The Relationship between Customer Collaboration and Software Project Overruns

Kjetil Moløkken-Østvold[1] and Kristian Marius Furulund[2].

[1] Simula Research Laboratory, P.O.Box 134, 1325 Lysaker, Norway
kjetilmo@simula.no

[2] Department of Informatics, University of Oslo, P.O.Box 1080 Blindern, 0316 Oslo, Norway
kristf@ifi.uio.no

Abstract

Most agile projects rely heavily on good collaboration with the customer in order to achieve project goals and avoid overruns. However, the role of the customer in software projects is not fully understood. Often, successful projects are attributed to developer competence, while unsuccessful projects are attributed to customer incompetence. A study was conducted on eighteen of the latest projects of a software contractor. Quantitative project data was collected, and project managers interviewed, on several issues related to estimates, key project properties, and project outcome. It was found that in projects where collaboration was facilitated by daily communication between the contractor and the customer, they experienced a lesser magnitude of effort overruns. In addition, employing a contract that facilitates risk-sharing may also have a positive impact.

1. Introduction

Project overruns pose a major and persistent challenge for software development [1]. However, a recent study found that projects that used flexible development processes (agile, evolutionary and/or incremental) had an average magnitude of overruns that was significantly less than for projects that employed sequential (waterfall) development processes [2]. The median effort overrun was 1% for the projects that used a flexible process, compared to 60% for those that used a sequential process. The projects were similar in other key respects, including estimation approach, project size and proportion of delivered functionality. In-depth interviews found that, from the perspective of the project managers, flexible development processes fostered good collaboration with the customer to a greater extent than sequential processes [2].

The *Agile Manifesto* stresses the importance of customer collaboration. An excerpt taken from http://www.agilemanifesto.org/ reads:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

The importance of customer collaboration in agile projects is further explored and explained in central textbooks, e.g., on XP [3], Scrum [4] and agile estimating and planning [5]. Mike Cohn elaborates: "*Customer collaboration is valued over contract negotiation because agile teams would like all parties to the project to be working towards the same set of goals" [5]*. Beck and Fowler state that a development project must adjust its course constantly, and further write that "*Communicating is the point. We've seen too many requirement documents written down that don't involve communication*" [3]. Frequent communication can be used to prioritize features, set focus on bug-fixing or include more functionality [3].

However, what constitutes good collaboration, how can it be achieved, and can it help to reduce overruns? This paper explores the possible relationship between one of the key agile principles, customer collaboration, and software project overruns. Section 2 provides an account of the previous research that motivated the research questions, which are presented in Section 3. Section 4 describes the methods used to collect and analyze the data, which is presented in Section 5. Section 6 discusses the data in the light of the research questions. Section 7 concludes.

2. Background

Recommending a high degree of customer involvement and collaboration is nothing new. In his book on Scrum [4], Ken Schwaber presents his experiences from the 1960s in the chapter about the role of the product owner: *"When I started developing software, customer involvement and collaboration weren't problems. … I used short iterations of one or two days. I'd meet with the customer, and we'd sketch out on paper what he or she wanted. We'd discuss the problem until I understood it. … We didn't realize it at the time, but this was heaven."*

In a textbook from 1976, Tom Gilb, who introduced evolutionary development, stated 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…"* [6].

The same year, Harlan D. Mills wrote 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"* [7].

Alistair Cockburn has presented a case for seeing software development as a series of resource-limited, goal-directed cooperative games of invention and communication [8]. His proposal was a reaction to both the existing view, and the failure to find a correlation between project success and the use of tidy engineering practices. He stresses that in a resource-limited game, communication, whether within the team, or between the team and users, is paramount. Rapid feedback is deemed to be particularly important. He summarizes the issues known to be important to project success within his model as: *cooperation, communication, cost-of-, rate-of-, and sufficiency in-communication*.

Ken Schwaber also presents his view on how collaboration and communication was hampered by the introduction of more process [4], which widened the gap between developers and stakeholders. From his perspective, the waterfall methodology, as it came to be commonly applied, embodied all the flaws of sequential development.

When it comes to the research community, the impact of poor collaboration between contractor/developer and customer on software overruns has not been studied to any great extent. The study described briefly in Section 1 found that projects that used flexible development processes had an average magnitude of overruns that was significantly less than for projects that employed sequential development processes, and that customer collaboration was important [2]. In addition, other studies have found that from the perspective of software professionals, the behavior of customers affects the accuracy of software estimates.

A systematic review of previous studies found that customer characteristics were often mentioned as both preventing and facilitating overruns [9]. None of the studies explicitly investigated the influence of customers on estimation accuracy, but seven out of eight studies reported that customer characteristics are perceived as important for estimation accuracy. Some of the reviewed studies emphasized the impact of project flexibility and communication [10], some involvement and commitment [11], and others communication and understanding [12]. In addition, it was found that it is easy for managers on the developer's side, to attribute failure to the customer, and correspondingly attribute success to themselves [9].

