

The Clients' Impact on Effort Estimation Accuracy in Software Development Projects

Stein Grimstad, Magne Jørgensen and Kjetil Moløkken-Østvold
Simula Research Laboratory
P.O. Box 134
NO-1325 Lysaker, Norway
Telephone: +47 67 82 82 00
Fax: +47 67 82 82 01
{steingr,magnej,kjetilmo}@simula.no

Abstract

It seems clear that there is no simple solution to improved predictability of software development projects. Over several decades various aspects of software development and their relationship to estimation accuracy have been investigated. This paper focus on one such relationship; the clients' impact on estimation accuracy in software development projects. Factors contributing to overruns as well as factors preventing overruns are investigated. Based on a literature review and a survey of 300 software professionals we find that: 1) It is software professionals' perception that clients impact estimation accuracy. Changed and new requirements are perceived as the clients' most frequent contribution to overruns, while overruns are prevented by the availability of competent clients and capable decision makers. 2) Survey results should not be used in estimation accuracy improvement initiatives without further analysis. Surveys typically identify direct and project specific causes for overruns, while substantial improvement is only possible when the underlying causes are understood.

Keywords: software estimation, survey, client-vendor relationship, overrun causes

1. Introduction

Overruns in software development projects have been a major concern for several decades. The topic of effort estimation has been given much attention in software engineering research, for instance by the development of numerous algorithmic estimation models. Still there is no evidence of improvement in effort estimation accuracy over the last 20 years. A recent review of estimation surveys [1] shows that most surveys of effort estimation performance in software development projects report average overruns of 30-40% in software development projects. Even if it is debatable whether the magnitude of these overruns constitutes a 'software crisis' [2], it is likely that organizations who improve estimation accuracy will gain a competitive advantage, i.e. as a consequence of increased predictability and more optimal resource allocation. This applies to clients as well as vendors, since disadvantages caused by overruns hit clients as well as vendors, irrespective of contract type, as discussed in [3].

One way to see software development, is as an economic cooperative game [4]. That something is an economic game means, among other things, that the result of the game depends on the behavior of the

players. A cooperative game means that the best result is achieved when the players cooperate. For software development projects, this suggests that clients, as well as vendors, influence effort estimation accuracy. Previous research supports this view. For instance, a survey of estimation overruns in software development projects showed that governmental projects on average had higher overruns of effort estimates than private projects [5]. The governmental and private projects did not differ on any of the project characteristics measured, such as development methodology, vendor capability, project size, or project duration. The only difference was the category of client.

By focusing solely on the clients' impact on estimation accuracy and by investigating both the negative and the positive impact the clients have on estimation accuracy we hope to gain a deeper understanding of this relationship. For clients such understanding is important in order to improve their acquisition processes. Besides, an understanding of their own capability as clients of software development projects might be valuable input to the selection of vendors. The understanding of clients' impact on estimation accuracy is also valuable to vendors. Realizing that the client is the cause of a troubled

project, enables appropriate actions to be taken. These might include changes to the development methodology, the project staffing, re-negotiations, juridical actions, etc.

The goal of this study is to examine 1) Do clients influence estimation accuracy? 2) How do clients impact estimation accuracy? 3) What can clients do to improve estimation accuracy? These questions are examined by reviewing software engineering literature and surveying software professionals. The paper is organized as follows: Section 2 presents a review of existing research on how (and if) customers impact estimation accuracy in software development projects. That review is the background for the study design described in Section 3. The results of the study are presented in Section 4, while Section 5 discusses the results of the study and the findings in the review with respect to the objectives of the study. Section 6 concludes the paper.

2. Review of Related Work

2.1 Review design

The aim of the review is to identify available research relevant to the research questions presented in Section 1 in an unbiased and auditable manner. The design of the review is based on the guidelines for structural reviews in software engineering proposed by Kitchenham [6]. The review is limited to what we consider the most important journals for software engineering; Information and Software Technology, Journal of Systems and Software, IEEE Transactions of Software Engineering and ACM Transactions on Software Engineering and Methodology. Papers were selected for inclusion by manually going through the online indexes of these journals, reading the abstracts that appeared relevant and finally the full versions of the remaining papers. The review was done by one of the researchers. For each included paper, the following information were extracted: the bibliographic information, study type, goals of the study, sample size and the client's impact on estimation accuracy. The criteria used to decide whether to include or exclude papers were:

