of the URL and version information.

Ali, M.S., M.A. Babar, L. Chen, and K.-J. Stol (2010) "A Systematic Review of Comparative Evidence of Aspect-oriented Programming", *Information and Software Technology,* (52)9, pp. 871–887.

Allen, B. (1982) "An Unmanaged Computer System Can Stop You Dead", *Harvard Business Review*, (60)6, pp. 77–87.

Aranda, J., and S. Easterbrook (2005) "Anchoring and Adjustment in Software Estimation", *ACM SIGSOFT Software Engineering Notes,* (30)5, pp. 346–355.

Armstrong, J.S. (2001) "The Forecasting Dictionary" in Armstrong, J.S. (ed.) *Principles of Forecasting: A Handbook for Researchers and Practitioners,* Boston, MA: Kluwer Academic Publishers, pp. 761–824.

Atkins, D., D. Best, P.A. Briss, M. Eccles, Y. Falck-Ytter, S. Flottorp, G.H. Guyatt, R.T. Harbour, M.C. Haugh, D. Henry, S. Hill, R. Jaeschke, G. Leng, A. Liberati, N. Magrini, J. Mason, P. Middleton, J. Mrukowicz, D. O'Connell, A.D. Oxman, B. Phillips, H.J. Schünemann, T.T.-T. Edejer, H. Varonen, G.E. Vist, J.W. Williams, and S. Zaza (2004) "Grading Quality of Evidence and Strength of Recommendations," *British Medical Journal,* (328)7454, pp. 1490–1496.

Austin, R.D. (2001) "The Effects of Time Pressure on Quality in Software Development: An Agency Model", *Information Systems Research,* (12)2, pp. 195–207.

Avison, D.E., and G. Torkzadeh (2009) *Information Systems Project Management,* Los Angeles, CA: Sage.

Basten, D., and W. Mellis (2011) "A Current Assessment of Software Development Effort Estimation" in IEEE Computer Society (ed.) *Proceedings of the 5th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, New York: IEEE Computer Society, pp. 235–244.

Basten, D., and A. Sunyaev (2011) "Guidelines for Software Development Effort Estimation", *IEEE Computer*, (44)10, pp. 88–90.

Boehm, B., B. Clark, E. Horowitz, C. Westland, R. Madachy, and R. Selby (1995) "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0", *Annals of Software Engineering,* (1)1, pp. 57–94.

Boehm, B.W. (2000) *Software Cost Estimation with COCOMO II,* Upper Saddle River, NJ: Prentice Hall PTR.

Briand, L.C., C.M. Differding, and H.D. Rombach (1996) "Practical Guidelines for Measurement-based Process Improvement", *Software Process: Improvement and Practice*, (2)4, pp. 253–280.

Burgess, C.J., and M. Lefley (2001) "Can Genetic Programming Improve Software Effort Estimation? A Comparative Evaluation", *Information and Software Technology,* (43)14, pp. 863–873.

Cadle, J., and D. Yeates (2008) *Project Management for Information Systems, fifth edition,* Harlow, UK: Pearson Prentice Hall.

CompTIA (2007) *Survey: Poor Communication Causes Most IT Project Failures. Inadequate Resource Planning, Unrealistic Deadlines,* also cited in "CompTIA Study", *Computerworld,* 2007.

Connolly, T., and D. Dean (1997) "Decomposed versus Holistic Estimates of Effort Required for Software Writing Tasks", *Management Science,* (43)7, pp. 1029–1045.

Costello, S.H. (1984) "Software Engineering Under Deadline Pressure", *ACM SIGSOFT Software Engineering Notes,* (9)5, pp. 15–19.

Dybå, T., and T. Dingsøyr (2008a) "Empirical Studies of Agile Software Development: A Systematic Review", *Information and Software Technology,* (50)9–10, pp. 833–859.

Dybå, T., and T. Dingsøyr (2008b) "Strength of Evidence in Systematic Reviews in Software Engineering" in ACM (ed.) *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement,* New York: ACM, pp. 178–187.

