# A Comparison of Software Project Overruns—Flexible versus Sequential Development Models

Kjetil Moløkken-Østvold, *Member*, *IEEE*, and Magne Jørgensen

**Abstract**—Flexible software development models, e.g., evolutionary and incremental models, have become increasingly popular. Advocates claim that among the benefits of using these models is reduced overruns, which is one of the main challenges of software project management. This paper describes an in-depth survey of software development projects. The results support the claim that projects which employ a flexible development model experience less effort overruns than do those which employ a sequential model. The reason for the difference is not obvious. We found, for example, no variation in project size, estimation process, or delivered proportion of planned functionality between projects applying different types of development model. When the managers were asked to provide reasons for software overruns and/or estimation accuracy, the largest difference was that more of flexible projects than sequential projects cited good requirement specifications and good collaboration/communication with clients as contributing to accurate estimates.

**Index Terms**—Cost estimation, management, project control and modeling, software development models.

## 1 Introduction

Software projects are infamous for exceeding their original estimates [1]. A recent review of surveys on estimation performance reports that 60-80 percent of all software projects encounter effort overruns [2]. The average effort overrun appears to be 30-40 percent. Similar findings are reported for schedule overruns, with 65-80 percent of all projects facing overruns of the delivery date [2].

Effort overruns usually lead to cost overruns when external contractors are used as work effort is the main expense in most software projects. Depending on the type of contract, these costs have to be paid by the customer and/or written off as losses by the contractors. Schedule overruns may lead to fines for the contractors for late delivery. For the customers, schedule overruns may result in problems such as lack of productivity or loss of business. For internal projects, the implications are somewhat different, but equally problematic.

Most research initiatives directed at improving estimation accuracy have focused on the development of formal estimation models. Formal estimation models have been based on a variety of measures of development size, such as lines of code [3] and function-points [4]. There is, however, no conclusive evidence to show that the employment of formal estimation models results in improved predictive estimation accuracy [5]. Expert estimation remains the preferred approach for most software professionals [2], [5].

Another line of research has focused on improving the expert estimation process by introducing a variety of supporting tools and processes, including work breakdown structures (WBS) [6], [7], checklists [8], [9], experience databases [10], and group-based estimates [9], [11].

The observations of project overruns in software engineering are similar to results obtained in other areas of research. For example, in the transportation infrastructure sector, it has been reported that 86 percent of projects face cost overruns and that the average magnitude of the overruns is 28 percent. These overruns appear to be of the same magnitude, independent of location and the era of the projects [12]. Further, the estimation accuracy seems to be similar for the various estimation approaches [12]. A consequence of these findings is that researchers have begun to explore factors other than the estimation process when trying to explain and avoid overruns [13].

In software engineering research, much less attention has been paid to ways in which improved project management may reduce overruns. One suggestion for reducing overruns is through improved development models. The use of incremental and evolutionary development models, for example, have, since 1976, been said to facilitate more accurate estimates and reduce overruns, when compared to sequential development [14], [15], [16], [17].

In order to assess the claims stated by proponents of flexible development models, we investigated the use of development models in a survey on effort estimation in Norwegian software projects. We also investigated other properties of the projects, the estimation process in particular, which may explain any observed differences. Previous surveys on software estimation [2] have not addressed this topic. For some of the surveys, this is because they were conducted before the use of flexible development models became widespread [18].

- *The authors are with the Simula Research Laboratory, PO Box 134, 1325 Lysaker, Norway. E-mail: {kjetilmo, magnej}@simula.no.*

*Manuscript received 3 Sept. 2004; revised 7 Apr. 2005; accepted 27 June 2005; published online 15 Sept. 2005.*
*Recommended for acceptance by A. Finkelstein.*
*For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number TSE-0183-0904.*

The remainder of this paper is organized as follows: Section 2 provides a background on the claims regarding choice of development models and on earlier empirical studies on project overruns and choice of development models. Section 3 states our research questions. Section 4 provides a brief description of differences between the most common development models. Section 5 describes our research method. Section 6 reports the results of our survey. A discussion of the results is provided in Section 7. Section 8 presents the conclusion of this paper.

An earlier version of this paper was presented at PROFES 2004 [19]. This version is substantially expanded, in terms of both the number of analyzed projects and depth of analysis. It also has a different focus and explores other aspects of the projects not directly related to the estimation process. There is also a thorough analysis of the postmortem experiences from all projects included in this paper.

## 2 Background

The use of terms to describe development models may frequently be confusing. It is, for example, not easy to separate iterative, evolutionary, agile, and incremental development models. In this paper, we apply the term "sequential" to denote development models similar to the waterfall model and the term "flexible" to denote all types of nonsequential development models, e.g., iterative and incremental development models. The term "flexible" to describe the broad class of nonsequential development models has previously been used by, for example, Iansiti and MacCormack [20] to reduce problems of classification and definition.

Advocates of flexible development models offer different explanations as to why such models should reduce overruns. The incremental development model is, for example, claimed to lead to better managed projects [3], [15], [21], [22]. This has been explained as an effect of the feedback from the experiences of the first increment(s) [15].

If a software project in which an incremental development model is applied suffers from overruns, later increments can also be eliminated or reduced, and the most useful parts of the system will still be produced within the original budget. There may, however, be management and control problems in connection with incremental development [15] and, hence, the overall possible benefits of the use of the incremental development model are not obvious.

Reduction in overruns has also been attributed to evolutionary development. Tom Gilb, for example, claims that "Evolutionary delivery, and the feedback it produces, often leads to management acceptance of revised estimates for time and money at early stages of the project" [23].

As stated in a comprehensive review of the history of iterative and incremental development, flexible development models have been in existence for some time [18]. In fact, they were first formulated as early as the 1930s. Still, such models were not adopted by practitioners, textbook authors, and government bodies until recently [18].

When seeking to reduce project overruns, there are two ways to approach the problem. First, you can increase the accuracy of your estimates through a better estimation process and, second, you can increase your project control. There are several factors in a software project which may make accurate estimation difficult and increase the chances of overruns. Such factors often include lack of well-defined scope, unstable requirements, quality of management, and skill of the developers. Depending on who and how you ask, such factors will receive a varying degree of attention when overruns are explained [24]. In addition, it is often impossible for the manager of a given project to select the skill of his staff or completely control scope creep or requirement changes submitted by the customers. Another important factor is that project overruns may be impacted by project priorities. If avoiding cost overruns have high priority and the requirement specification is vague, it is more likely that cost overruns are avoided [25].

In this paper, we will look at many possible differences in the estimation process between flexible and sequential projects, as well as investigate aspects related to the managers' experience of project performance, management, and control.

Interestingly, from a research perspective, the claimed benefits related to increased estimation accuracy and reduced overruns in flexible projects have not been empirically investigated except in company-specific case studies. We were unable to find surveys or experiments that examined possible connections between particular development models and estimation accuracy and software overruns. Most case studies have cited positive experiences, e.g., IBM [26], NASA [27], and the North American Aerospace Defense Command/Air Force Space Command [28]. However, there have also been less positive experiences reported [29].

## 3 Research Questions

The study reported contributes with more evidence-based discussion about the claimed benefits of selecting a particular development model. The research questions are as follows:

**RQ 1:** *Are there significant differences in the occurrence of effort overruns between projects following flexible and sequential development models, respectively?*

**RQ 2:** *Are there significant differences in the occurrence of schedule overruns between projects following flexible and sequential development models, respectively?*

In order to investigate the claims that flexible development models will reduce overruns, we compare effort and schedule estimation accuracy and bias (defined as the *direction* of the inaccuracy, either a bias toward optimistic or pessimistic estimates) in projects based on flexible and sequential development models. The differentiation between sequential and flexible development models may appear coarse grained. However, the grouping is sensible from several points of view, elaborated in the following section.