- the papers has to report causes or risk factors for estimation accuracy/performance/overruns or project success/failure (when success/failure was related to schedule/budget or estimates)
- the papers has to report empirical evidence from software development projects
- the papers has to identify general reasons (as opposed to investigate a specific issue)
- if two studies are based on the same data set, only one were included
- the papers has to be available online

There are a number of limitations to this review. Among the most important is that the survey is based on a limited selection of journals. We are aware of other journal papers, books, conference papers and industry reports that have made important contributions to the topic such as Brooks' classic *The Mythical Man-Month* [7]. We have chosen this approach in an attempt to get an unbiased sample of material. There is also a risk that papers from the selected journals have been wrongly excluded, i.e. that we were not able to recognize the relevance of the paper by the name and/or the abstract. Some verification was done by checking against a previous review [8], and we had at least captured the papers from the selected journals listed there.

Another risk is the subjective identification of which factors are client influenced. This task is hard as the categories/terms/reasons are most often only explained in broad terms. We have not found any good way to address this risk other than to rely on our experience as researchers and software professionals. There is also the problem that papers built on considerable experience are excluded from this review as they do not directly report empirical evidence. However, inclusion of such articles would lower the quality of the review, as it would be even more subjectively decided which articles to include and which to exclude.

2.2 Review results

2.2.1 Verner, et.al. [9]

Verner, et.al. [9] conducted structured interviews with 20 senior software development professionals from a number of different organizations in the USA. The goal of the study was to compare the software project management advices given in Brooks' famous book *The Mythical Man-Month* with practices employed 25 years later. For each topic discussed, the factors leading to project success were identified, and then how the same factors could contribute to failure. The client related success factors found, and their frequency, are: High level management support (about 50%), Customer and user involvement (15%), Good requirements (nearly 50%), Flexibility (frequency not available) and Communication (67%). The corresponding failure factors are: Lack of higher level management support (almost all), Lack of involvement/confidence and too many customers involved (nearly 50%), Vague/poor requirements and no clear vision (40%), Poor estimates made by management and dictated dates (50%) and Feature and scope creep ("many").

2.2.2 Procaccino et.al. [10]

Procaccino et.al. [10] had 21 IT professionals reflecting 42 software development projects complete two questionnaires (failed/successful projects). The goal of the study was to investigate some of the most influential success factors early in the development process. The respondents were project leaders, technical support personal and developers. All were from the same organization. The respondents were asked what they perceived as success factors, and what they thought management considered as success factors (there were differences). The significant success factors, when applying Chi-square tests, were: Presence of a committed IT sponsor, Customer/user's involvement/commitment/confidence in the project, Customer/users involvement in schedule estimation, Customer/users had realistic expectations, Establishment of complete and accurate requirements and Customers/users allocated adequate time for requirement gathering.

2.2.3 Lederer and Prasad [11]

Lederer and Prasad [11] used a questionnaire to have 112 systems managers and other information systems professionals rate 24 predefined reasons for overruns. The customer factors perceived as important out of the ten most important are: Change requests by users, Users' understanding of requirements, User-analyst communication and understanding, Poor or imprecise problem definition and Coordination of company functions during development. The causes where correlated with the organization's percentage of inaccurate estimates by use of the Pearson r coefficient correlation. The customer factors of the top ten most statistic significant reasons are: Reviewers don't consider whether estimates are met, Lack of careful examination of the estimate by management and Poor or imprecise problem definition.

2.2.4 van Genuchten [12]

van Genuchten [12] performed weekly interviews with project leaders and collected data on activity level for six software development projects in the same software development department in order to gain an insight in the reasons for delays of software development projects. The most frequent reasons for delays were "more time spent on other work than planned" (mainly as a result of maintenance tasks) and "complexity of application underestimated". None of these is customer related. Measurements in other departments revealed that distribution of causes varies strongly for each department.

2.2.5 Jiang and Klein [13]

Jiang and Klein [13] had 86 members of Project Management Institute complete a questionnaire to test

the linkage between previously identified software development risks and various dimensions of system success. Only two of the risks had a significant impact on "meet budgets" and "meet schedules": "Lack of teams general expertise" (including the ability to work with uncertain objectives, ability to work with top management and ability to understand human implications of a new system) and "Lack of role clarity" (including the role of each person involved in the project is not clearly defined and communications between those involved in the project are unpleasant).