Dybå, T., T. Dingsøyr, and G.K. Hanssen (2007) "Applying Systematic Reviews to Diverse Study Types: An Experience Report" in IEEE Computer Society (ed.) *Proceedings of the 1st International Symposium on Empirical Software Engineering and Measurement,* pp. 225–234.

Eisenhardt, K.M. (1989) "Building Theories from Case Study Research", *The Academy of Management Review,* (14)4, pp. 532–550.

Engström, E., and P. Runeson (2011) "Software Product Line Testing—A Systematic Mapping Study", *Information and Software Technology*, (53)1, pp. 2–13.

Finnie, G.R., G.E. Wittig, and J.-M. Desharnais (1997) "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, pp. 281–289.

Ford, C.V. (1996) *Lies!, Lies!!, Llies!!!,* Washington, DC: American Psychiatric Press.

Furulund, K.M., and K. Moløkken-Østvold (2007) "Increasing Software Effort Estimation Accuracy—Using Experience Data, Estimation Models and Checklists" in IEEE Computer Society (ed.) *Proceedings of the Seventh International Conference on Quality Software,* New York: IEEE Computer Society Press, pp. 342–347.

Gibbert, M., W. Ruigrok, and B. Wicki (2008) "What Passes as a Rigorous Case Study?", *Strategic Management Journal,* (29)13, pp. 1465–1474.

Glass, R.L., J. Rost, and M.S. Matook (2008) "Lying on Software Projects", *IEEE Software*, (25)6, pp. 90–95.

Gray, A.R., S.G. MacDonell, and M. Shepperd (1999) "Factors Systematically Associated with Errors in Subjective Estimates of Software Development Effort: The Stability of Expert Judgment" in IEEE Computer Society (ed.) *Proceedings of the Sixth International Software Metrics Symposium,* New York: IEEE Computer Society Press, pp. 216–227.

Grimstad, S., and M. Jørgensen (2007) "Inconsistency of Expert Judgment-based Estimates of Software Development Effort", *Journal of Systems and Software,* (80)11, pp. 1770–1777.

Grimstad, S., M. Jørgensen, and K. Moløkken-Østvold (2005) "The Clients' Impact on Effort Estimation Accuracy in Software Development Projects" in IEEE Computer Society (ed.) *Proceedings of the 11th IEEE International Software Metrics Symposium,* New York: IEEE Computer Society Press, pp. 4–13.

Grimstad, S., M. Jørgensen, and K. Moløkken-Østvold (2006) "Software Effort Estimation Terminology: The Tower of Babel", *Information and Software Technology,* (48)4, pp. 302–310.

Henrion, M., G. Fischer, and T. Mullin (1993) "Divide and Conquer? Effects of Decomposition on Accuracy and Calibration of Subjective Probability Distribution", *Organizational Behavior and Human Decision Process,* (55)2, pp. 207–227.

Hora, S., N. Dodd, and A. Hora (1993) "The Use of Decomposition in Probability Assessment of Continuous Variables", *Behavioral Decision Making,* (6)2, pp. 133–147.

Humphrey, W.S. (2004) *Introduction to the Personal Software Process, twelfth edition,* Boston, MA: Addison-Wesley.

Jankowicz, D. (2004) *The Easy Guide to Repertory Grids,* Chichester, UK: Wiley.

Johnson, P.M., and A.M. Disney (1999) "A Critical Analysis of PSP Data Quality: Results from a Case Study", *Empirical Software Engineering,* (4)4, pp. 317–349.

Jørgensen, M. (2004a) "A Review of Studies on Expert Estimation of Software Development Effort", *Journal of Systems and Software,* (70)1–2, pp. 37–60.

Jørgensen, M. (2004b) "Regression Models of Software Development Effort Estimation Accuracy and Bias", *Empirical Software Engineering,* (9)4, pp. 297–314.