## 4 Software Development Models

It is outside the scope of this paper to provide a more complete description of development models used in the software industry. This is especially true given that the different models are combined in different ways and often tailored for specific purposes, projects, or companies. There exist several variants of the different sequential and flexible development models. An example of this is how Royce's description of what would become the waterfall model was (mis)interpreted and used in a fashion stricter than that which he originally suggested [18].

We will, for the support of the classification process described in Section 5, provide a brief description of the main properties of the most commonly employed development models used by the software companies observed in our study. The main development models observed were the following:

- waterfall models,
- incremental models,
- evolutionary models, and
- agile models.

The traditional waterfall model is a sequential model [30]. It separates system development into distinct phases that are supposed to be completed in sequence, i.e., one phase should not be started before the preceding phase is completed. The phases typically are analysis, design, programming, and testing [31]. Depending on the type of system, one might also need to consider integration with other systems. Waterfall-based development models are widely used. Possible reasons for this are that a sequential development model is

1. easy to explain and recall,
2. gives the impression of an orderly, accountable, and measurable process,
3. has been the standard development procedure in many communities, and
4. has, until recently, been enforced by government regulations [18].

Opponents of such models, however, state that they only work well when technology, product features, and competitive conditions are predictable or evolve slowly [20]. Such preconditions are, however, rarely present in software projects.

Incremental development is based on a division of the systems into parts (increments). The increments of the system are developed in sequence or in parallel [15]. Analysis, design, programming, and testing are performed for each increment. Each of the increments provides a subset of the product's functionality. To minimize risks, the most critical part of the system may be delivered first.

The introduction of evolutionary project management in software engineering has been attributed to Gilb [18]. In evolutionary development, the system is developed cyclically, with system deliveries in versions [23]. This is in contrast to the "big bang" delivery provided in the traditional waterfall development model. The delivered versions are adjusted according to client response and delivered again for a new assessment. Gilb states that "*You have the opportunity of receiving some feedback from the real world before throwing in all resources intended for a system, and you can correct possible design errors...*" [32].

Abrahamson et al. define an agile development model as having the following properties [33]: incremental (small software releases with rapid cycles), cooperative (client and developers working constantly together with close communication), straightforward (the model itself is easy to learn and to modify and is well-documented), and adaptive (last minute changes can be made). Light and agile development addresses only the functions most needed by the client (the functions that deliver the most business value to the client). These functions are then delivered as quickly as possible and feedback collected, which will then be used to prompt further development.

As indicated in the description of the development models, a project may use more than one development model, e.g., a project may use a combination of incremental and evolutionary development. In addition, it may not be easy to decide whether a project follows an agile or incremental development model. These factors motivate our decision to separate the development models into two categories only: sequential and flexible. Sequential development models include waterfall-based development models, while flexible development models include all development models based on increments and evolution. This broad categorization is sensible for several reasons:

- Both in textbooks and the industry, development methods are often combined and tailored. This is especially true for incremental, evolutionary, and agile methods. Many textbooks and companies have their own definitions, and descriptions of evolutionary/iterative methods are frequent. On the other hand, there is most often a clear distinction between these models and sequential (waterfall) models.
- From a research perspective, the differentiation between flexible and sequential methods has been used in several substantial papers [18], [20]. It should therefore be familiar to researchers.
- From a practical perspective, there is also a clear distinction between flexible and sequential development methods. Flexible development methods rely more on interaction with customers and repetition of the different phases and less on large amounts of upfront specification.

## 5 Survey Method

The survey was conducted between February and November 2003. The original intent was to compare estimation practices and performance in the Norwegian software industry with findings from other countries.

### 5.1 The Participating Companies

In order to ensure a representative sample, stratified random sampling [34] from the population of Norwegian software development companies was used. This is a reasonable approach since we were going to investigate a limited number of companies. It was necessary to ensure that we had companies that represented different types of organizations, such as software houses developing products for the mass market, contractors who develop for clients,

TABLE 1
Overview of Participating Companies

| *Comp. ID* | *# Employees* | *Main client type* | *Project type* | | | *Estimation method* |
|---|---|---|---|---|---|---|
| | | | *New* | *Maintenance* | *Combination* | |
| 1 | 70 | Private/public | 65 | 35 | 0 | Expert/combination |
| 2 | 28 | Private/public | 40 | 40 | 20 | Expert |
| 3 | 94 | Private/public | 80 | 20 | 0 | Expert |
| 4 | 750 | Internal | 100 | 0 | 0 | Expert |
| 5 | 200 | Mass-market | 30 | 70 | 0 | Expert |
| 6 | 137 | Private/public | 50 | 20 | 30 | Expert |
| 7 | 504 | Private | 30 | 70 | 0 | Expert |
| 8 | 120 | Private | 70 | 30 | 0 | Expert |
| 9 | 65 | Mass-market | 80 | 20 | 0 | Expert |
| 10 | 50 | Private/public | 30 | 40 | 30 | Expert |
| 11 | 30 | Private/public | 64 | 36 | 0 | Expert/combination |
| 12 | 180 | Internal | 18 | 83 | 0 | Expert |
| 13 | 100 | Internal | 40 | 60 | 0 | Expert |
| 14 | 50 | Internal | 30 | 70 | 0 | Expert |
| 15 | 10 | Private/internal | 70 | 30 | 0 | Combination |
| 16 | 49 | Public | 66 | 34 | 0 | Combination |
| 17 | 80 | Public | 65 | 35 | 0 | Combination |
| 18 | 15 | Private | 100 | 0 | 0 | Expert |

and the internal development departments of large companies. We also wanted companies of different sizes, both small (< 25 employees), medium (25 to 100 employees), and large (> 100 employees). The classification was based on different Norwegian sources, e.g., business magazines [35]. It is impossible to get a completely random sample of all Norwegian software-developing companies as some companies are too new to be listed, others have recently merged, some have been bought by foreign companies, or some may be categorized under disciplines other than software. However, our main priority was to obtain a sufficiently representative, diverse, and nonbiased sample, which we took great care in accomplishing.

Each company was contacted by phone and the main outlines of the study were presented to them. This included informing them about topic (software estimation and related aspects), research procedure (interviews), and possible incentives for participation (free estimation course). A total of 37 companies were contacted. For some of the companies, we were not able to contact the appropriate person in charge. In total, 18 companies agreed to participate, indicating a response rate of 48.6 percent. All who agreed to participate were included in the study. The most frequent reason stated by those who declined to participate was that they did not have resources available for interviews. Only two companies declined because they were not interested in the research.

For those who agreed to participate, they were given time to prepare before they were visited by our researchers. We sent a brief sketch of the study to each company, with information about who we wanted to interview and what kind of data they had to retrieve in advance. They were not informed about any of the specific questions or hypotheses.

The unit of investigation was either the entire company or a specific department in cases where the company had more than 1,000 employees. We will use the term company for our unit of research in this paper. An overview is presented in Table 1.

The 18 companies (departments) that participated had between 10 and 750 employees, with an average of 141. Four of the companies developed projects to be used in-house, while two developed products for sale to the mass market. The rest had private and/or public customers.

Nearly all of the companies were involved in new and maintenance projects. The estimation method was, to a large extent, expert-based, although some of the companies combined expert estimation and use case-based models.

