Jurka Rahikkala

On Top Management Support for Software Cost Estimation

TURKU CENTRE *for* COMPUTER SCIENCE

TUCS Dissertations
No 240, June 2019

ON TOP MANAGEMENT SUPPORT FOR SOFTWARE COST ESTIMATION

JURKA RAHIKKALA

To be presented, with the permission of the Faculty of Science and Engineering of the University of Turku, for public criticism in Agora lecture hall XX on June 28, 2019, at 12 noon.

University of Turku
Department of Future Technologies
Vesilinnantie 5, 20500 Turku, Finland

2019

supervisors

Prof. Ville Leppänen, Ph.D.
Department of Future Technologies
University of Turku
Finland

Asst. Prof. Sami Hyrynsalmi, D.Sc. (Tech.)
Computing Sciences, Pori
Tampere University
Finland

reviewers

Prof. Markku Oivo, Ph.D.
Department of Information Processing Science
University of Oulu
Finland

Prof. Dr. Jürgen Münch
Department of Business Informatics
Reutlingen University
Germany

opponent

Assoc. Prof. Kari Systä D.Sc. (Tech.)
Tampere University
Finland

ISBN 978-952-12-3835-2
ISSN 1239-1883

ABSTRACT

Inaccurate software cost estimates continue causing project overruns and hurting firms' economy. This thesis addresses the problem by focusing on top management role in applying estimation methodologies successfully in organisations. The research questions are 1) How does top management support software cost estimation, and 2) What are the impacts of top management support for creating a good cost estimate for a software project?

Three empirical studies, one quantitative and two qualitative, were conducted to address the research questions. The studies identified practices, through which top management is involved in cost estimation, and collected evidence on the impact of practices on estimation success. The quantitative study is based on views of 114 Finnish software professionals, and the quantitative studies are based on in-depth findings from three Finnish software producing companies and projects.

The results show that top management support for estimation is mostly indirect. Management focuses on creating a successful environment for estimation instead of hands-on participation. The key factors of top management support include adequate resources, demonstrating the importance of estimation and seeking realism. This indirect role is enough for successful estimation. On the other hand, the results provide evidence that top management may negatively impact estimation. For example, unclear expectations may cause the project team to aim for the wrong outcome, expressed expectations may bias estimation and interpreting estimates as commitments may decrease estimators' motivation and cause them to give high estimates.

The practical implication is that top management should avoid direct participation in software estimation and focus on sustaining a supportive and unbiased environment. By doing this, many projects should be able to avoid failures hurting firms' competitiveness. From the research perspective, the results provide evidence that people-related perspectives are an important factor in software estimation, implying that a shifting focus from methodologies toward managerial topics is justified.

TIIVISTELMÄ

Epätarkat ohjelmistoprojektien kustannusarviot johtavat suunnitelmien ylittymiseen ja rasittavat yritysten taloutta. Tämä väitöskirja keskittyy ylimmän johdon rooliin arviointimenetelmien menestyksekkäässä soveltamisessa organisaatioissa. Väitöskirjan tutkimuskysymykset ovat 1) kuinka ylin johto tukee ohjelmistojen kustannusarviointia ja 2) mitä vaikutuksia johdon tuella hyvän kustannusarvion laatimiseksi on ohjelmistoprojektille?

Väitöskirjan tulokset perustuvat yhteen määrälliseen ja kahteen laadulliseen tutkimukseen. Tutkimukset tunnistivat tapoja, joilla johto osallistui kustannusarviointiin sekä keräsi näyttöä osallistumiskäytänteiden vaikutuksista arvioinnin onnistumiseen. Määrällinen tutkimus pohjautuu 114 suomalaisen ohjelmistoammattilaisen näkemyksiin, kun taas laadulliset tutkimukset pohjautuvat löydöksiin kolmen suomalaisen ohjelmistoyrityksen toteuttamista kolmesta ohjelmistoprojektista.

Tulokset osoittavat, että ylimmän johdon tuki arvioinnille on pääasiallisesti epäsuoraa. Johto keskittyy hyvien edellytysten luomiseen sen sijaan, että osallistuisi arviointiin henkilökohtaisesti. Tärkeimpiin tapoihin tukea arviointia kuuluvat mm. riittävien resurssien varmistaminen ja realististen arvioiden tavoittelu. Yllä kuvattu epäsuora osallistuminen on riittävää arvioinnin onnistumiseksi. Toisaalta johdon toimet voivat myös vaikuttaa arviointiin negatiivisesti. Esimerkiksi epäselvät tavoitteet saattavat johtaa väärien asioiden tavoitteluun, johdon esittämät odotukset voivat vääristää arvioinnin tuloksia ja arvioiden tulkitseminen lupauksiksi voi laskea arvioitsijoiden motivaatiota ja johtaa perusteettoman korkeiden arvioiden antamiseen.