Jørgensen, M. (2004c) "Top-down and Bottom-up Expert Estimation of Software Development Effort", *Information and Software Technology,* (46)1, pp. 3–16.

Jørgensen, M. (2007) "A Critique of How We Measure and Interpret the Accuracy of Software Development Effort Estimation", *Proceedings of the First International Workshop on Software Productivity Analysis and Cost Estimation,* pp. 16–21.

Jørgensen, M. (2010) "Identification of More Risks Can Lead to Increased Over-optimism of and Over-confidence in Software Development Effort Estimates", *Information and Software Technology,* (52)5, pp. 506–516.

Jørgensen, M., B. Boehm, and S. Rifkin (2009) "Software Development Effort Estimation: Formal Models or Expert Judgment?", *IEEE Software,* (26)2, pp. 14–19.

Jørgensen, M., B. Faugli, and T.M. Gruschke (2007) "Characteristics of Software Engineers with Optimistic Predictions", *Journal of Systems and Software,* (80)9, pp. 1472–1482.

Jørgensen, M., and S. Grimstad (2008) “Avoiding Irrelevant and Misleading Information When Estimating Development Effort”, *IEEE Software,* (25)3, pp. 78–83.

Jørgensen, M., and T.M. Gruschke (2009) “The Impact of Lessons-learned Sessions on Effort Estimation and Uncertainty Assessments", *IEEE Transactions on Software Engineering*, (35)3, pp. 368–383.

Jørgensen, M., and T. Halkjelsvik (2010) “The Effects of Request Formats on Judgment-based Effort Estimation”, *Journal of Systems and Software,* (83)1, pp. 29–36.

Jørgensen, M., and K. Moløkken (2003) “Situational and Task Characteristics Systematically Associated with Accuracy of Software Development Effort Estimates” in Information Resources Management Association (ed.) *Proceedings of the Information Resources Management Association Conference,* Hershey, PA: Information Resources Management Association, pp. 824–826.

Jørgensen, M., and K. Moløkken-Østvold (2004) “Reasons for Software Effort Estimation Error: Impact of Respondent Role, Information Collection Approach, and Data Analysis Method”, *IEEE Transactions on Software Engineering,* (30)12, pp. 993–1007.

Jørgensen, M., and M. Shepperd (2007) “A Systematic Review of Software Development Cost Estimation Studies”, *IEEE Transactions on Software Engineering,* (33)1, pp. 33–53.

Jørgensen, M., and D.I.K. Sjøberg (2001) “Impact of Effort Estimates on Software Project Work”, *Information and Software Technology,* (43)16, pp. 939–948.

Jørgensen, M., and D.I.K. Sjøberg (2004) “The Impact of Customer Expectation on Software Development Effort Estimates”, *International Journal of Project Management,* (22)4, pp. 317–325.

Jurison, J. (1999) “Software Project Management: The Manager’s View”, *Communications of Association for Information Systems*, (2)17, pp. 1–57.

Kappelman, L.A., R. McKeeman, and L. Zhang (2006) “Early Warning Signs of IT Project Failure: The Dominant Dozen”, *Information Systems Management,* (23)4, pp. 31–36.

Kemerer, C.F. (1987) “An Empirical Validation of Software Cost Estimation Models”, *Communications of the ACM,* (30)5, pp. 416–429.

Kitchenham, B. (2007) *Guidelines for Performing Systematic Literature Reviews in Software Engineering,* Technical Report.

Laudon, K.C., and J.P. Laudon (2009) *Management Information Systems: Managing the Digital Firm, eleventh edition,* Upper Saddle Rive, NJr: Pearson.

Lederer, A.L., R. Mirani, B.S. Neo, C. Pollard, J. Prasad, and K. Ramamurthy (1990) “Information System Cost Estimating: A Management Perspective”, *MIS Quarterly,* (14)2, pp. 159–176.

Lederer, A.L., and J. Prasad (1993) “Information Systems Software Cost Estimating: A Current Assessment”, *Journal of Information Technology*, (8)1, pp. 22–33.