However, the findings from previous studies are diverging and incomplete, and the area is far from understood. The differences can probably be explained by variance in purpose of the studies, terminology ambiguousness, various roles of the respondents, the method of analysis and the differences in sample sizes.

To follow up on the review of previous studies on the customer role, a study with 307 Norwegian software professionals was performed, in which the perceptions of the professionals were gathered by questionnaire [9]. The three customer-related reasons most frequently perceived as contributing to project overruns were 1) frequent requirement changes and new requirements, 2) lack of well defined requirements and 3) lack of competent customers able to make decisions. The most important reasons perceived as preventing project overruns were 1) competent customers able to make decisions, 2) well defined requirements and 3) adequate project administration [9].

Changed and new requirements were perceived to be the customer's most frequent contribution to overruns. Here, it is interesting to note the agile community's attitude towards welcoming change, as opposed to treating changes as unwelcome.

Availability of *competent customers* and *capable decision makers* was found to be the most important success-factor [9]. However, terms such as customer competence and customer decision-making ability are ambiguous.

The respondents also reported the estimation accuracy of their last completed project, and the projects with large overruns differed most from the projects that had smaller overruns with regard to 1) less realism in plans and budgets, 2) less project flexibility and 3) poorer customer and vendor communication.

3. Research Questions and Definitions

The Agile manifesto emphasizes "**Customer collaboration** over contract negotiation". This, and research on the possible importance of customers and collaboration that was presented in Sections 1 and 2 motivated the research questions in this section.

It is virtually impossible to measure *good collaboration* objectively in an industrial setting. Instead, we have to investigate key properties that are relevant for collaboration, and rely on manager responses. The study that found a relationship between flexible development processes and lesser overruns, also uncovered that the respondents felt that good collaboration and communication could help reduce overruns [2]. In a study by Verner et. al. that explored factors that contribute to the success and failure of projects [10], the managers who responded, reported good communication with customers as the foremost success property. However, communication and other related concepts are, to the same extent as is, collaboration, not precisely defined terms. Hence, motivated in part by Cockburn [8], we explore the *frequency of communication between the contractor and the customer*. This is a more objective measure and is also, we believe, representative of good collaboration.

RQ1: *Do projects with daily contact between contractor and customer have a lesser magnitude of effort overruns than do other projects?*

Beck and Fowler state that a fundamental problem with fixed-price contracts is that they pit the interests of supplier and customer against each other. It also appears as if many types of contracts used in the public sector has favored, or even required, a sequential (i.e. waterfall) approach. This has been previously observed both in Norway [13], the United States [14] and Australia [15], and many contractors have not been allowed to use more agile approaches. A consequence of this, as described in a recent paper by Jamieson, Vinsen and Callender [16] is that "*…it is probable that a supplier will have accepted terms and conditions relating to scope and price that are philosophically in opposition to agile principles, particularly in respect of an iterative elaboration of requirements*". However, there appears to have been a recent change away from the strict requirements of a waterfall approach [15].

Concerns regarding estimation and contracts are frequently omitted in the agile literature. As described by Jamieson, Vinsen and Callender [15]: "*Software can be developed in-house, but is more often obtained from vendors either as a package or through bespoke software development services*" and "*contemporary agile methods of software development do not appear to consider the role of the procurement process in influencing success, although the agile Unified Process (UP) recommends running projects in two contract phases, each of multiple timeboxed iterations.*"

We are aware of several organizations in Norway who are proponents of introducing a risk-sharing mechanism into contracts. In addition a new contract standard aimed at agile projects and risk-sharing in the public sector is currently being developed. However, no one has explored the effects of different types of contracts on overruns and other related aspects. Hence, we investigated differences in overruns that are related to procurement and *type of contract.*

RQ2: *Do projects that employ contracts facilitating risk-sharing have a lesser magnitude of effort overruns than do other projects?*

Regarding type of contract, we differentiated among the following: 1) by the hour (time and material), 2) fixed price, 3) target price (risk sharing between contractor and customer), and 4) other.

A target price contract is a contract that shares the risk of overruns between the contractor and the customer. The contract defines a target of effort (in hours), and often a date, for completing the project.

If the project is completed, and has used fewer hours than defined by the contract, the contractor and customer share the profit. If more effort than what is defined by the target is needed, the contractor and the customer share the extra costs. This is set up so that the customer pays a percentage, often 50%, of full price for the hours that make up the difference between the target and actual effort.