2.2.6 Ropponen and Lyytinen [14]

Ropponen and Lyytinen [14] surveyed 83 project managers from the Finnish Information Processing Association (at most two respondents from each company) using a questionnaire. The goal of the study was to investigate the impact of risk management practices on software development. Schedule and timing risks were influenced by the following client related factors: the project size (larger projects performed worse than smaller), the client's industry (retail business, accommodation, nutrition performed better than other industries) and the application type (interactive applications performed worse than other).

2.2.7 Jørgensen and Moløkken-Østvold [8]

Jørgensen and Moløkken-Østvold collected data from estimation experience reports of 68 projects and interviewed eight employees (in different roles) in a Norwegian software company. The goal of the study was to understand how roles, information collection approaches and analysis techniques supplement each other when examining reasons for errors in software effort estimates. The reasons for estimation error mentioned in the interviews were: Lack of realism in HCI-requirements, Lack of requirement change control processes, Unrealistic expectations by clients and Lack of good requirement specifications leading to unplanned re-work. Table 1 shows the reasons for estimation (in)accuracy found in the experience reports.

Table 1 Reasons for estimation accuracy

Experience report-based reasons for inaccurate estimates	Experience report-based reasons for accurate estimates
Change requests from clients or "functionality creep"	Simple projects
Resource allocation problems	A high degree of flexibility in how to implement the requirement specification
Poor requirement specifications or problems with communication with the client	
High priority on quality, cost accuracy not of high importance	

In addition, the study contains a statistical analysis (stepwise regression) of some project characteristics and how they relate to estimation accuracy. The only significant client based factor was the client's priority of time-to-delivery.

2.2.8 Subramanian and Breslawski [15]

Subramanian and Breslawski [15] had 40 members of the ACM Special Interest Group on Software Engineering responded to a mail questionnaire. The study seeks, among other things, to explain the percentage of relative error in software effort estimation. The customer influenced reasons for failure were: "requirements change/addition/definition", "design changes, scope and complexity", "upper management influence, bidding and time constraints"

2.2.9 Summary

None of the studies identified in this review explicitly investigated client influence on estimation accuracy in software development projects, but almost all (seven out of eight) of the studies report that client factors are perceived as important for estimation accuracy. This strongly suggests that clients influence estimation accuracy. Factors related to management, communication and involvement in the project along with factors related to requirements and realistic expectations are the most frequent reasons perceived as impacting estimation accuracy that can be attributed to clients. Other client related factors, such as project size, the industry of the client, application type and flexibility are also reported, but less often.

The studies further report that vendors are more likely to attribute failure than success to clients [9] and that who you ask (developers, project managers, management, etc) influence the perception of the clients' impact on estimation accuracy [8][10]. Also, statistically analysis of factors give different results than surveys and interviews of what reasons are perceived as most important [8, 11]. One of the papers finds reasons for estimation accuracy to be largely project specific. When the results of a case study conducted in one department [12] was compared to results from another department within the same organization, there were significant differences.

However, it is not clear what clients should do to improve estimation accuracy in software development projects. Our study, described in the remaining sections of this paper, explicitly investigates the clients' contribution to estimation accuracy. The aim is to get more data regarding these phenomena so that we can do a more thorough analysis and better understand them.

3. Survey Design

To investigate our research questions, we conducted a survey at a technical conference. Sections 3.1-3.3 describes the design of the schema and the sample, while sections 3.4 and 3.5 comment the data collection and the analysis.

3.1 General

The findings in some of the studies in Section 2, along with our previous research and industry experience, determined the content in the survey instrument we used to investigate the research questions described in Section 1. The survey collected three types of data; 1) context information about the respondents (such as estimation experience and project roles) 2) the respondents' perception of clients' impact on estimation accuracy (positive and negative) and 3) the estimation performance in the respondents' last completed project along with their rating of a predefined set of client factors.

3.2 Schema design

Four software professionals validated the schema in a pilot study. This led to clarification of some of the questions. The final schema consisted of 20 multiple choice questions and two open-ended questions. The schema was written in Norwegian and the data are later translated by us. The motivation for using Norwegian, and thus introducing the risk of translation errors, is that we believe this would lower the burden of participation and therefore increase the number of respondents. This approach would only exclude the non-Scandinavian speakers since Norwegian is quite similar to Swedish and Danish. The schema took approximately 10 minutes to complete. To motivate participation, the respondents could win prizes in a lottery. In order to participate in the lottery, the respondents had to write their email address, but participation in the lottery and hence writing the email address was voluntarily. However, all respondents chose to write their e-mail address. A threat to this form of data collection is the risk for misunderstandings, but a strength of using a survey instrument, compared to interviews, is that the number of data points are increased.