Lederer, A.L., and J. Prasad (1995) “Causes of Inaccurate Software Development Cost Estimates”, *Journal of Systems and Software,* (31)2, pp. 125–134.

Lederer, A.L., and J. Prasad (2000) “Software Management and Cost Estimating Error”, *Journal of Systems and Software,* (50)1, pp. 33–42.

Lerner, J.S., and P.E. Tetlock (1999) “Accounting for the Effects of Accountability”, *Psychological Bulletin,* (125)2, pp. 255–275.

Li, J, and G. Ruhe (2008) “Analysis of Attribute Weighting Heuristics for Analogy-based Software Effort Estimation Method AQUA+”, *Empirical Software Engineering,* (13)1, pp. 63–96.

Li, Y.F., M. Xie, and T.N. Goh (2009) “A Study of Project Selection and Feature Weighting for Analogy Based Software Cost Estimation”, *Journal of Systems and Software,* (82)2, pp. 241–252.

Linberg, K.R. (1999) “Software Developer Perceptions About Software Project Failure: A Case Study”, *Journal of Systems and Software,* (49)2–3, pp. 177–192.

Magazinović, A., and J. Pernstål (2008) “Any Other Cost Estimation Inhibitors?” in ACM (ed.) *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement,* New York: ACM Press, pp. 233–242.

Matson, J.E., B E. Barrett, and J.M. Mellichamp (1994) “Software Development Cost Estimation Using Function Points”, *IEEE Transactions on Software Engineering*, (20)4, pp. 275–287.

McDonald, J. (2005) “The Impact of Project Planning Team Experience on Software Project Cost Estimates”, *Empirical Software Engineering,* (10)2, pp. 219–234.

McManus, J.J., and A.T. Wood-Harper (2003) *Information Systems Project Management: Methods, Tools and Techniques,* Harlow, UK: Prentice Hall.

Moløkken-Østvold, K., M. Jørgensen, S.S. Tanilkan, H. Gallis, A.C. Lien, and S.E. Hove (2004) “A Survey on Software Estimation in the Norwegian Industry” in IEEE Computer Society (ed.) *Proceedings of the Tenth International Symposium on Software Metrics,* New York: IEEE Computer Society Press, pp. 208–219.

Moløkken-Østvold, K., and K.M. Furulund (2007) “The Relationship Between Customer Collaboration and Software Project Overruns”, *Proceedings of AGILE 2007 Conference,* pp. 72–83.

Moløkken-Østvold, K., and M. Jørgensen (2005a) “A Comparison of Software Project Overruns—Flexible Versus Sequential Development Models”, *IEEE Transactions on Software Engineering,* (31)9, pp. 754–766.

Moløkken-Østvold, K., and M. Jørgensen (2005b) “Expert Estimation of Web-development Projects: Are Software Professionals in Technical Roles More Optimistic Than Those in Non-technical Roles?”, *Empirical Software Engineering,* (10)1, pp. 7–29.

Morgenshtern, O., T. Raz, and D. Dov (2007) “Factors Affecting Duration and Effort Estimation Errors in Software Development Projects”, *Information and Software Technology,* (49)8, pp. 827–837.

Mukhopadhyay, T., S.S. Vicinanza, and M.J. Prietula (1992) “Examining the Feasibility of a Case-based Reasoning Model for Software Effort Estimation”, *MIS Quarterly,* (16)2, pp. 155–171.

Mussweiler, T., and F. Strack (2001) “Considering the Impossible: Explaining the Effects of Implausible Anchors”, *Social Cognition,* (19)2, pp. 145–160.

Nunnally, J.C. (1978) *Psychometric Theory, second edition,* New York: McGraw-Hill.