E.g., the contractor charges $ 150/hour and the target for completing the project is set to 1000 hours. If 800 hours is used to complete the project the customer has to pay full price for the 800 hours used and $75 for the remaining 200 hours (not used). If 1200 hours is used the customer pays full price for the 1000 hours set by the target, and $75 for the additional 200 hours.

A target contract can also be set up with floors and ceilings defining the minimum and maximum hours that the contractor could charge the customer.

Related to collaboration, the customers procurement capability (frequently labeled maturity) have been said to have an impact on overruns [9]. This suggested impact has typically been found in responses from managers that tend to explain overruns by citing vague goals, lack of IT competence and poor management. This motivated the following research question:

RQ3: *Do projects where the customer is perceived to have a good procurement capability have a lesser magnitude of effort overruns than do other projects?*

The customer procurement capability we investigate is comprised of: a) their ability to collaborate, b) their IT skills, c) their decision-making ability and d) their clarity of goals

In the software engineering literature, the term *customer* is often not clearly defined; sometimes it refers to a user, at other times to those who finance a project. A *customer* is also frequently called *product owner* by many. Since there is no uniform definition of customer; we apply a coarse-grained definition here: as whoever it is that engages the services of professional software engineers to produce a piece of software.

Different types of projects include:

1. Contracting – A case in which a software contractor develops custom-built solutions for an outside party. In this case the role of customer resides with the outside party, and can be assigned to one of the eventual users of the end-product.
2. In-house development - A case in which a software department typically develops solutions for another department in the same company. In this case, the role of customer resides in another department, and can be assigned to one of the eventual users of the product.
3. Shrink-wrap software - A case in which a software department of a company develops software for sale to businesses and/or individuals. The role of customer is often assigned to a business manager.

This is not intended to be an exhaustive overview, but represent most typical cases. In the study reported herein, we explored projects in category 1.

4. Method

The study was conducted in Norway from March 14th to October 16th 2006.

4.1. The company studied

The company studied is a medium-sized Norwegian software consultancy that, at the time of the study, had about 300 employees. The company operates as an independent contractor and offers a wide range of complete software solutions to its various customers.

In order to account for limitations in previous surveys [2], we wanted to explore several projects within one company to better isolate effects such as communication and customer competence.

4.2. Data collection and analysis

We interviewed the project managers of 18 different projects. These projects were selected by the company, which was not informed of our research questions. The inclusion criteria were as follows (i) relevant project data was stored and available, (ii) we were able to interview the responsible project managers, and (iii) the projects had a workload of at least 100 man-hours. This last criterion is in line with previous surveys, in which "*trivial tasks routinely handled without effort estimation*" were also filtered out [17].

We collected data via personal interviews, which yields data of high quality and ensures that ambiguities are resolved [18]. It also allows the respondents to add valuable information that it is not possible to include when completing a predefined questionnaire. Another point in favor of this approach is that personal involvement on the part of the researchers indicates a seriousness of intent to the participants, and this may increase the likelihood of obtaining serious contributions from them. The main limitation of the approach is that it is time-consuming and hence prevented us from investigating as many projects as would be possible by using mailed questionnaires.

Each interview lasted between 20 and 80 minutes. The respondents were informed that their responses were anonymous, and that no feedback about the respondents' answers would be reported to outsiders or to company managers.

The following factors are explored in this paper: 1) communication frequency, 2) contract type, and 3) customer procurement capability. The measures of 3 are based on the subjective opinions of the project managers, while measures of 1 and 2 are more objective measures.

In addition, several questions were aimed at general project properties. In addition to having their own value, these questions are also relevant for isolating effects and exploring confounding factors in the study. These possible confounding factors included project size, perceived project complexity and technical knowledge.

When conducting research on the accuracy of estimates of software projects, it is necessary to differentiate between dissimilar types of estimates. What estimate(s) to use depends on the focus of the research. If the goal is to compare the actual effort with the estimated effort, as it is in this paper, it is meaningful to use the *most likely* estimates at the *planning stage*, instead of, for example, project bids. The latter may be affected by outside factors, such as market competition.

In order to compare actual effort and estimated effort, and measure any differences in project overruns dependent on the studied properties, we used the *BREbias* measure, previously used in related research, e.g., [2, 19]. It is calculated as:

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

The BREbias measures both the magnitude and direction of effect when comparing the actual effort to estimated effort. BREbias is based on the Balanced Relative Error (BRE) [20, 21].

Even though MRE has been the most widely used measure of estimation accuracy [22], one must be aware that it has undesirable properties [20, 23]. The main concern for our case is the fact that underestimated and overestimated projects are weighted unevenly.

To measure the size of any difference, we used Cohen's size of effect measure (d) [24], where:

$$d = \frac{\overline{sample1} - \overline{sample2}}{pooledStdDev}$$

We include the representation of the data in tables and figures, and provide statistical Kruskal-Wallis test for difference where those are appropriate. The Kruskal-Wallis is more robust for unequal sample sizes [25], as was the case in this study.