Each company submitted from one to four of their projects for scrutiny. The criteria that the projects needed to meet in order to be included in the study were that they should be over 100 hours (to exclude trivial tasks), be closed (either completed or abandoned), be the most recent cases (to avoid biased selection), and that we had access to the managers of the projects. This resulted in a repository of 52 project interviews.

### 5.2 Data Collection and Analysis

We collected data via personal semistructured interviews, based on a predefined set of questions and interview instructions. Each interview lasted between 30 and 70 minutes, and all of them were taped. This approach yields data of high quality and helps to resolve ambiguities [34]. This was especially important in our survey since there may be variations in the use and interpretation of terms related to development models and estimation approaches. It also allowed the respondents to add valuable information that did not fit into a predefined questionnaire. Another point in favor of this approach is that face-to-face interviews may increase the likelihood of serious contributions from the companies. The main limitation of the approach is that it is time-consuming and, hence, prevents us from investigating as many companies and projects as would be possible by using questionnaires sent by mail.

The interviews at company level were mainly concerned with background information on such general matters as number of employees, business segment, types of client, general aspects of development models, estimation approaches, and process improvement efforts. At the project level, we collected detailed information about each project. This information included the type of project, type of client, estimation approach, development process, and people involved. Most important, however, was the collection of estimate(s) and actual(s) for effort and schedule. This was always based on recorded data from the participants so that these results were not influenced by hindsight bias.

The managers defined their development model as being sequential (waterfall) or flexible (evolutionary, incremental, and agile) based on predefined categories. They could also specify the extent to which component-based development and prototyping was used and the extent to which combined models, e.g., evolutionary and incremental development, were used. In cases where the participants reported that they followed a company-defined development model, we asked them to provide descriptions of the process. All projects were also asked to describe their process (free-text response) to ensure correct categorizations.

They also provided a free-text description of their estimation process. The terminology used in the context of software estimation is often ambiguous [36]. Different respondents may, for example, interpret estimated most likely effort differently. We were aware of such problems of interpretation and tried to ensure common interpretations of important terminology used in the interviews. We asked the managers to provide all available estimates, both those used internally at different stages and those relayed to the customer. Often, these differed, as the latter are affected by price-to-win. All estimates and actual outcomes were based on documented data from the companies and not the managers' memory.

The written interview answers and responses on tapes were registered into an Access database by two independent researchers who had no stake in the research. Their job was to ensure that the responses written down by the interviewers were in line with what the subjects had answered (on tape). Both analysts checked the entire Access database with the notes and the taped recordings. This was a straightforward task, and no errors were discovered. The most important task for the independent analysts was to ensure that the development method was correctly classified by the interviewers. For most cases, this was also straightforward, but, when projects used a company-specific development approach, errors could be introduced. The use of two independent analysts on this task was especially important because it ensured that the development models and estimation approaches were correctly classified. There was no indication that the researchers had disagreed on the classifications for any of the projects, thus indicating a high level of interrater reliability. The data from the Access database was subsequently exported to Excel and Minitab, where calculations and statistical analyses were performed.

In all interviews, the project managers were also asked to provide free-text responses to explain the project outcomes, related to estimates, actual effort, and delivered functionality. The respondents described as many positive and negative aspects as they felt necessary. These responses were translated and grouped into broader categories by the two authors, independent of each other. There were only minor discrepancies (a classification deviation of one reason in any given project) between the two researchers concerning eight of the projects. This ambiguity was resolved through a second round of classification where both researchers agreed on the classifications.

### 5.3 Measuring Estimation Accuracy

There exist several different ways of calculating estimation accuracy. The most common is the MRE (Magnitude of Relative Error) measure [37], which is calculated as:

$$MRE = \frac{|x-y|}{x}, \quad \text{x = actual and y = estimated.} \qquad (1)$$

Even though the MRE is the most widely used measure of estimation accuracy [38], one must be aware that it has unfortunate properties. The main concern is the fact that underestimated and overestimated projects are weighted unevenly, with underestimation not weighted sufficiently [39]. In addition, it does not seem sensible from a practitioner's point of view since estimation performance in the software industry is often rated relative to the estimated effort [40], [41], [42], [43].

The BRE (Balanced Relative Error) is, as its name indicates, a more balanced measure [44] than MRE. It is calculated as:

$$BRE = \frac{|x-y|}{\min(x,y)}, \quad \text{x = actual and y = estimate.} \qquad (2)$$

Assume, for example, that two projects estimate the required effort to be 1,000 work-hours. Project A spends 500 work-hours, while project B spends 2,000 work-hours. The MRE of Project A is 1.00, while the MRE of Project B is only 0.50. The BRE evenly balances overestimation and underestimation, leading to a result of 1.00 for both projects. It is possible for the choice of accuracy measure to have a huge impact on results, as has been shown in other studies [39], [43]. The same could well be true of the study in this paper. The continuous use of MRE instead of other measures is probably due to lack of awareness of the shortcomings of MRE.

For our own survey, it was important to be able to choose the most appropriate evaluation criteria, as well as being able to compare with previous studies and make our findings accessible for practitioners. We decided to use measures based on BRE since this places an even emphasis on overestimation and underestimation. It is robust and sensible, from both theoretical and practical points of view. However, we also choose to include measures based on

MRE in order to accommodate readers more accustomed to that measure.

In addition, whether using BRE, MRE, or some other measure, there are further factors to address. Are we interested only in the accuracy of the estimates or in the direction of the effect or both? Absolute values, whether BRE or MRE, do not capture estimation bias because they are not concerned with the direction of the estimation inaccuracy. In a review of previous surveys [2], we observed that the surveys relied mainly on measures that are able to reveal the direction of the effect, e.g., not MRE or BRE.

MREbias and BREbias measure both the size and the direction of the estimation error, i.e., whether there is a bias toward optimism or pessimism:

$$MREbias = \frac{(x-y)}{x}, \quad \text{x = actual and y = estimate.} \quad (3)$$

$$BREbias = \frac{(x-y)}{\min(x,y)}, \quad \text{x = actual and y = estimate.} \quad (4)$$

In addition, for a set of observations, we will probably have differences if we compare the mean and the median of the observations, e.g., MMRE and MdMRE, as illustrated by Foss et al. [39]. When aggregating the results, we decided to report both the mean and median results, for MRE, BRE, MREbias, and BREbias. The statistical analyses are based on either the mean or medians based on properties of the samples, elaborated in the next section.

### 5.4 The Problem of Multiple Estimates

Previous surveys on software estimation have tended to treat a software estimate as a single fixed value. During the course of our research, however, we have noticed that software projects often have several effort estimates. This property of software projects and how it poses challenges to estimation models has also been addressed by Edwards and Moores [45].

Part of the problem is that the effort estimate often changes over the course of a project, depending on the *stage* at which the estimate is made. For example, a project might have an early estimate, based on vague requirements, a planning estimate based on a detailed requirement specification, and one (or more) reestimates during the course of development. Another factor contributing to the problem is the different receivers of the effort estimate. A project may, for example, have two estimates at the planning stage: one that is used internally in the project team and another communicated to the client.

Projects can consequently operate with one, two, or even more different effort estimates. In our survey, we found that a project could have as many as six different effort estimates. This poses a significant challenge to estimation surveys. In this paper, the goal is to investigate the estimation ability of professionals in software projects in relation to the development model used. For this reason, we found it meaningful to compare the *most likely* estimates at the *planning stage*, i.e., the estimate used internally by the developers at the stage where the decision to start implementation was made. This is in accordance with research on estimation accuracy in other areas [12].