Petersen, K., R. Feldt, S. Mujtaba, and M. Mattsson (2008) “Systematic Mapping Studies in Software Engineering” in Visaggio, G., M.T. Baldassarre, S. Linkman and M. Turner (eds.) *Twelfth International Conference on Evaluation and Assessment in Software Engineering,* Bari, Italy: University of Bari, pp. 1–10.

Prechelt, L., and B. Unger (2000) “An Experiment Measuring the Effects of Personal Software Process (PSP) Training”, *IEEE Transactions on Software Engineering*, (27)5, pp. 465–472.

Regopoulos, M. (1966) “The Principle of Causation as a Basis of Scientific Method”, *Management Science,* (12)8, pp. C135–C139.

Rombach, D., J. Münch, A. Ocampo, W.S. Humphrey, and D. Burton (2008) “Teaching Disciplined Software Development”, *Journal of Systems and Software,* (81)5, pp. 747–763.

Schutt, R.K. (2009) *Investigating the Social World, sixth edition,* Thousand Oaks, CA: Pine Forge Press.

Shepperd, M., and C. Schofield (1997) “Estimating Software Project Effort Using Analogies”, *IEEE Transactions on Software Engineering,* (23)11, pp. 736–743.

Subramanian, G.H., and S. Breslawski (1995) “An Empirical Analysis of Software Effort Estimate Alterations”, *Journal of Systems and Software,* (31)2, pp. 135–141.

Tversky, A., and D. Kahneman (1974) “Judgment Under Uncertainty: Heuristics and Biases”, *Science,* (185)1457, pp. 1124–1131.

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, pp. 582–590.

Webster, J., and R.T. Watson (2002) “Analyzing the Past to Prepare for the Future: Writing a Literature Review”, *MIS Quarterly,* (26)2, pp. xiii–xxiii.

Wesslén, A. (2000) “A Replicated Empirical Study of the Impact of the Methods in the PSP on Individual Engineers”, *Empirical Software Engineering,* (5)2, pp. 93–123.

Xia, W., D. Ho, and L.F. Capretz (2008) “A Neuro-fuzzy Model for Function Point Calibration”, *WSEAS Transactions on Information Science and Applications*, (5)1, pp. 22–30.

APPENDIX A: DISTRIBUTION OF PUBLICATION FORA

Table A1: Distribution of Publication Fora		
Publication Fora	Type	#
ACM SIGSOFT Software Engineering Notes	Journal	1
ACM-IEEE International Symposium on Empirical Software Engineering and Measurement	Conference	1
AGILE Conference	Conference	1
Empirical Software Engineering	Journal	4
IEEE Software	Journal	2
IEEE Transactions on Software Engineering	Journal	5
Information and Software Technology	Journal	4
Information Resources Management Association Conference	Conference	1
International Conference on Quality Software	Conference	1
International Journal of Project Management	Journal	1
International Software Metrics Symposium	Conference	2
Journal of Systems and Software	Journal	7
Management Science	Journal	1
MIS Quarterly	Journal	1
Total		32

APPENDIX B: OVERVIEW OF PRIMARY STUDIES

Table B1: Overview of Primary Studies with Regard to the Goal-Question Metric [According to Briand et al., 1996]						
No.	Study	Object of Study	Purpose	Quality Focus	Viewpoint	Context
1	Aranda and Easterbrook, 2005	Software development project	Characterization	Anchoring in estimation	18 graduate computer science students and 5 software professionals	Fictional software project
2	Connolly and Dean, 1997	Software writing task	Evaluation	Bias in decomposition and holistic estimates	45 junior and senior undergraduate management information systems students	Design and implementation course
3	Furulund and Moløkken-Østvold, 2007	Projects	Evaluation	Magnitude of effort overruns with regard to the use of experience data, estimation models and checklist	18 project managers	Norwegian software consultancy
4	Glass et al., 2008	Software projects	Characterization	Lying in the software profession	62 software professionals	Distributed across Europe, U.S. and Australia
5	Gray et al., 1999	System development	Characterization	estimation error	Not applicable	Healthcare System
6	Grimstad and Jørgensen, 2007	Software development effort estimates	Characterization	Inconsistency of estimates	7 software professionals	Various software development tasks
7	Grimstad et al., 2005	Projects	Characterization	Clients impact on effort estimation	300 Norwegian speaking software professionals at JavaZone conference 2004	Not applicable