The respondents decided themselves how to interpret "estimate", "overruns of estimates" and "no, or small, overrun of estimates". Previous studies has shown that the estimation terminology in use is ambiguous, i.e. the term estimate is frequently used for most-likely estimates, estimates of budgets and price to customer [16]. In this study we have chosen to use broad terms

Table 2 Demographic data

Question	Category	Result	Comment
How many projects have you been involved in estimating?	None 1 – 4 5 – 20 20 +	27 (9,0%) 113 (37,7%) 118 (39,3%) 42 (14%)	N = 300.
Has the projects you have been involved estimating been for an internal or an external client?	Not estimated Internal External Both internal and external Other	25 (8,3%) 74 (24,7%) 126 (42%) 75 (25%) 0 (0%)	N = 300. Nine respondents gave multiple answers. These are interpreted as "equally many". One respondent did not answer the question, but had answered "no estimation experience" on a previous question. This is interpreted as "has not estimated".
What sector have you primarily been estimating for?	Not estimated Governmental Telecom Financial Industry Other	26 (8,7%) 93 (31,1%) 96 (32,1%) 97 (32,4%) 46 (15,4%) 43 (14,4%)	N = 299. Each respondent were allowed to select several industries. The "other" category was mainly retail. 222 respondents (74,2%) had estimated for more than one sector.
What has been your role(s) in the above mentioned projects?	Developer Architect Project manager Other	271 (91,8%) 141 (48,0%) 62 (21,0%) 6 (2%)	N = 295. Each respondent was allowed to select several roles. All the non-respondents replied "not estimated" on the questions above.
In your latest completed project, how large was the overrun of estimates?	None 0 – 20% 21 – 50% Above 50% Do not know	57 (19,2%) 133 (44,8%) 55 (18,5%) 18 (6,1%) 34 (11,5%)	N = 297. 80% of the projects had overrun of estimates which is similar to other estimation surveys [1].
In your latest completed project, what was the budget (approximately)?	Do not know > 100.000 100.000 – 1 million 1 – 10 millions Above 10 millions	40 (13,4%) 28 (9,4%) 86 (28,9%) 102 (34,2%) 42 (14,1%)	N = 298. The figures are in Norwegian Kroner (NOK). 1 US Dollar is approximately 6,3 NOK.

such as "overruns" and "estimate". This is because we did not want to confuse the participant with unfamiliar terms, and since we believe that the reasons for overruns are the same for the different meanings of the term "estimate". To measure the magnitude of overrun in their last completed projects we used broad categories that were further combined in the analysis. The most recent project, instead of "average values" or a "typical project", was selected to reduce the influence of poor memory and selection bias.

3.3 Sample

The survey was conducted at the JavaZone 2004 conference in Oslo, Norway. The conference was arranged by the Norwegian java user group (javaBin). JavaZone targets Scandinavian professionals with an interest in Java technology, and is one of the leading technical conferences in Scandinavia. 800 persons registered for the conference including speakers, journalists, arrangement committee and expo personnel. 307 of these participated in the survey. Table 2 presents some demographic data collected. The first column

states the questions the respondents answered, the second shows the categories the respondents could choose from, the third column presents the frequency of each category and the percentage of the total (excluding non respondents) while the last column contains comments. The data suggest that the respondents have a technical focus (91,5% has programmed in at least one project), they are experienced estimators and that they are distributed across industries. It is also reasonable to believe that the attendants are above average interested in software development (since they attend the conference), and that the general level of competence is high.

3.4 Data Collection

The survey was handed out from the organizers' stand in the conference expo area. It was labeled as a joint effort between the University of Oslo and the Norwegian java user group. A research assistant distributed and collected the schemas, and the survey was open for participation from the start of the conference till the social program started, a total of 12 hours. About 400 schemas were handed out and 307 of these were returned.

We were unable to collect reasons for non-participation in a systematic way, as the organizers declined our request to include questions regarding survey participation in the official conference evaluation form. To collect data on non-participation, we asked conference attendees during the social program about their participation. Three reasons for non-participation dominated; the conference program was too packed so they did not prioritize participation in the survey, they did not visit the expo area and therefore were not aware of the survey or they did not feel qualified to answer (marketing staff, journalists, etc). This means that there does not appear to be any harmful bias regarding which conference attendees participated in the survey as would have been the case if for instance one company refused to let its employees participate or if people were ashamed to participate because they had a bad track record for estimation performance.