### 5.5 Measuring Estimation Accuracy in Flexible Development Projects

There are a number of hazards related to comparing the estimation accuracy of sequential and flexible projects. In some guidelines for flexible models, it is stated that one should set a deadline for effort and schedule and then simply produce whatever output is possible within those boundaries, with the most important functionality first [23]. Obviously, with this approach, one will end up on target each time. It would be meaningless to do a comparison if we were just comparing the flexibility of flexible development models to the accuracy of sequential development models. However, in real-life projects, especially when one develops for a client (bespoke system development), stakeholders often expect some kind of predefined and agreed-upon functionality to be delivered. In all projects in this survey, regardless of the development model, there existed schedule and effort estimates for the project total, along with the functionality expected, when the decision to start the project was made. These estimates were always important. For contractors with external projects, it was the basis for their bids and contract negotiations, whereas internal development projects used them for staffing, planning, and prioritization.

Another problem that is related to the estimation accuracy of projects with a flexible development model is that many of them encourage estimation revisions throughout the project. An estimate given closer to the end of a project will probably be more accurate than one provided in the beginning. As discussed in the last section, this does not raise any problems for this survey since we based the estimation accuracy calculations on the *most likely* estimates at the *planning* stage. This was done for both flexible and sequential projects.

### 5.6 Measuring Delivered Functionality

In order to investigate whether there were variations between the sequential and flexible projects regarding the completeness of delivered functionality, we interviewed the project managers about the delivered functionality of the finished products. They were asked about the extent to which the delivered functionality met the original specification on which the most likely estimates at the planning stage were based. Since we did not have access to the opinions of the users of the products, there is a potential problem regarding the objectivity of the answers. In addition, there may be a hindsight bias. However, we have no reason to believe that such factors would differ from one survey group to another.

## 6 Results

Out of the 52 project interviews, we excluded eight projects, either because they lacked information or because most of the estimation and implementation work had been conducted by external subcontractors. Two of the projects were abandoned before completion. This left 42 projects for the analysis. A complete record of the most relevant data is presented in Table 2. Nineteen projects were classified as using a flexible model. The other 23 projects followed a sequential development model. Effort is measured in work-hours, and schedule

TABLE 2
Survey Data

| # | *Model* | *Effort actual* | *Estimate* | *BREbias* | *Schedule actual* | *Estimate* | *BREbias* | *Functionality* |
|---|---|---|---|---|---|---|---|---|
| | | *(person-hours)* | *(person-hours)* | | *(calendar days)* | *(calendar days)* | | |
| 1 | Seq. | 180 | 90 | 1.00 | 56 | 21 | 1.67 | 100% |
| 2 | Seq. | 195.5 | 145 | 0.35 | 84 | 79 | 0.06 | 100% |
| 3 | Seq. | 261 | 133.5 | 0.96 | 334 | 273 | 0.22 | 100% |
| 4 | Seq. | 432 | 400 | 0.08 | 56 | 42 | 0.33 | 110% |
| 5 | Seq. | 562.5 | 487.5 | 0.15 | 106 | 103 | 0.03 | 110% |
| 6 | Seq. | 600 | 300 | 1.00 | 156 | 156 | 0.00 | 100% |
| 7 | Seq. | 696 | 650 | 0.07 | 49 | 49 | 0.00 | 100% |
| 8 | Seq. | 866.5 | 340 | 1.55 | 91 | 72 | 0.26 | 125% |
| 9 | Seq. | 955 | 1410 | -0.48 | 64 | 74 | -0.16 | 98% |
| 10 | Seq. | 1000 | 560 | 0.79 | 140 | 84 | 0.67 | 110% |
| 11 | Seq. | 1085 | 640 | 0.70 | 84 | 54 | 0.56 | 90% |
| 12 | Seq. | 1150 | 1077 | 0.07 | 49 | 42 | 0.17 | 110% |
| 13 | Seq. | 1200 | 750 | 0.60 | 457 | 117 | 2.91 | 100% |
| 14 | Seq. | 1400 | 700 | 1.00 | 224 | 182 | 0.23 | 100% |
| 15 | Seq. | 1903 | 914 | 1.08 | 217 | 196 | 0.11 | 120% |
| 16 | Seq. | 3140 | 1982 | 0.58 | 153 | 153 | 0.00 | 145% |
| 17 | Seq. | 3831 | 2170 | 0.77 | 54 | 54 | 0.00 | 100% |
| 18 | Seq. | 4012.5 | 3937.5 | 0.02 | 152 | 138 | 0.10 | 95% |
| 19 | Seq. | 5170 | 4227 | 0.22 | 98 | 98 | 0.00 | 110% |
| 20 | Seq. | 8063 | 7520 | 0.07 | 395 | 395 | 0.00 | 99% |
| 21 | Seq. | 8910 | 5450 | 0.63 | 304 | 212 | 0.43 | 110% |
| 22 | Seq. | 9000 | 4000 | 1.25 | 335 | 293 | 0.14 | 110% |
| 23 | Seq. | 14000 | 12000 | 0.17 | 640 | 619 | 0.03 | 95% |
| 24 | Flex. | 101 | 190 | -0.88 | 85 | 60 | 0.42 | 100% |
| 25 | Flex. | 319 | 330 | -0.03 | 235 | 235 | 0.00 | 100% |
| 26 | Flex. | 342 | 292 | 0.17 | 113 | 113 | 0.00 | 105% |
| 27 | Flex. | 466.5 | 533.5 | -0.14 | 104 | 97 | 0.07 | 100% |
| 28 | Flex. | 506 | 506 | 0.00 | 70 | 70 | 0.00 | 100% |
| 29 | Flex. | 593.5 | 593.5 | 0.00 | 152 | 152 | 0.00 | 100% |
| 30 | Flex. | 907 | 570 | 0.59 | 116 | 109 | 0.06 | 112% |
| 31 | Flex. | 1000 | 1125 | -0.13 | 106 | 106 | 0.00 | 100% |
| 32 | Flex. | 1000 | 705 | 0.42 | 70 | 60 | 0.17 | 105% |
| 33 | Flex. | 1242 | 1249 | -0.01 | 214 | 153 | 0.40 | 98% |
| 34 | Flex. | 1242 | 1249 | -0.01 | 106 | 92 | 0.15 | 100% |
| 35 | Flex. | 1335 | 1030 | 0.30 | 122 | 122 | 0.00 | 115% |
| 36 | Flex. | 1512 | 1500 | 0.01 | 70 | 70 | 0.00 | 100% |
| 37 | Flex. | 2732 | 2265 | 0.21 | 245 | 210 | 0.17 | 120% |
| 38 | Flex. | 3454 | 2340 | 0.48 | 245 | 140 | 0.75 | 150% |
| 39 | Flex. | 3746 | 3784 | -0.01 | 266 | 266 | 0.00 | 100% |
| 40 | Flex. | 5631 | 1932 | 1.91 | 281 | 220 | 0.28 | 120% |
| 41 | Flex. | 7844 | 3086 | 1.54 | 183 | 183 | 0.00 | 100% |
| 42 | Flex. | 28645 | 27241 | 0.05 | 336 | 296 | 0.14 | 90% |
| 43 | Aborted | n/a | 6728 | n/a | n/a | 151 | n/a | n/a |
| 44 | Aborted | n/a | 2720 | n/a | n/a | 152 | n/a | n/a |

in calendar days. For the projects, the mean actual effort was 3,125 work-hours, while the median was 1,175 work-hours. The mean schedule was 177 calendar days, while the median was 131 calendar days.

The overall mean effort BREbias was 0.41. This corresponds to an average effort overrun of 41 percent. A negative BREbias means that the estimate was too high, while zero BREbias means that the estimate was on target, and a positive BREbias means that the estimate was too low.