Table B1: Overview of Primary Studies with Regard to the Goal-Question Metric – Continued [According to Briand et al., 1996]						
8	Jørgensen, 2004b	Software development effort estimation accuracy	Prediction	Bias due to task characteristics	Project leader and developers	Norwegian Web-development company
9	Jørgensen, 2004c	Expert estimation of software development effort	Control and change	Estimation strategy (top-down, bottom-up)	7 estimation teams	Norwegian branch of an international IT consulting company
10	Jørgensen, 2010	Software effort estimates	Characterization	Risk, over-optimism, overconfidence	Software professionals and managers	Various software development tasks
11	Jørgensen et al., 2007	Effort estimate predictions	Characterization	Estimators' optimism	25 software engineering students and 60 software professionals	Programming tasks, software project
12	Jørgensen and Grimstad, 2008	Software effort estimates	Control and change	Bias due to irrelevant and misleading information	338 software professionals	Norwegian software development
13	Jørgensen and Gruschke, 2009	Effort estimation, uncertainty Assessment	Evaluation	Lessons-learned sessions	20 consultants and 83 software professionals	Norwegian software industry
14	Jørgensen and Halkjelsvik, 2010	Judgment-based effort estimates	Characterization	Request formats of judgment-based estimation	131 software professionals	Web-development company, Web-based reservation system
15	Jørgensen and Moløkken, 2003	Software development effort estimates	Characterization	Situational and Task characteristic, accuracy	Software estimators	Norwegian Web-development company
16	Jørgensen and Moløkken-Østvold, 2004	Software estimation effort error	Characterization	respondent role, information collection approach, data analysis method	8 Interviews and experience reports from 68 projects	Norwegian software development company
17	Jørgensen and Sjøberg, 2001	Software development estimates	Characterization	Pre-planning effort estimates	More than 38 under-graduate computer science students and 17 software professionals,	e-business development, telecom development, University of Oslo
18	Jørgensen and Sjøberg, 2004	Software development effort estimates	Characterization	Customer expectation	32 students and 12 software developers	Fictional software development
19	Lederer et al., 1990	Projects	Characterization, evaluation	Method for managing the estimating process	Information systems managers and staff members	A chemical manufacturer
20	Lederer and Prasad, 1995	Projects	Characterization	Potential causes of inaccurate estimates	Information systems manager and professionals	Various industries

Table B1: Overview of Primary Studies with Regard to the Goal-Question Metric – Continued [According to Briand et al., 1996]						
21	Lederer and Prasad, 2000	Software development effort estimates	Characterization	Causes of inaccuracy	112 information system managers and professionals	Software development in U.S. organizations
22	Magazinović and Pernstål, 2008	Cost estimation (error)	Characterization	Inhibitors	20 project managers	Software development in Swedish car industry
23	McDonald, 2005	Software project cost estimates	Characterization	Project planning (team-) experience	135 teams of 6 to 8 students	Workstation software
24	Moløkken-Østvold and Furulund, 2007	Software project overrun	Characterization	Customer collaboration	18 project managers	Norwegian Software consultancy
25	Moløkken-Østvold and Jørgensen, 2005a	Software project overruns	Evaluation	Flexible and sequential development models	42 professionals	18 Norwegian software companies
26	Moløkken-Østvold and Jørgensen, 2005b	Expert estimation	Evaluation	Optimism due to estimator's role (technical and non-technical roles)	20 professionals	Norwegian Web-development company
27	Morgenshtern et al., 2007	Duration and effort estimation errors	Characterization	Project uncertainty estimation development, estimation management, estimator's experience, estimation technique	24 project managers of 43 projects	Israeli governmental organization (IT division)
28	Prechelt and Unger, 2000	Software estimation	Evaluation	Personal software process training	40 computer science master students	Computer program Phoneword
29	Rombach et al., 2008	Personal software process	Evaluation	Effort estimation accuracy	3090 engineers	Not applicable
30	Subramanian and Breslawski, 1995	Accuracy of effort estimation	Characterization	Alterations	45 project managers	Northeastern U.S. software industry
31	van Genuchten, 1991	Delay in software development	Characterization	Various reasons	6 project leaders	Software development department
32	Wesslén, 2000	Individual engineers' performance	Evaluation	Personal software process	131 students	Personal software process course