3.5 Analysis

The data were analyzed and structured using Excel and Minitab. Seven out of the schemas were excluded from the analysis because we regarded them to be of insufficient quality (primarily because they were mostly blank). Preliminary results from the study were verified at two of the monthly member meetings in the java user group. The attendants at the meetings agreed with the interpretations.

The data regarding participants' perception of the customers' influence on estimation accuracy were grouped in categories. The categories are derived by analyzing the answers and joining answers into groups until we had a sufficiently small number of categories. The categorization was done independently by one of the authors and an experienced software professional. Disagreements were discussed. In the few cases where we did not reach an agreement, we included both categories. For the completed projects, overrun magnitude were collected by asking the respondents to choose the category that best fitted their project. They could choose from: "no overrun", "0-20%", "21-50%", "more than 50%" and "do not know". In the analysis, "no overrun" and "0-20%" is merged together as "< 20% overrun", while "21-50%" and "more than 50%" is merged together as "> 20% overrun". 73 of the respondents were in the category "> 20% overrun" and 189 in the category "< 20% overrun".

To preserve anonymity, we did not ask for any form for project identification. This is unlikely to be a problem for the part of the survey that is concerned with the respondents perception of factors that influence estimation accuracy, as this address the respondents opinions based on their total experience. However, for the rating of factors in the last completed project, we do not know how many projects (experimental units) the

respondents represent. The data suggests that the distribution is fairly good (based on combining estimation accuracy, size of project and type of customer), but it is likely that at least some projects are overrepresented.

4. Results

4.1 Software professionals' perception of client factors impacting estimation accuracy

This section presents the respondents perception of customer factors that frequently caused overruns and customer factors that prevented overruns. The responses were asked (translated):

Q1: "In projects where estimates have been overrun, what factors of the clients has contributed to the overrun?"

Q2: "In the projects where estimates were not overrun, or there were only minor overruns, what factors of the client contributed to prevention of overrun?"

The responses were grouped in categories, see table 3, and are presented in table 4.

Table 3 Categories of perceived reasons

Code	Category
STAB	Requirement changes and new requirements
REQU	Well defined requirements
COMM	Client – vendor communication
FLEX	Flexibility in the project (give and take)
TECH	Integration with technical environment, infrastructure and development environment
SUBC	Integration and co-operation with 3 rd party vendors
SIZE	Project size
REAL	Realistic expectations (requirements, time, budget, etc)
SKILL	Availability of competent customers and capable decision makers
PROJ	Project administration and steering
OTHE	Other reasons that we were not able to classify

Table 4 Customer factors impacting estimation accuracy

Factor type	Perceived as causing overruns by	Perceived as preventing overruns by
STAB	118	23
REQU	97	50
COMM	13	34
FLEX	9	28
TECH	17	3
SUBC	4	2
SIZE	0	4
REAL	15	13
SKILL	70	76
PROJ	38	50
OTHE	14	18

Table 5 Factors tested for correlation with overruns

Code	Factor	Statement type	Translated statement/question
METH	Project Methodology	Neutral	The project used an incremental/iterative development method.
REAL	Realism in plans and budgets	Positive	The project had realistic plans and budgets
GOAL	Clear project goals	Positive	The goals of the project were clearly defined and communicated.
PRIO	Client's priority of the project	Positive	The project had high priority in the client organization.
RESO	Client's resource allocation	Positive	The client had allocated sufficient resources for an efficient project execution (test environment, end-users, etc).
SKILL	Client skills	Positive	The clients had the right skills for an efficient project execution.
COMM	Client and vendor communication	Positive	The communication between client and vendor were adequate.
STAB	Scope creep	Negate	The requirement specification were frequently expanded.
FLEX	Project flexibility	Positive	The project had the flexibility to reduce scope (functionality/quality) in order to meet plan and budget.
REWO	Client change of minds	Negative	Clarifications made by the client were later changed so that work had to be re-done or thrown away.
UNFO	Unforeseen tasks	Negative	Unforeseen tasks occurred frequently
PARA	Projects run in parallel	Negative	The project were delayed by projects running in parallel.
LUCK	Luck/bad luck	Neutral	Luck or bad luck had a significant impact on the outcome of the project.

Out of the 300 replies included in the analysis, 38 respondents had no response to Q1, while 79 respondents had no response to Q2. In the cases where the same respondents had made several answers that fell into the same category, each answer was counted.