5. Results

Out of the 18 completed projects, three were overestimated, one was on target, while 14 were underestimated. A full account of key project data is provided in Appendix 1.

The mean and median BREbias were 0.27 and 0.22, respectively (this corresponds to an effort overrun of 27% and 22%). This is in line with findings in previous surveys and case studies on software estimation [26], and indicates that in this respect, the projects studied were fairly representative.

5.1. Communication frequency

The respondents stated how often they communicated with the customer. In this analysis, we have differentiated between those who had daily communication (11 projects), and those who did not have daily communication (seven projects). The results are displayed in Figure 1 and Table 1.

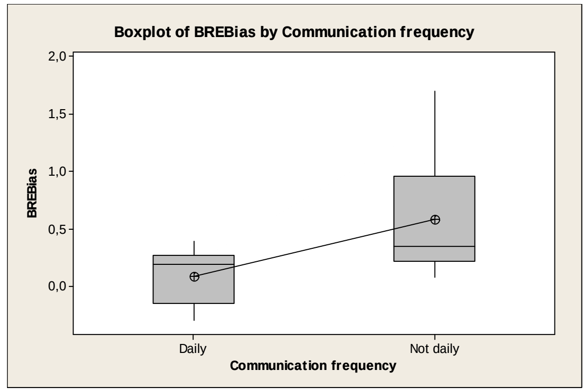

Figure 1: BREbias based on communication

Table 1: BREbias based on communication

Level	*N*	*Mean*	*Median*
Daily	11	0.09	0.19
Not Daily	7	0.58	0.35

The projects in which the developers did not have daily communication with the customer appear to have a larger magnitude of effort overruns. A Kruskal-Wallis test for difference results in p=0.023. The corresponding size of effect is d=1.25, indicating a large size of effect [24].

5.2. Type of contract

The following types of contracts were employed: 1) by the hour (time and material), 2) fixed price, 3) target price (risk sharing between contractor and customer), and 4) other.

The results are displayed in Figure 2 and Table 2.

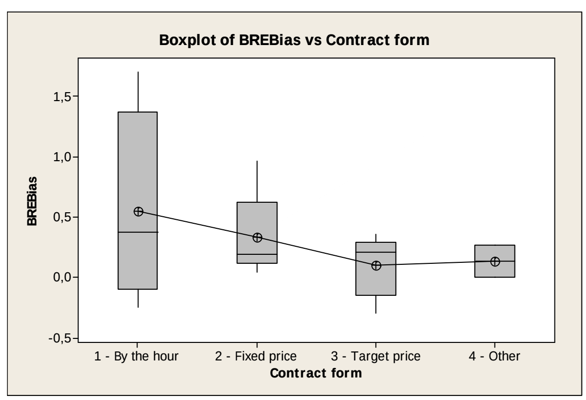

Figure 2: BREbias based on type of contract

Table 2: BREbias based on type of contract

Level	*N*	*Mean*	*Median*
By the hour	4	0.55	0.37
Fixed price	5	0.33	0.19
Target price	7	0.10	0.21
Other	2	0.13	0.13

There are relatively few observations in the different categories, so we do not include a statistical analysis. However, there appear to be some differences in overruns related to the different contract types.

5.3. Customer procurement capability

The managers were asked to rate the customer on different procurement capabilities, frequently presented as relevant in previous studies [9]. The managers rated the customer on a five-point Likert-scale (1= very good, 3= average, 5=very poor) on four different properties: collaboration skill (CO), IT-competence (IT), decision making ability (Dec.) and clarity of project goals (Goals).

In general, the managers gave their customers a high rating on most factors; mean scores were all above the neutral level. The scores were summarized in order to calculate a total customer score for each project. The best possible score was 4 (1 on all four properties), while the lowest possible was 20 (5 on all four properties). The projects received total scores from 4 to 15, with mean and median results for all projects calculated to 8.7 and 8 respectively. The main results are presented in Table 3.

Table 3: Customer capability as rated by the managers

	Co	*IT*	*Dec.*	*Goals*	***Sum***
Mean	1.8	2.8	2.2	1.9	**8.7**
Median	2	3	2	1	**8**

We differentiated between those who received a total score of 8 or better (rated as good or better), and those with a total score of 9 or worse. See Table 4 and Figure 3 for an overview.