APPENDIX C: QUALITY ASSESSMENT CHECKS

Table C1: Questions Used for Quality Assessment Checks [Adopted from Dybå and Dingsøyr, 2008a]

No.	Question
Q1	Is there a rationale for why the study was undertaken?
Q2	Is there an adequate description of the context (e.g., industry, laboratory setting, products used, etc.) in which the research was carried out?
Q3	Is there a justification and description for the research design?
Q4	Have the researchers explained how the study sample (participants or cases) was identified and selected, and what was the justification for such selection?
Q5	Is it clear how the data was collected (e.g. through interviews, forms, observation, tools, etc.)?
Q6	Does the study provide description and justification of the data analysis approaches?
Q7	Has “sufficient” data been presented to support the findings?
Q8	Is there a clear statement of the findings?
Q9	Did the researchers critically examine their own role, potential bias, and influence during the formulation of research questions, sample recruitment, data collection and analysis, and selection of data for presentation?
Q10	Do the authors discuss the credibility of their findings?
Q11	Are limitations of the study discussed explicitly?

APPENDIX D: DISTRIBUTION OF RESEARCH METHODS

Table D1: Timely Distribution of Research Methods

Research Method	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	Total
Survey		1				2					1				1	2		3	1			11
Case Study	1											1							1			3
History-based Evaluation										1					1							2
Real-life Evaluation														1	1	2		2	1			7
Experiment								1			2	1			4	2		2	1	1	2	16
Total	1	1				2		1		1	3	2		1	7	6		7	4	1	2	39

APPENDIX E: QUALITY ASSESSMENT CHECKS

Table E1: Overview of the Quality Assessment Checks on the Primary Studies

No.	Study	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Overall
1	Aranda and Easterbrook, 2005	1	1	0.5	0.5	1	0	0.5	1	0	1	0	6.5
2	Connolly and Dean, 1997	1	0.5	0.5	0.5	1	0.5	0.5	1	0	1	0.5	7
3	Furulund and Moløkken-Østvold, 2007	1	1	0	0.5	0.5	1	0.5	1	0	1	1	7.5
4	Glass et al., 2008	0.5	0.5	0	0.5	1	0	0.5	0.5	1	0.5	1	6
5	Gray et al., 1999	1	0	0	0	0.5	1	0.5	1	0	0	1	5
6	Grimstad and Jørgensen, 2007	1	0.5	1	1	1	1	0	1	0	1	1	8.5
7	Grimstad et al., 2005	1	1	1	1	1	0.5	1	1	0	1	1	9.5
8	Jørgensen, 2004b	1	0.5	1	0.5	1	1	0.5	1	0	1	0.5	8
9	Jørgensen, 2004c	1	1	1	0.5	1	1	0.5	1	0	0.5	1	8.5
10	Jørgensen, 2010	1	1	1	0.5	1	1	1	1	0	0.5	1	9
11	Jørgensen et al., 2007	1	0.5	0.5	0	1	1	0.5	1	0	0.5	1	7
12	Jørgensen and Grimstad, 2008	0.5	0.5	1	0	0.5	0.5	1	1	0	0	0	5
13	Jørgensen and Gruschke, 2009	1	1	1	1	1	1	0.5	1	0	0.5	1	9
14	Jørgensen and Halkjelsvik, 2010	1	1	1	0	1	0.5	1	1	0.5	0.5	1	8.5
15	Jørgensen and Moløkken, 2003	1	0.5	0.5	0	1	0.5	0.5	1	0	0	0	5
16	Jørgensen and Moløkken-Østvold, 2004	1	1	1	0.5	1	1	0.5	1	0	1	1	9
17	Jørgensen and Sjøberg, 2001	1	1	1	0.5	1	1	0.5	1	1	1	1	10
18	Jørgensen and Sjøberg, 2004	1	0.5	1	0.5	0.5	1	0	1	0	1	0	6.5
19	Lederer et al., 1990	1	1	1	0	1	0.5	0.5	1	0	0.5	0	6.5
20	Lederer and Prasad, 1995	1	1	1	0.5	1	1	1	1	0	0	0	7.5
21	Lederer and Prasad, 2000	1	0.5	0.5	0.5	1	1	0.5	1	0	0	0	6
22	Magazinović and Pernstål, 2008	1	1	1	0.5	1	0.5	0.5	1	0	1	0	7.5
23	McDonald, 2005	1	1	1	0.5	1	1	1	1	0	1	1	9.5