A short summary of the estimation results of projects with different development models is presented in Table 3. As seen in the table, the results are consistent independent of which of the four-evaluation criterion (BRE, BREbias, MRE, and MREbias) that is used. The

TABLE 3
Estimation Results by Development Model

| | | *Mean* | | *Median* | |
|---|---|---|---|---|---|
| | | **Flexible (n=19)** | **Sequential (n=23)** | **Flexible (n=19)** | **Sequential (n=23)** |
| **Effort** | **Estimate (person-hours)** | 2659 | 2169 | 1125 | 750 |
| | **Actual (person-hours)** | 3296 | 2983 | 1242 | 1150 |
| | **Accuracy (BRE)** | 0.36 | 0.59 | 0.14 | 0.60 |
| | **Bias (BREbias)** | 0.24 | 0.55 | 0.01 | 0.60 |
| | **Accuracy (MRE)** | 0.21 | 0.33 | 0.14 | 0.39 |
| | **Bias (MREbias)** | 0.09 | 0.29 | 0.01 | 0.38 |
| **Schedule** | **Estimate (days)** | 145 | 152 | 122 | 103 |
| | **Actual (days)** | 164 | 187 | 122 | 140 |
| | **Accuracy (BRE)** | 0.14 | 0.35 | 0.06 | 0.14 |
| | **Bias (BREbias)** | 0.14 | 0.34 | 0.06 | 0.11 |
| | **Accuracy (MRE)** | 0.10 | 0.17 | 0.06 | 0.13 |
| | **Bias (MREbias)** | 0.10 | 0.15 | 0.06 | 0.10 |
| **Functionality** | **Delivered (%)** | 106 | 106 | 100 | 100 |

flexible projects receive a lower score than the sequential projects (indicating less bias/more accuracy). Since most of the projects had effort/schedule overruns (as seen in Table 2), the BRE/BREbias scores are generally higher than the MRE/MREbias scores.

A visual inspection and an Anderson-Darling test [46] on normality revealed that none of the samples were normally distributed for the measures presented in Table 3. We therefore applied the more robust nonparametric statistical Kruskal-Wallis test [47] of difference in median accuracy on BRE, MRE, BREbias, and MREbias. In addition, in order to measure the magnitude of any observed difference, we included Cohen's size of effect measure (d) [48]. The size of effect (d) is calculated as: d = (mean value sequential group – mean value flexible group)/pooled standard deviation among the groups. For example, the median effort BRE was 0.14 for the flexible group and 0.60 for the sequential group. The Kruskal-Wallis test on difference in median values resulted in a p-value of 0.017. To measure the magnitude of the observed accuracy difference in effort BRE, we calculated Cohen's size of effect measure (d), with the result d = 0.5, which is considered a medium effect [48]. The results are summarized in Table 4.

TABLE 4
Statistical Summary

| | | p | d |
|---|---|---|---|
| **Effort** | **Accuracy (BRE)** | 0.017 | 0.47 |
| | **Bias (BREbias)** | 0.007 | 0.56 |
| | **Accuracy (MRE)** | 0.029 | 0.54 |
| | **Bias (RE)** | 0.007 | 0.69 |
| **Schedule** | **Accuracy (BRE)** | 0.211 | 0.42 |
| | **Bias (BREbias)** | 0.356 | 0.39 |
| | **Accuracy (MRE)** | 0.193 | 0.44 |
| | **Bias (RE)** | 0.356 | 0.34 |

As indicated in the statistical summary, it appears as if the results are similar whether the analysis is based on BRE or MRE. In the remainder of the paper, we will therefore only present the results based on BRE and BREbias.

Regarding effort estimation accuracy and bias, the analysis indicates that there is a possible difference, whereas the results for schedule estimation accuracy and bias remain inconclusive. The size of the effect (Cohen's d) for differences in effort estimation accuracy are located around d = 0.5, indicating a medium effect. Even though the results regarding schedule estimation were inconclusive, Cohen's d indicates a small to medium (0.2-0.5) effect.

As an example of how the results are distributed, a graphical representation of the BREbias data for effort is shown in Figs. 1 and 2.

When comparing the figures, one can, for example, notice that, for the sequential projects, the same proportion (about 20 percent) have a BREbias around 0.0 as have a BREbias around 1.0. This indicates that 20 percent of the sequential projects had accurate estimates and that 20 percent of them used twice the amount of effort originally estimated.

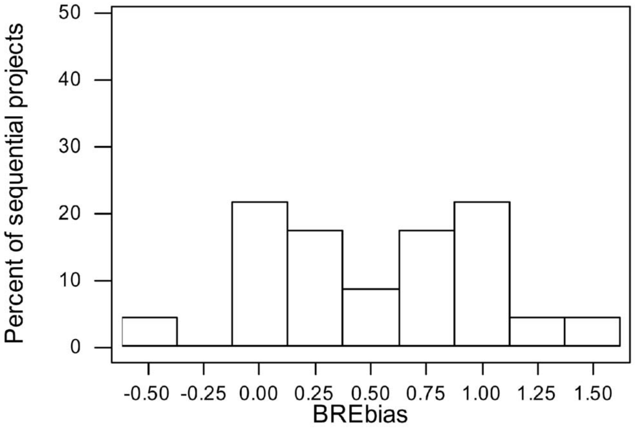

Fig. 1. Effort estimation bias of sequential projects.

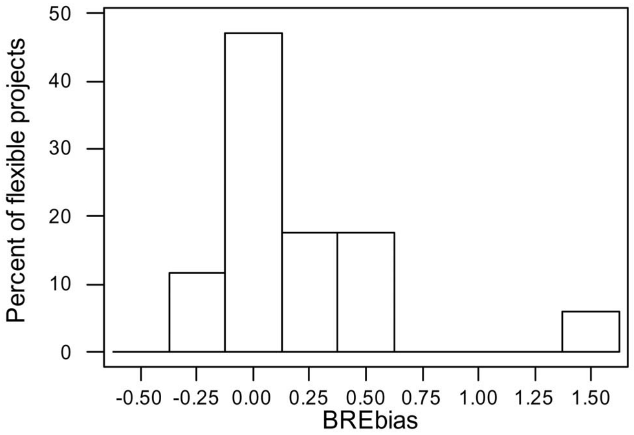

Fig. 2. Effort estimation bias of flexible projects.

For the sequential projects, we can see that almost 50 percent of them had estimates that were accurate.

## 7 Discussion

Our results suggest that the projects that used flexible development models were less prone to effort overruns than were those which used a sequential model. The lack of a significant difference in schedule overruns is, we believe, a result of the low number of observations, i.e., low power of the study compared to the size of the effect. We therefore need more observations to test the difference in schedule estimation accuracy.

The reasons for these observations are more difficult to discern. In general, it is not possible to provide a causal relationship in a survey. In order to provide such relationships, one usually needs to conduct controlled experiments. Such controlled experiments are very difficult to conduct with real-life software engineering projects. However, we are able to investigate several properties of the project groups in order to better understand our observations. The following sections investigate possible explanations. We mainly address the explanations often mentioned as the advantages of flexible development, such as different estimation processes [16] and flexibility in delivered functionality [23]. In addition, we investigate if there were substantial differences in project size between the groups, which may explain our findings. Finally, we analyze the responses made by the project managers on project outcome in order to generate new hypotheses and provide a basis for further work.

### 7.1 Different Estimation Process

There are two elements related to the estimation process which could explain our observation if one or both of the elements had a more significant presence in the flexible projects than in the sequential projects. These are:

- more estimation revisions, conducted as the project proceeds, and
- a more advanced estimation process with the use of experience databases, WBS, checklists, or other supporting tools.