The three reasons most frequently perceived as contributing to overruns are 1) frequent requirement changes and new requirements, 2) lack of well defined requirements and 3) lack competent customers able to make decisions. The most important reasons perceived as preventing overruns are 1) competent clients able to make decisions, 2) well defined requirements and 3) adequate project administration.

4.2 Correlation between client factors and effort overruns in the respondents' last completed project

The respondents were asked to rate their last completed project (from 1 = "totally agrees" to 5 = "totally disagrees") according to a set of predefined statements. Table 5 describes the statements the respondents rated (column four) and the associated factors the statements were intended to test (column two). The statements have a mixed framing strategy. Column three says whether the statement is positively or negatively framed. For the positively framed statements, low scores (that is, agreement) are believed to be better, while high scores (disagreement) are believed to be better for negative framed statements.

Table 6 presents the average ratings for each factor. The first column in table 6 lists the different factors' by code. The two next columns show the estimation performance (divided into above or below 20% overrun), while the last column shows the

difference between the groups. The average is calculated after the responses in "Don't know" and "No response" were removed. 21 was the highest count removed for any factor.

The projects with large overruns differ most from the projects less overruns for 1) realism in plans and budgets, 2) project flexibility and 3) client and vendor communication. The clients' priority of the project is the only factor where there is almost no difference.

Table 6 Factors correlated to estimation performance

Project outcome	More than 20% overrun	Less than 20% overrun	Difference
METH	3,58	3,90	-0,32
REAL	2,74	3,99	-1,25
GOAL	3,46	3,89	-0,43
PRIO	4,24	4,25	-0,01
RESO	3,92	3,42	0,50
SKILL	3,26	3,55	-0,29
COMM	3,34	3,85	-0,51
STAB	4,25	3,77	0,48
FLEX	3,07	3,68	-0,61
REWO	3,15	2,63	0,52
UNFO	3,82	3,37	0,45
PARA	3,15	2,89	0,26
LUCK	2,44	2,03	0,41

5. Discussion of Results

5.1 Do clients influence estimation accuracy?

252 out of 300 software professionals (84%) mention at least one client factor that they consider to be a major contribution to estimation overrun. Similar, 211 out of 300 (70%) listed one or more client factors contributing to prevention of overruns. Also, the respondents'

projects with the largest overruns have worse average rating for all the rated factors (on their last completed projects), with the exception of "Client's priority of the project" which is virtually identical for both categories of projects. The importance of the client impact on estimation accuracy is further supported by related work (see Section 2.2.9).

Still, this does not prove that clients impact estimation accuracy. The results reported in this study, and in related work, are the perceptions of the respondents, and not objective facts. There might be a number of reasons why respondents blame clients for overruns (some of these reasons are discussed in the following section). However, the results strongly suggests that clients are perceived as impacting estimation accuracy by vendors.

5.2 How do clients impact estimation accuracy?

The factors perceived as influencing estimation accuracy in this survey correspond well to factors reported in earlier surveys in the sense that at least one other study mention a category similar to each category identified in this survey as a major contributor to effort overruns (see Section 2.2.9). Still, even if there is a great deal of overlap between the surveys, none of them have provided the exact same list of factors causing overruns. These differences can probably be explained by the different focuses of the studies, terminology ambiguousness, roles of the respondents, the method of analysis and the size differences in the samples.

Almost all of the customer factors perceived by the software professionals as influencing estimation accuracy are project specific and direct causes, and little focus is on more general and underlying causes (see table 4). For instance, "requirement changes and new requirements" is perceived as important, but less focus is on why scope is increased (for instance, "business changes") or why this caused an overrun (for instance, "the project buffer was too small"). This is consistent with studies by Procaccino et.al. [10] and Jørgensen and Moløkken-Østfold [8]. Both these studies have observed that project participants are likely to provide direct reasons for estimation inaccuracy, and that managers might provide different reasons. Even if the underlying reasons might be most valuable, it is unrealistic to expect that the participants in a survey should be aware of such reasons (for instance, insufficient learning) in the client organization.

The perception of the factors that cause overruns correspond well to the perception of factors that prevent overruns in the sense that there are matching counterparts (see table 4). The only difference is the relatively small success factor "project size" which has no apparent counterpart in the factors causing overruns. However, there are some substantial

differences in the weights the factors are given. While "requirement changes and new requirements" is perceived as the most frequent contributor to overruns, only 23 mentions this as a factor that prevents overruns. Similarly, 97 mentions (lack of) "well defined requirements" as a factor causing failure, while only 50 mention it as a reason for success. The most important success factor ("availability of competent customers and capable decision makers ") mentioned rank third of the factors causing failure. A possible explanation for these differences is the human characteristic of over-optimism ("we made the project a success" while "they made the project a failure"). In Verner, et. al. [9] more than half of the projects blamed customers for overruns, while only a small fragment credited the customer for success. Another explanation is that it might be easier to note the presence of something than the absence.