Table E1: Overview of the Quality Assessment Checks on the Primary Studies – Continued													
24	Moløkken-Østvold and Furulund, 2007	0.5	0.5	1	0.5	1	1	0.5	1	0	1	0.5	7.5
25	Moløkken-Østvold and Jørgensen, 2005b	1	1	1	0.5	1	1	0.5	1	0	1	1	9
26	Moløkken-Østvold and Jørgensen, 2005a	1	1	0.5	1	1	1	0.5	1	0	1	1	9
27	Morgenshtern et al., 2007	1	1	0.5	0	1	1	0.5	1	0	0.5	0.5	7
28	Prechelt and Unger, 2000	1	1	1	0	0.5	1	0.5	1	0	1	1	8
29	Rombach et al., 2008	1	0.5	0.5	0.5	1	1	1	0.5	0	1	1	8
30	Subramanian and Breslawski, 1995	1	0.5	0.5	0.5	1	0.5	0	1	0	0	0	5
31	van Genuchten, 1991	0.5	0.5	0.5	0	1	1	0.5	0.5	0	0	0	4.5
32	Wesslén, 2000	0.5	1	1	0.5	0.5	1	1	1	0	1	1	8.5

APPENDIX F: ACCURACY FACTORS AND COST DRIVERS

Table F1: Comparison of Accuracy Factors (This Study) and Cost Drivers in CoCoMo II [Boehm, 2000]	
Accuracy Factors	Cost Drivers
Task complexity	Database size, product complexity, multisite development
Changes in personnel	Personnel continuity
Newness of technology	Platform volatility
Requirements specification, overlooked tasks, Neutrality and relevance of information, diligence	Analyst capability, application experience, language and tools experience, platform experience
Assessment of developer skills/assessment of task complexity	Programmer capability

ABOUT THE AUTHORS

Dirk Basten is a post-doctoral researcher in the Department of Information Systems and Systems Development at the University of Cologne, Germany. His research focuses on IS project success, software development effort estimation, knowledge management, and organizational agility. He has published in journals such as *Journal of Computer Information Systems*, *International Journal of Information Technology Project Management*, and *IEEE Computer*.

Ali Sunyaev is an assistant professor in the Department of Information Systems at the University of Cologne, Germany. His research interests include design and management of information systems, health IT, and information systems security. His research has been published in *Communications of the ACM*, *ACM Journal of Data and Information Quality*, *IEEE Software*, *IEEE Computer*, *International Journal of Medical Informatics*, *Business and Information Systems Engineering*, *Communications of the Association for Information Systems*, and other journals. Dr. Sunyaev is also affiliated at the Intelligent Health Laboratory, Harvard-MIT Division of Health Sciences and Technology, Harvard University, U.S.

Communications of the Association for Information Systems

ISSN: 1529-3181