Proponents of flexible development models have specified several estimation revisions as one of the reasons for reduced overruns when these models are applied [16]. Obviously, the closer a project is to completion, the easier it is to estimate the total effort and schedule. Therefore, it is important to remember that, when we compare estimation accuracy in our survey, we compare the estimates *used internally at the planning stage* and compare those with the actual result (as explained in Section 4).

In any case, surprisingly enough, we observed that the number of revisions of the estimates did not seem to be different for flexible and sequential projects. Only a couple of the flexible projects reestimated the project during development. However, this was also done for a couple of the sequential projects, but neither of them had more than one revision.

The projects were mainly estimated during the early and/or the planning stage. Also, here, there were no differences between the sequential and flexible projects, with half of them estimated during both the early-phase and the planning-phase, while the rest only estimated in one of the phases.

Since it is possible that the projects that employ flexible development models are more mature related to development than those which rely on sequential development, it is possible that they also have a better estimation process, which may be the main explanation for the reduced overruns. Regarding the quality of the estimation process, all the projects involved expert judgment-based, bottom-up estimation, and there were no differences among the groups related to the use of supporting estimation tools, such as checklists, experience databases, and predefined work breakdown structures. For the flexible projects, 15 used expert estimation and four used a combination of expert estimation and estimation models. Five used WBS, eight experience databases (four of them informal), and one used checklists. For the sequential projects, 20 used expert estimation and three used a combination of expert estimation and estimation models. Eight used WBS, five experience databases (four of them informal), and eight used checklists. Out of all the sequential and flexible projects that used a combination of expert estimates and estimation model, all but one relied on a use-case-based model.

To summarize, the choice of estimation process did not appear to correlate with choice of development method. The estimation process applied seems to be independent of the development model used. It is therefore unlikely that the observed difference in effort estimation accuracy and bias between the groups was caused by a systematic difference in the estimation process.

### 7.2 More Flexibility in Delivered Functionality

We found no difference related to delivered functionality that depended on the development model used. The mean proportion of delivered functionality for both groups was 106 percent of estimated functionality. The managers were also asked to provide free-text responses to eventual overruns. None of the managers stated that the development of unnecessary functionality contributed to the overruns. This implies that, at least from the managers' point of view, differences in overruns between sequential and

TABLE 5
Reasons Provided by Managers

| | *Sequential* | *Flexible* | *Difference* |
|---|---|---|---|
| | *(n=23)* | *(n=19)* | |
| ***Reasons for accuracy*** | | | |
| **Internal** | | | |
| High skill | 17.4% | 15.8% | 1.6% |
| Good development process | **8.7%** | **0.0%** | 8.7% |
| Good estimation process | 4.3% | 5.3% | -0.9% |
| **Client/collaboration** | | | |
| Good requirement specification | 0.0% | 15.8% | **-15.8%** |
| Good collaboration/communication | 0.0% | 21.1% | **-21.1%** |
| Good change management | 0.0% | 5.3% | -5.3% |
| Fixed price contract | 0.0% | 5.3% | -5.3% |
| Mature/professional client | 4.3% | 0.0% | 4.3% |
| High project priority | 8.7% | 0.0% | 8.7% |
| Low uncertainty | 4.3% | 5.3% | -0.9% |
| Known technology | 4.3% | 15.8% | **-11.4%** |
| ***Reasons for inaccuracy*** | | | |
| **Internal** | | | |
| Quality most important | 8.7% | 5.3% | 3.4% |
| Poor development process | **8.7%** | **0.0%** | 8.7% |
| Price-to-win | 8.7% | 5.3% | 3.4% |
| Lack of skill | 17.4% | 10.5% | 6.9% |
| Poor internal communication | 8.7% | 0.0% | 8.7% |
| Subcontractor problems | 17.4% | 5.3% | **12.1%** |
| **Client/collaboration** | | | |
| Weak/ambiguous requirement specification | **17.4%** | **26.3%** | -8.9% |
| Difficult customer procurement process | 8.7% | 5.3% | 3.4% |
| Immature client | 8.7% | 5.3% | 3.4% |
| Poor collaboration/communication | **17.4%** | **15.8%** | 1.6% |
| Slow customer decisions | 8.7% | 10.5% | -1.8% |
| Scope creep | 13.0% | 5.3% | 7.8% |
| Chaotic environment | 0.0% | 5.3% | -5.3% |
| Contract problems | 0.0% | 5.3% | -5.3% |
| To many people involved | 0.0% | 5.3% | -5.3% |
| New technology | 17.4% | 5.3% | **12.1%** |

flexible projects are not due to differences in the amount of delivered functionality.

### 7.3 Smaller Projects

There have been studies describing situations where the effort overruns of software projects increase with project size [24], [49]. If large projects more frequently follow a sequential development process, this may explain the difference in estimation accuracy. However, an analysis of the data suggests that this cannot explain the observed difference between the flexible and sequential projects since the choice of development model did not seem to correlate with the size of the project. Mean actual project effort was 3,296 hours for projects that followed a flexible development model, as opposed to 2,983 hours for the projects that followed a sequential development model.

### 7.4 Reasons Reported by the Project Managers

The managers provided a variety of free-text responses when asked about the project outcome related to the effort and schedule estimates and delivered functionality. They reported both reasons for accuracy and reasons for inaccuracy. For both of these categories, some reasons are related to internal factors at the contractors, whereas some are related to the client or collaboration. As previously mentioned, we must be careful when interpreting these results as they may be affected by hindsight bias and the managers' subjective views. Another method for data collection might have provided a different result [24].

An overview of the manager responses, split into the sequential and flexible projects, is presented in Table 5. We have boldfaced what we feel are the most interesting observations in the table.

TABLE 6
Mean BREbias Based on Client and Development Model Type

| | *Flexible* | *Sequential* | *Total* |
|---|---|---|---|
| **Private** | -0.02 (n=11) | 0.40 (n=13) | 0.21 (n=24) |
| **Public** | 0.58 (n=8) | 0.74 (n=10) | 0.67 (n=18) |
| **Total** | 0.24 (n=19) | 0.55 (n=23) | *0.41 (n=42))* |

A value of, e.g., 15.8 percent, for *high skill* in the flexible column indicates that 15.8 percent of the projects in that group cited *high skill* as a factor that contributed to increased estimation accuracy. As seen in the table, there were not many large differences in the managers' responses. A positive difference indicates a higher presence of the reasons in the sequential projects, whereas a negative difference indicates the opposite.

The two reasons with the largest difference between sequential and flexible projects are that more managers on flexible projects cite good requirement specifications and good collaboration/communication as providing good estimation accuracy. However, a high percentage of *both* flexible and sequential projects cite "poor requirement specification" and "poor collaboration/communication" as contributing to inaccuracy as well. The other three factors with a difference larger than 10 percent are:

- A larger percentage of flexible projects cite a preference for known technology as contributing to more estimation accuracy.
- Correspondingly, a larger percentage of sequential projects cite a preference for new technology as contributing to inaccurate estimates.
- A larger percentage of sequential projects cite problems with subcontractor(s) for new technology as contributing to inaccurate estimates.

In addition, it is also interesting to notice that *none* of the flexible projects cited a "good development process" as contributing to accurate estimates. However, neither did any of the flexible projects cite a "poor development process" as contributing to inaccuracy. For the sequential projects, there was an 8.7 percent response in both categories.