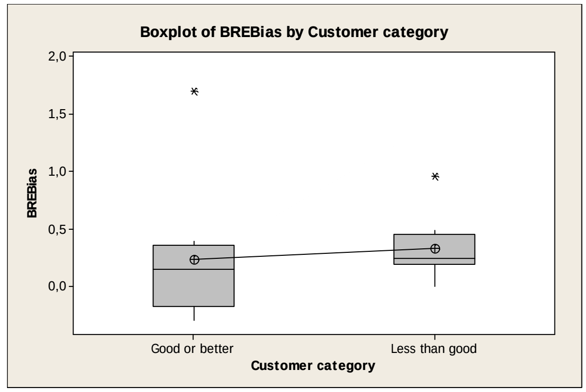

Figure 3: BREbias based on customer procurement capability

Table 4: BREbias based on customer procurement capability

Total score	*N*	*Mean*	*Median*
Good or better	10	0.23	0.15
Less than good	8	0.33	0.24

There was no apparent, large difference when comparing the total customer score. A Kruskal-Wallis test resulted in p=0.35, Cohen's d=0.21. However, there is a slight tendency towards a lesser magnitude of effort overruns when the customers are perceived as being more capable.

5.4. Possible confounding factors

In observational studies like this, where one does not have the same degree of control over variables as in an experiment, it is important to measure and control for possible confounding factors. These factors include differences in underlying project properties, and it is necessary to explore how they interact with the factors that are investigated. Typical project properties, as described in a framework for analyzing software overruns [27], that may have an impact on overruns include project size, project complexity, and familiarity with technology.

Regarding the effect of project size on overruns, we divided the projects into two equal samples, with a "small" and "big" group, based on actual effort. A representation of BREbias based on project size (measured in actual effort) is presented in Figure 4 and Table 5.

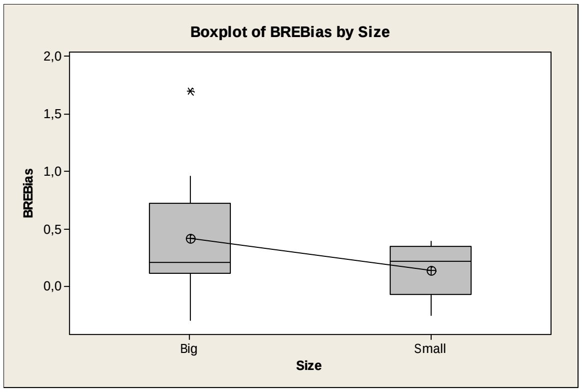

Figure 4: BREbias based on project size

Table 5: BREbias based on project size

Size	*N*	*Mean*	*Median*
Big	9	0.39	0.21
Small	9	0.14	0.22

As seen in Figure 4 and Table 5, the median values indicate that the typical case in both groups has a similar overrun. However, there are substantial differences in mean values. Taken together, this indicates a tendency for relatively large projects to have some incidents with a large magnitude of overrun.

Complexity is a more subjective measure. The respondents rated the project complexity on a scale of: high, medium and low. No projects received a "low" score. BREbias based on complexity is presented in Figure 5 and Table 6.

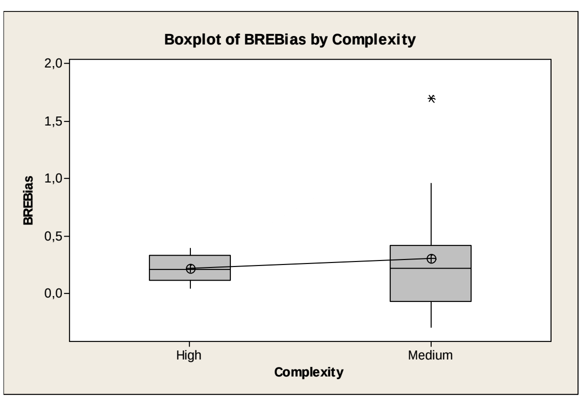

Figure 5: BREbias based on project complexity

Table 6: BREbias based on project complexity

Level	*N*	*Mean*	*Median*
High	5	0.22	0.21
Medium	13	0.29	0.23

From the results, there appear to be no large differences between the groups based on perceived complexity.

Technical knowledge is also somewhat subjective, but we are fairly confident that the respondents accurately can categorize their technical knowledge on a project as *good*, *okay*, or *bad*. We interpreted "technical knowledge" as the contractor's previous technical knowledge with respect to the type of system being developed.

BREbias based on complexity is presented in Figure 6 and Table 7.

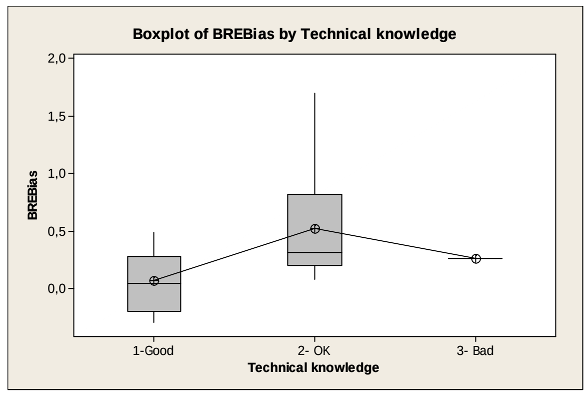

Figure 6: BREbias based on technical knowledge

Table 7: BREbias based on technical knowledge

Level	*N*	*Mean*	*Median*
Good	9	0.04	0.04
OK	8	0.52	0.31
Bad	1	0.26	0.26

There appear to be smaller overruns in the projects where the respondents had good technical knowledge.