Even if most of the perceived reasons for overruns are not directly tested in the correlation of factors to actual estimation performance (see table 6), the results correspond in the sense that the perceived reasons for overruns have related factors that rate worse for the projects with bad estimation performance. However, the frequency of the perceived factors does not correspond to the impact of the factors in the rating. The most frequent factor perceived as causing overruns is "requirement changes and new requirements". Even though the corresponding factor rates worse for projects with overruns than for projects without, the rating is no worse than for other factors corresponding to reasons that are much less frequently mentioned as contributing to success or overruns. This effect is also found in Lederer and Prasad [11] and in Jørgensen and Moløkken-Østfold [8].

It seems that clients are perceived as impacting estimation accuracy in many ways. However, the results obtained in surveys will vary according to several factors. The questions you ask (overrun prevention vs. contribution), who you ask (developers, managers, etc) and the method of analysis applied (statistical analysis vs. respondents' perceptions). Still, there seems to be a limited number of symptoms of troubled projects that point to customer specific causes. These symptoms are : a) increased scope b) increased complexity c) waiting d) lack of control e) re-work f) lack of incentives.

5.3 What can clients do to improve estimation accuracy?

Before any action is taken to change client behavior to improve estimation performance, it should be investigated whether there actually is an estimation accuracy problem that need to be solved. For instance, this is not the case if estimation inaccuracy was caused by a deliberate choice (i.e. extended functionality to increase return of investment) or if it was caused by

known events that are unlikely to re-occur (i.e. organizational changes). However, in most cases, we believe, estimation inaccuracy is undesirable. If analysis shows that the troubled projects have been suffering from any of the symptoms in section 5.2, clients might be contributing to the inaccuracy. But the solution is rarely as simple as that the client should just stop doing whatever actions leading to the causes identified. In order to obtain sustainable improvements, we need to identify the *underlying causes* [17]. Example 1 illustrates how one of the most frequently mentioned client reason for overrun can have several different underlying causes.

EXAMPLE 1: Let us assume that a company suffers from overruns in their software development projects, and that an analysis of the experience reports has identified that "frequent change requests during development" seems to be the most prominent factor causing problems. So, does this mean that the client should stop making change requests? In order to answer that question we need to understand why these change requests occur. Some possible answers are:

- 1) The client tries to get more functionality than originally agreed.
- 2) The original requirements were wrong and/or incomplete, so that the originally specified solution will have no or little value.
- 3) Unexpected business changes demand changes to the solution.

All the above mentioned reasons are possible causes for "frequent changes to the requirement specification". But they point to different underlying problems, and require different actions to eliminate. A possible solution for the first issue can be introduction of contracts that enforce better change management. The second and third issues require further analysis. Was it because insufficient time was spent on making the specification? Did the wrong people work on it? Was it impossible to know the requirements up front? In order to efficiently remove the problem, we need to understand the answer to these questions.

Another way to attack the problem of changing requirements would be to address the conditions that make the changes harmful. Table 6 shows that even if the projects that had large overruns generally rated worse on requirement stability, more than half of the successful projects (projects with overruns less than 20%) reported that they had unstable requirement. A previous study by one of the authors [18] has even shown that while the projects experience reports state that one of the most important reasons for overruns were incomplete requirement specifications, comparisons of the requirement specification information and the estimation precision indicated the

opposite! Therefore, understanding which circumstances make "changes to requirement specification" cause estimation overruns, is essential in order to improve the situation. Questions such as the following should be asked: Is the overruns due to lack of flexibility in the project methodology? Is the risk buffers appropriate sized for the expected amount of change in requirements?

Similar examples could easily be created for most of the factors perceived as causing overruns reported in our survey. Such examples illustrate the danger of relying on individuals' perceptions in estimation accuracy improvement initiatives. A consequence is that surveys on reasons for estimation inaccuracy/project success/overruns, etc. should solely be used as a starting point for further analysis. As discussed in section 5.2, an estimation survey does not report an objective truth, but the identified causes depends on several factors. Therefore, a first step in an estimation improvement initiative might be to investigate your own projects to determine which factors are relevant for your organization. This should be followed by an in-depth analysis, for instance by applying structured interviews and investigating project artifacts, to determine the underlying causes of the problem. It is important to note that such an analysis should investigate both the actions that cause problems, and the conditions that enable the problem to arise. Once such understanding is obtained, sustainable improvement can be made.