In order to explore this, a further analysis of the clients is required. An interesting observation in our study was that projects with public (government) clients had mean effort overruns three times larger than had projects with private clients, i.e., BREbias of 0.67 versus 0.21. However, there was no difference in the use of development models related to the type of client, i.e., differences in the type of client cannot explain the observed difference in estimation accuracy based on development model. Table 6 displays the mean effort estimation bias (BREbias) dependent on type of model and client.

Table 6 indicates that the difference in BREbias between sequential and flexible projects is larger when there was a private client (0.42) than when there was a public client (0.16). The trend is similar for the other measures used previously in this paper (BRE, MRE, and MREbias).

A further analysis of the manager responses for estimation accuracy and inaccuracy is therefore interesting. An overview is presented in Table 7.

Most interesting here are the figures in bold. A majority of public projects were described as having some sort of problem related to the client and/or collaboration, independent of the development model. However, only one (9.1 percent) of the flexible private projects described a problem related to the client and/or collaboration. On the contrary, almost half of them described client and/or collaboration reasons as contributing to increased estimation accuracy. As previously underscored, we must be careful with interpreting these free-text responses. However, they are suitable for generating hypotheses for further studies.

It is believed, based on international and Norwegian studies, that public projects have confusing or contradictory goals more frequently than private projects (or, indeed, lack goals altogether), a diffusion of managerial responsibility, limited user involvement, and legislation constraints [49], [50]. It has also been frequently stated that public projects have a shortage of competent staffing on the client side with respect to IT and project management competence [51].

These problems illuminate our observations. It may, for example, be a prerequisite for successful implementation of software projects with a flexible development approach that the clients are competent in IT and project management and

TABLE 7
Reasons by Client and Development Model

| | *Public* | | *Private* | |
|---|---|---|---|---|
| | sequential (n=10) | Flexible (n=8) | Sequential (n=13) | Flexible (n=11) |
| **Reasons for accuracy** | | | | |
| Internal | 10.0% | 37.5% | 38.5% | 0.0% |
| Client/collaboration | 10.0% | 25.0% | 23.1% | **45.5%** |
| **Reasons for inaccuracy** | | | | |
| Internal | 70.0% | 25.0% | 38.5% | 27.3% |
| Client/collaboration | **70.0%** | **75.0%** | 46.2% | **9.1%** |

able to deliver fast and unambiguous responses to the contractors inquiries.

### 7.5 Threats to Validity

We believe the most important threats to validity are the following:

- small sample size,
- the projects studied are not necessarily representative of projects in other countries and in other situations,
- some of the data are subjective and subject to different interpretations, e.g., the proportion of delivered functionality,
- no uniform terminology for description of development models and estimates, and
- misinterpretation of the development method in categorization.

The small sample size was the result of a trade-off between quality of data and number of observations. We decided to focus on quality of data. The projects we studied may not be representative of other types of project. However, the accuracy results we report are similar to those reported by other surveys on effort estimation presented in a recent review [2]. The problems of subjective data and a lack of common interpretations of terminology are difficult to solve. We have tried to solve them through systematic interview processes and independent assessors.

## 8 Conclusion and Further Work

It appears as if Mills may have made a good point when, as far back as 1976, he stated that "The evolution of large systems in small stages, with user feedback and participation in goal refinements at each step is a way of going from grandiose to grand software system development" [17].

We found that projects which used a flexible development model, (e.g., evolutionary or incremental) were more likely to have a lesser magnitude effort overruns than comparable projects which used a sequential (waterfall) development model. The reason for this difference is not obvious as software projects are complicated entities with an almost infinite amount of interacting variables. In short, we found support for RQ1 (*Are there significant differences in the occurrence of effort overruns between projects following flexible and sequential development models, respectively?*), whereas RQ2 (*Are there significant differences in the occurrence of schedule overruns between projects following flexible and sequential development models, respectively?*) remains inconclusive.

We were able to investigate some of the common explanations frequently attributed to flexible projects, e.g., less delivered functionality and more estimation revisions. These explanations, or the assertion that flexible development methods are more frequently applied to smaller projects, do not appear to be valid for our data set as there were no differences in these factors that correlated with the type of development model. It is also important to note that the estimation approach did not vary with respect to the type of development model. Almost all projects were estimated with an informal expert approach, whereas the rest were estimated with a combination of expert opinion and a use-case-based model. Similarly, there were no variances in the use of checklists, WBS, combination of estimates, or other estimation related aspects based on the choice of development method.

Responses from the managers suggest that a flexible development model correlates with a positive dialogue between client and developers. Most likely, such a positive dialogue requires competent clients. Regarding the type of feedback (e.g., from customers), which may be important for project control in flexible projects, we regrettably had no mechanism for controlled collection of this in our survey. We had to rely on the managers' responses postmortem.

We feel that this is surely an important topic for further research. This can, for example, be done in a multiple case study, where flexible and sequential projects (with otherwise similar properties) are logged/monitored with regards to such aspects as the interaction with clients, feedback-loops, etc. We are also considering other work focusing on client-contractor aspects of software projects, e.g., how involvement at an early stage and frequent communication between clients and developers, in combination with flexible development models, may lead to better software projects with respect to satisfying stakeholder needs.

## Acknowledgments

This research was funded by the Research Council of Norway under the project INCO. The authors would like to thank Sinan S. Tanilkan, Hans Gallis, Anette Lien, and Siw E. Hove for execution of the study, and Dag Sjøberg, Tom Gilb, Erik Arisholm, Martin Shepperd, Erik Stensrud, and Chris Wright for valuable comments.

## References

[1] E. Yourdon, *Death March.* Prentice-Hall, 1997.

[2] K. Moløkken-Østvold and M. Jørgensen, "A Review of Surveys on Software Effort Estimation," *Proc. 2003 ACM-IEEE Int'l Symp. Empirical Software Eng. (ISESE 2003),* pp. 220-230, 2003.

[3] B. Boehm, *Software Estimation with COCOMO II.* Prentice-Hall, 2000.

[4] J.E. Matson, B.E. Barrett, and J.M. Mellichamp, "Software Development Cost Estimation Using Function Points," *IEEE Trans. Software Eng.,* vol. 20, no. 4, pp. 275-287, 1994.

[5] M. Jørgensen, "A Review of Studies on Expert Estimation of Software Development Effort," *J. Systems and Software,* vol. 70, nos. 1-2, pp. 37-60, 2004.

[6] H. Woodward, *Project Management Institute Practice Standard for Work Breakdown Structures.* Project Management Inst., 2001.

[7] R.C. Tausworthe, "The Work Breakdown Structure in Software Project Management," *J. Systems and Software,* vol. 1, pp. 181-186, 1980.

[8] M. Jørgensen and K. Moløkken-Østvold, "A Preliminary Checklist for Software Cost Management," *Proc. Int'l Conf. Quality Software,* pp. 134-140, 2003.

[9] M. Shepperd and U. Passing, "An Experiment on Software Project Size and Effort Estimation," *Proc. 2003 ACM-IEEE Int'l Symp. Empirical Software Eng. (ISESE 2003),* pp. 120-129, 2003.

[10] S. Engelkamp, S. Hartkopf, and P. Brossler, "Project Experience Database: A Report Based on First Practical Experience," *Proc. Int'l Conf. Product Focused Software Process Improvement,* pp. 204-215, 2000.

[11] K. Moløkken-Østvold and M. Jørgensen, "Software Effort Estimation: Unstructured Group Discussion as a Method to Reduce Individual Biases," *Proc. 15th Ann. Workshop Psychology of Programming Interest Group (PPIG 2003),* pp. 285-296, 2003.