It would seem that relatively large projects in which good technological familiarity is absent may be related to larger overruns. It is therefore necessary to explore these factors further in the discussion.

6. Discussion

In a coarse-grained categorization, we can say that overruns in software projects can be caused by characteristics related to the following factors:

1. Development organization properties
2. Customer properties
3. Properties that depend on the interaction between the developing organization and customers

Most previous research on estimation has focused on category 1. Here, we have tried to explore issues related to category 2 and 3.

6.1. Communication frequency

Daily frequency of communication was associated with fairly small overruns in our projects, and the difference from the projects that did not have daily communication has to be considered large and significant. However, there may be confounding factors that influenced these results.

Project size, measured in actual effort, based on the communication frequency, is presented in Table 8.

Table 8: Actual effort grouped by communication frequency

Level	*N*	*Mean*	*Median*
Daily	11	6623	2700
Not daily	7	3966	1830

The table shows that project size was larger for the group with daily communication with the customer.

Technical knowledge based on the communication frequency is presented in Table 9.

Table 9: Technical knowledge grouped by communication frequency

Level	*N*	*Good*	*OK*	*Bad*
Daily	11	7	4	0
Not daily	7	2	4	1
Sum	18	9	8	1

There is an overrepresentation of favorable technical knowledge in the projects with daily communication with the customer. This implies that some of the estimation accuracy gains in projects in which developers and customer communicated daily could actually be attributed to the technical knowledge, and not to the frequency of communication. However, the possible advantage of better technical knowledge is offset by the larger size of the projects in which developers and customer communicated daily.

However, frequent communication may only be a symptom of, and not a reason for, a good and trusting relationship. On the flip side, one may also surmise that in a project that has gone off-course, the customers would communicate frequently with the contractors. Nonetheless, this provides some indications that frequent communication with the customer may help to avoid overruns.

Good collaboration/communication is a frequent top-listed property that managers point to when explaining project-success, e.g. [2, 9]. It is likely that frequent communication helps to build trust and resolve differences, and helps the developers to focus on solving the customers' problems, instead of wasting effort on developing functionality that is neither required nor correct. This should apply whether the customer is in-house or external.

6.2. Type of contract

There were three different main types of contracts in our sample, and thus it is difficult to draw any clear conclusions. Some key points were that, not surprisingly, contracts that were paid for by the hour had the largest overruns. This type of contract may often add incentives for gold-plating by the contractor, frequently in the form of unnecessary functionality for the customers.

Fixed-price contracts had smaller overruns, probably because overruns are paid for by the contractor.

On the other hand, target-price contracts, which involves risk-sharing, and technically should foster collaboration, fared even better in terms of small overruns.

However, confounding factors may have an impact on this issue. Project size, measured in actual effort, based on contract form, is displayed in Table 10.

Table 10: Actual effort grouped by contract form

Level	*N*	*Mean*	*Median*
By the hour	4	2693	2277
Fixed price	5	11235	12500
Target price	7	4288	1830
Other	2	1826	1826

The table shows a substantial variance in size for the different contract forms. It may be particularly relevant to note the difference in size between fixed-price and target-price contracts.

Technical knowledge based on the type of contract is displayed in Table 11.

Table 11: Technical knowledge grouped by contract form

Level	*N*	*Good*	*Ok*	*Bad*
By the hour	4	3	1	0
Fixed price	5	2	3	0
Other	2	1	0	1
Target price	7	5	2	0
Sum	18	11	6	1

The table shows a slight variance in technical knowledge for the different types of contract. Again, it may be especially relevant to note the difference in technical knowledge between fixed-price and target-price contracts.

It might be that contractors seek to use a target price only when they believe that they have the technical knowledge. At the same time, customers may seek a fixed-price contract in order to free themselves of risk in large, complex projects.

In recent years, risk-sharing contracts have been increasingly popular in Norway. However, we are not aware of any previous studies that systematically evaluate their possible impact on overruns and project success. This, together with our small sample and possible confounding factors, makes these results tentative and so they should only be applied as a stepping stone for further research.