6. Conclusion

This study suggests that clients are perceived by software professionals as impacting estimation accuracy. In a survey of 300 software professionals, the respondents answered that the most common contributions to estimation inaccuracy by clients are frequently changing, and new, requirement along with the lack of well-defined requirements and the absence of competent customers and capable decision makers. Overruns are prevented when competent customers and capable decision makers are present, the project administration and steering is adequate and requirements are well-defined. However, the results of such surveys have limited value for an organization's estimation accuracy initiatives since the factors identified in surveys typically are project specific and "direct" causes, and not the underlying causes necessary to eliminate the problem. Also, the survey results strongly depend on a number of factors such as the data collection approach, method of analysis and the framing of the questions. Despite these limitations, surveys on factors impacting estimation accuracy might

be valuable as a starting point for more thorough analysis of factors impacting estimation accuracy.

References

1. Moløkken, K. and M. Jorgensen. *A review of software surveys on software effort estimation*. in *International Symposium on Empirical Software Engineering*. 2003. Rome, Italy: Simula Res. Lab. Lysaker Norway.
2. Glass, R., *Failure Is Looking More like Success These Days*. IEEE Software, 2002. **January/February**: p. 103-104.
3. Jørgensen, M. and S. Grimstad. *Over-optimism in Software Development Projects: "The winner's curse"*. in *CONIELECOMP*. 2005. Puebla, Mexico.
4. Cockburn, A., *The End of Software Engineering and the Start of Economic-Cooperative Gaming*. Computer Science and Information Systems, 2005. **1**(1).
5. Moløkken-Østfold, K., et al. *Management of Public Software Projects: Avoiding Overruns*. Accepted for *Hawaiian International Conference on Business*. 2005. Hawaii, USA.
6. Kitchenham, B., *Procedures for Performing Systematic Reviews*, in *Keele University Technical Report TR/SE-0401*. 2004, Keele University: Keele.
7. Brooks, F., *The Mythical Man-Month: Essays on Software Engineering*, 20th Anniversary Edition. 1995: Addison-Wesley Professional.
8. Jørgensen, M. and K. Moløkken-Østfold, *Reasons for Software Effort Estimation Error: Impact of Respondent Role, Information Collection Approach, and Data Analysis Method*. IEEE Transactions on Software Engineering, 2004. **30**(12): p. 993-1007.
9. Verner, J.M., 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, 1999. **41**: p. 1021-1026.
10. Procaccino, J.D., et al., *Case Study: factors for early prediction of software development success*. Information and Software Technology, 2002. **44**: p. 53-62.
11. Lederer, A.L. and J. Prasad, *Causes of Inaccurate Software Development Cost Estimates*. Journal of Systems and Software, 1995. **31**: p. 125-134.
12. van Genuchten, M., *Why is Software Late? An Empirical Study of Reasons for Delay in Software Development*. IEEE Transactions on Software Engineering, 1991. **17**(6).
13. Jiang, J. and G. Klein, *Software development risks to project effectiveness*. The Journal of Systems and Software, 2000. **52**: p. 3-10.
14. Ropponen, J. and K. Lyytinen, *Components of Software Development Risk: How to Address them? A project Manager Survey*. IEEE Transactions on Software Engineering, 2000. **26**(2).
15. Subramanian, G. and S. Breslawski, *An Empirical Analysis of Software Effort Estimate Alterations*. Journal of Systems and Software, 1995. **31**(2): p. 135-141.
16. Grimstad, S., M. Jørgensen, and K. Moløkken-Østfold, *Software Estimation Terminology - The Tower of Babel*. submitted to Information and Software Technology, 2005.
17. Gano, D., V. Lee, and V. Mitchell, *Apollo Root Cause Analysis - A New Way Of Thinking*. 1999: Apollonian Pubns.
18. Jørgensen, M., L. Moen, and N. Løvstad. *Combining Quantitative Software Development Cost Estimation Precision Data with Qualitative Data from Project Experience Reports at Ericsson Design Center in Norway*. in *Conference on Empirical Assessment in Software Engineering*. 2002. Keele, England: Keele University.