[12] B. Flyvbjerg, M.S. Holm, and S. Buhl, "Underestimating Costs in Public Works Projects—Error or Lie?" *J. Am. Planning Assoc.,* vol. 68, no. 3, pp. 279-295, 2002.

[13] M. Jørgensen, "Practical Guidelines for Expert-Judgement-Based Software Effort Estimation," *IEEE Software,* pp. 57-63, May/June 2005.

[14] E.L. May and B.A. Zimmer, "The Evolutionary Development Model for Software," *Hewlett-Packard J.,* vol. 47, no. 4, pp. 39-45, 1996.

[15] D.R. Graham, "Incremental Development and Delivery for Large Software Systems," *Software Eng. for Large Software Systems,* 1990.

[16] T. Gilb, "Estimating Software Attributes: Some Unconventional Points of View," *ACM Sigsoft Software Eng. Notes,* vol. 11, no. 1, pp. 49-59, 1986.

[17] H.D. Mills, "Software Development," *IEEE Trans. Software Eng.,* vol. 2, no. 4, pp. 265-273, 1976.

[18] C. Larman and V.R. Basili, "Iterative and Incremental Development: A Brief History," *Computer,* pp. 2-11, June 2003.

[19] K. Moløkken-Østvold et al., "Does Use of Development Model Affect Estimation Accuracy and Bias?" *Proc. Fifth Int'l Conf. Product Focused Software Process Improvement PROFES 2004,* pp. 17-29, 2004.

[20] M. Iansiti and A. MacCormack, "Developing Products on Internet Time," *Harvard Business Rev.,* pp. 107-117, Sept. 1997.

[21] A. Cockburn, "In Search of Methodology," *Object Magazine,* pp. 52-56, 1994.

[22] A. Cockburn, *Surviving Object-Oriented Projects.* Addison-Wesley, 1998.

[23] T. Gilb, *Principles of Software Engineering Management.* Addison-Wesley, 1998.

[24] M. Jørgensen and K. Moløkken-Østvold, "Understanding Reasons for Errors in Software Effort Estimates," *IEEE Trans. Software Eng.,* vol. 30, no. 12, pp. 993-1007, Dec. 2004.

[25] M. Jørgensen and D.I.K. Sjøberg, "Impact of Effort Estimates on Software Project Work," *Information and Software Technology,* vol. 43, pp. 939-948, 2001.

[26] H.D. Mills, "The Management of Software Engineering, Part 1: Principles of Software Engineering," *IBM Systems J.,* vol. 19, no. 4, pp. 414-420, 1980.

[27] T.D. Hendrix and M.P. Schnenider, "NASA's TReK Project: A Case Study in Using the Spiral Model of Software Development," *Comm. ACM,* vol. 45, no. 4, 2002.

[28] W. Royce, "TRW's Ada Process Model for Incremental Development of Large Software Systems," *Proc. 12th Int'l Conf. Software Eng. (ICSE '12),* pp. 2-11, 1990.

[29] A.C. Lien and E. Arisholm, "Evolutionary Development of Web-Applications—Lessons Learned," *Proc. European Software Process Improvement Conf. (EuroSPI '2001),* 2001.

[30] W. Royce, "Managing the Development of Large Software Systems: Concepts and Techniques," *Proc. IEEE WESTCON Conf.,* pp. 1-9, 1970.

[31] R. Pressman, *Software Engineering,* fifth ed. McGraw-Hill, 2000.

[32] T. Gilb, *Software Metrics.* Little, Brand and Co., 1976.

[33] P. Abrahamson et al., *Agile Software Development Methods. Review and Analysis.* p. 107, VTT Publication, 2002.

[34] P.C. Cozby, *Methods in Behavioral Research,* fifth ed. Mountain View, Calif.: Mayfield Publishing, 1993.

[35] HegnarOnline, Kapital DATAs 1500 Største, Nov. 2002, http://www.hegnar.no/external/default.asp?nSpace=ns_28.

[36] S. Grimstad, M. Jørgensen, and K. Moløkken-Østvold, "Software Effort Estimation Terminology: The Tower of Babel," *Information and Software Technology,* 2005.

[37] S.D. Conte, H.E. Dunsmore, and V.Y. Shen, *Software Engineering Metrics and Models.* Menlo Park, Calif.: Benjamin-Cummings, 1986.

[38] L.C. Briand and I. Wieczorek, "Resource Modeling in Software Engineering," *Encyclopedia of Software Eng.,* J. Marciniak, ed., Wiley, 2001.

[39] T. Foss et al., "A Simulation Study of the Model Evaluation Criterion MMRE," *IEEE Trans. Software Eng.,* vol. 29, no. 11, pp. 985-995, Nov. 2003.

[40] M. Jørgensen and D. Sjøberg, "An Effort Prediction Interval Approach Based on the Empirical Distribution of Previous Estimation Accuracy," *J. Information and Software Technology,* vol. 45, no. 3, pp. 123-136, 2003.

[41] E. Stensrud et al., "An Empirical Validation of the Relationship between the Magnitude of Relative Error and Project Size," *Proc. Eighth IEEE Symp. Software Metrics,* pp. 3-12, 2002.

[42] E. Stensrud et al., "A Further Empirical Investigation of the Relationship Between MRE and Project Size," *Empirical Software Eng.,* vol. 8, no. 2, pp. 139-161, 2003.

[43] B. Kitchenham et al., "What Accuracy Statistics Really Measure," *IEE Proc.—Software Eng.,* vol. 148, no. 3, pp. 81-85, 2001.

[44] Y. Miyazaki et al., "Method to Estimate Parameter Values in Software Prediction Models," *Information and Software Technology,* vol. 33, no. 3, pp. 239-243, 1991.

[45] J.S. Edwards and T.T. Moores, "A Conflict between the Use of Estimating and Planning Tools in the Management of Information Systems," *European J. Information Systems,* vol. 3, no. 2, pp. 139-147, 1994.

[46] R. Christensen, *Analysis of Variance, Design and Regression. Applied Statistical Methods.* Chapman and Hall/Crc., 1998.

[47] S. Siegel and N.J. Castellan, *Nonparametric Statistics for the Behavioral Sciences,* second ed. McGraw-Hill College Div., 1988.

[48] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences.* New York: Academic Press, 1969.

[49] P. Sørgaard and M. Vestre, "Country Report from Norway," *OECD-PUMA Expert Meeting on Management of Large IT Projects,* 2003, http://www.statskonsult.no/publik/rapporter/2003/2003-01eng.pdf.

[50] L . West-Knights, "Getting IT Right for Government—A Review of Public Sector IT Projects," 2000, Intellect, http://www.intellectuk.org/publications/business_guidance_papers/Get_IT_Right_for_Govt.pdf.

[51] K. Moløkken-Østvold, "Management of Public Software Projects: Avoiding Overruns," *Proc. Hawaiian Int'l Conf. Business,* pp. 2210-2220, 2005.

**Kjetil Moløkken-Østvold** received the MSc and PhD degrees in informatics from the University of Oslo, Norway, in 2002 and 2004, respectively. He is an assistant director at the Simula Research Laboratory in Norway. His main research interests are software estimation, group processes, management of public projects, and software process improvement. He is a member of the IEEE.

**Magne Jørgensen** received the Diplom Ingeneur degree in Wirtschaftswissenschaften from the University of Karlsruhe, Germany, in 1988 and the Dr. Scient. degree in informatics from the University of Oslo, Norway, in 1994. He is a professor of software engineering at the University of Oslo and a member of the software engineering research group at the Simula Research Laboratory. He has 10 years industry experience as a consultant and manager. His research interests include software estimation, uncertainty assessments in software projects, expert judgment processes, and learning from experience.