6.3. Customer procurement capability

We did not find any large differences in effort overruns based on customer procurement capability regarding different factors. This may be due to the following:

1. Interviews are not an accurate instrument for measuring such properties, e.g., the managers are not able to assess customer IT skill accurately.
2. Over a certain threshold, customer competence only has a small effect, as we observed. Only in clear cases of incompetence do customers severely affect overruns. Overall, the customers in this sample received favorable ratings.
3. These factors are, in general, not important for project execution.

Previously, unsuccessful projects have frequently been attributed to customer incompetence, while successful projects have been attributed to good performance by the contractors [9].

This study paint a somewhat different picture, and it appears, as previously found [28], that who and how you ask affects the responses you get. When asked to explain overruns in free-test responses, it might just be that respondents are too eager to attribute them to customers [9].

6.4. Threats to validity

In general, the sample size of this study is too small for statistical analysis, and should be used with caution. As in a previous related study [2], it was a priority to collect in-depth, high-quality data on a wide range of factors, instead of using mailed questionnaires. The work presented herein must be seen as a starting point for future research, and also as a framework for how different companies can analyze the performance of their projects.

Regarding internal validity, one must also be aware that some of the measures, e.g. customer competence, are subjective perceptions of the respondents, and not objective facts. This has been mitigated by having respondents rate their customers on a predefined scale.

In a study like this, which is not a controlled experiment, relationships of cause and effect are impossible to pinpoint. In addition, in accordance with the guidelines presented in [27], one must be aware of possible confounding factors. Nonetheless, if the results from the objectively gathered data and previous studies are combined, tendencies can be seen.

Regarding external validity, the size of the overruns is similar to those in previous surveys and case studies, which indicates that the sample was not particularly biased. However, in other environments, such as in-house development, other factors need to be taken into account. Therefore, the external validity of the study is limited to contractors developing projects of a similar size and complexity.

7. Summary

The main finding was that good collaboration with customers, facilitated by frequent communication, was associated with projects that experienced a lesser magnitude of effort overruns. It might also be that risk-sharing contracts can reduce overruns, but one must be aware of possible confounding factors with respect to this issue.

In addition, it was not observed a clear relationship between customer procurement capability and magnitude of overruns. This might be due to the small sample size, but it may also be that the results constitute evidence that this frequent explanation of overruns is somewhat exaggerated.

Taken together, the findings presented in this paper indicate that contractors can implement a few key practices to facilitate collaboration with their customers in order to reduce overruns and achieve greater project success.

These results should, perhaps, not be surprising to many in the agile community, given the commitment to *customer collaboration over contract negotiation.*

In addition to what has been observed related to reducing overruns, risk-sharing contracts and frequent communication may also have benefits that extend beyond the current project. These benefits include fostering a contractor/customer relationship that is beneficial in the long-run and that will bring returning business.

Future research should address the need for an extended set of project data for analysis, and should further seek to explore subtopics, such as different types and lengths of communication between customers and contractors, and various ways to introduce risk-sharing contracts.

Acknowledgements

This research was funded by the Research Council of Norway under the project INCO. We thank the studied company for providing us with data and access to their managers. Also thanks to Mike Cohn and Chris Wright for valuable comments.

References

[1] K. Moløkken-Østvold and M. Jørgensen, "A Review of Surveys on Software Effort Estimation," in *2003 ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2003)*, Frascati, Monte Porzio Catone (RM), ITALY, 2003, pp. 220-230.

[2] K. Moløkken-Østvold and M. Jørgensen, "A Comparison of Software Project Overruns – Flexible vs. Sequential Development Models," *IEEE Transactions on Software Engineering,* vol. 31, pp. 754-766, 2005.

[3] K. Beck and M. Fowler, *Planning Extreme Programming*: Addison-Wesley, 2001.

[4] K. Schwaber, *Agile Project Management with Scrum*: Microsoft Press, 2004.

[5] M. Cohn, *Agile Estimating and Planning*: Addison-Wesley, 2005.

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

[7] H. D. Mills, "Software Development," *IEEE Transactions on Software Engineering,* vol. 2, pp. 265-273, 1976.

[8] A. Cockburn, "The End of Software Engineering and the Start of Economic-Cooperative Gaming," *ComSIS,* vol. 1, 2004.

[9] S. Grimstad, M. Jørgensen, and K. Moløkken-Østvold, "The Clients' Impact on Effort Estimation Accuracy in Software Development Projects," in *11th IEEE International Software Metrics Symposium (METRICS 2005),* Como, Italy, September 19-22., 2005.

[10] J. M. Verner, S. P. Overmyer, and K. W. McCain, "In the 25 years since The Mythical Man-Month what have we learned about project management?," *Information and Software Technology,* vol. 41, pp. 1021-1026, 1999.

[11] J. D. Procaccino, J. M. Verner, S. P. Overmyer, and M. E. Darter, "Case study: factors for early prediction of software development success," *Information and Software Technology,* vol. 44, pp. 53-62, 2002.

[12] A. L. Lederer and J. Prasad, "Cause of inaccurate software development cost estimates," *Journal of systems and software,* vol. 31, pp. 125-134, 1995.

[13] K. Moløkken-Østvold, M. Jørgensen, P. Sørgaard, and S. Grimstad, " Management of Public Software Projects: Avoiding Overruns," in *Hawaiian International Conference on Business*, Hawaii, USA., 2005.

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

[15] D. Jamieson, K. Vinsen, and G. Callender, "Agile Procurement: New Acquisition Approach to Agile Software Development," in *EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO-SEAA '05)*, 2005, pp. 266-273.

[16] D. Jamieson, K. Vinsen, and G. Callender, "Agile Procurement and Dynamic Value for Money to Facilitate Agile Software Projects," in *EUROMICRO Conference on Software Engineering and Advanced Applications, 2006. SEAA '06.*, 2006, pp. 248-255.

[17] A. L. Lederer and J. Prasad, "Nine management guidelines for better cost estimating," *Communications of the ACM,* vol. 35, pp. 51-59, Feb. 1992 1992.

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

[19] K. Moløkken-Østvold and N. C. Haugen, "Combining Estimates with Planning Poker – An Empirical Study," in *To appear in 18th*

Australian Conference on Software Engineering, Melbourne, Australia, 2007.

[20] M. Jørgensen and D. Sjøberg, "An effort prediction interval approach based on the empirical distribution of previous estimation accuracy," *Journal of Information and Software Technology*, vol. 45, pp. 123-136, 2003.

[21] Y. Miyazaki, A. Takanou, H. Nozaki, N. Nakagawa, and K. Okada, "Method to estimate parameter values in software prediction models.," *Information and Software Technology*, vol. 33, pp. 239-243, 1991.

[22] L. C. Briand and I. Wieczorek, "Resource modeling in software engineering," in *Encyclopedia of Software Engineering*, J. Marciniak, Ed.: Wiley, 2001.

[23] E. Stensrud, T. Foss, B. Kitchenham, and I. Myrtveit, "An empirical validation of the relationship between the magnitude of relative error and project size," in *Eighth IEEE Symposium on Software Metrics.*, 2002, pp. 3-12.

[24] J. Cohen, *Statistical power analysis for the behavioral sciences*. New York: Academic Press, Inc., 1969.

[25] T. H. Wonnacott and R. J. Wonnacott, *Introductory statistics*, 5th ed.: John Wiley & Sons, Inc., 1990.

[26] K. Moløkken-Østvold, M. Jørgensen, S. S. Tanilkan, H. Gallis, A. C. Lien, and S. E. Hove, "A Survey on Effort Estimation in Norwegian Software Industry," in *10th International Symposium on Software Metrics*, Chicago, Illinois, USA, 2004, pp. 208-219.

[27] S. Grimstad and M. Jørgensen, "A Framework for the Analysis of Software Cost," in *ISESE 2006*, Rio de Janeiro, Brazil, 2006, pp. 58–65.

[28] M. Jørgensen and K. Moløkken-Østvold, "Understanding Reasons for Errors in Software Effort Estimates," *IEEE Transactions on Software Engineering.*, vol. 30, pp. 993-1007, 2004.

Appendix 1: Key project data

ID	Est. effort	Actual effort	BREBias	Communication frequency	Contract form	Customer score	Project complexity	Tech. knowledge
1	5660	7301	0.290	Not daily	Target price	13	Medium	Good
2	1810	1570.5	-0.152	Daily	Target price	4	Medium	Good
3	12000	14500	0.208	Daily	Target price	12	High	Good
4	990	1252	0.265	Not daily	Other	8	Medium	Bad
5	9900	12600	0.273	Daily	Fixed price	10	High	OK
6	5600	11000	0.964	Not daily	Fixed price	12	Medium	OK
7	2400	2400	0.000	Daily	Other	13	Medium	Good
8	12000	12500	0.042	Daily	Fixed price	7	High	Good
9	3430	4076	0.188	Daily	Fixed price	11	Medium	OK
10	3135	2507	-0.250	Daily	By the hour	5	Medium	Good
11	700	947	0.353	Not daily	Target price	11	Medium	Good
12	13500	16000	0.185	Daily	Fixed price	9	High	Good
13	1582	4268	1.698	Not daily	By the hour	7	Medium	OK
14	953	1164	0.221	Not daily	Target price	7	Medium	OK
15	1695	1830	0.080	Not daily	Target price	7	Medium	OK
16	1400	1950	0.393	Daily	By the hour	7	High	OK
17	3500	2700	-0.296	Daily	Target price	8	Medium	Good
18	1511	2047	0.355	Daily	By the hour	6	Medium	OK