Esitettyjen tulosten perusteella johdon pitäisi välttää suoraa osallistumista arviointiin ja keskittyä arviointia tukevan ilmapiirin luomiseen. Näillä toimilla useat projektit voisivat todennäköisesti välttää yrityksille vahingolliset epäonnistumiset. Tutkimusnäkökulmasta tulokset osoittavat, että inhimilliset tekijät ovat merkittävässä roolissa kustannusarvioinnissa, ja lisäpanostukset johtamisnäkökulmien tutkimiseen ovat perusteltuja.

ACKNOWLEDGMENTS

Now, a few weeks before defending my thesis in front of the public, is a good time to stop and think why did I do it. I'm 45 years old, working long days in a busy company, and have a family and hobbies. It's been busy. However, I've always enjoyed learning new things. Also, I value education, and I admire researchers and practitioners, who discover novelties, contribute and make a difference. With my thesis, I wanted to make sure that I continue learning. I also had a dream of contributing myself. Now, that dream comes true. I wish that my work could help software professionals, even a little, to run more successful projects. That's why I did it.

Good things get even better in good company. I've had a dream team backing me and my work during the process, and I've enjoyed every bit of it. Ville, please accept my gratitude for your effort, persistence, practicality, wisdom and advise, among many other things. Without you and your team this wouldn't have been possible. Sami, you taught me everything, and showed that science can be fun. I'll always be grateful to you for being a mentor and companion to me during the journey. I'd also like to thank Jukka Teuhola for planting the seed for all of this during my Master's studies.

My sincere thanks also go to prof. Markku Oivo and prof. Jürgen Munch, who acted as pre-examiners to my thesis. Your suggestions were welcome. I'm also thankful to prof. Kari Systä for accepting the demanding role of the opponent, and taking the time to review my work. Of other scholars, I'd like to thank prof. Tommi Mikkonen for minding and supporting, prof. Marko Seppänen for the NPD stuff, and prof. Ivan Porres for methodology ninjutsu. Jukka and Johannes, thanks for the stats, you rock!

Joonas, thanks for making it possible to combine the research with my position as Vaadin's COO. Tapio, thanks for your generous offer, which I happily accepted, to pay for the drinks consumed in the after party. I'd also like to thank N4S research programme for providing a superior framework for research, and Nokia Foundation for acknowledgement and financial support.

Finally, mother and father, thank you for your love, and Linn – for everything.

LIST OF ORIGINAL PUBLICATIONS

[1] Jurka Rahikkala, Sami Hyrynsalmi and Ville Leppänen. "Accounting Testing in Software Cost Estimation: A Case Study of the Current Practice and Impacts". In: *Proceedings of 14th Symposium on Programming Languages and Software Tools*. Ed. by Jyrki Nummenmaa, Outi Sievi-Korte and Erkki Mäkinen. Tampere, Finland: University of Tampere, 2015, pp. 64–75.

[2] Jurka Rahikkala, Ville Leppänen, Jukka Ruohonen and Johannes Holvitie. "Top management support in software cost estimation: A study of attitudes and practice in Finland". In: *International Journal of Managing Projects in Business* 8.3 (2015), pp. 513–532.

[3] Jurka Rahikkala, Sami Hyrynsalmi, Marko Seppänen and Ville Leppänen. "The Impact of a Delayed Software Project on Product Launch Coordination: A Case Study." In: *Proceedings of 22nd ICE/IEEE International Technology Management Conference*. 2016.

[4] Jurka Rahikkala, Sami Hyrynsalmi, Ville Leppänen, Tommi Mikkonen and Johannes Holvitie. "Top Management Support for Software Cost Estimation-A Case Study of the Current Practice and Impacts." In: *PROFES*. 2017, pp. 89–107.

[5] Jurka Rahikkala, Sami Hyrynsalmi, Ville Leppänen and Ivan Porres. "The Role of Organisational Phenomena in Software Cost Estimation: A Case Study of Supporting and Hindering Factors". In: *e-Informatica Software Engineering Journal* 12.1 (2018).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

SCE	Software Cost Estimation
AI	Artificial Intelligence
CMMI	Capability Maturity Model Integration
IoT	Internet of Things
PMBOK	Project Management Body of Knowledge
IPMA ICB	International Project Management Association, Individual Competence Baseline
COCOMO	The Constructive Cost Model
SEER–SEM	Software Evaluation and Estimation of Resources – Software Estimating Model
PM	Project Management
TMS	Top Management Support
SW	Software
CMM	Capability Maturity Model
IT	Information Technology
IS	Information System
EDI	Electronic Data Exchange
HW	Hardware
CEO	Chief Executive Officer
SME	Small and Medium Enterprises
RQ	Research Question
ERP	Enterprise Resource Planning
SPI	Software process Improvement

CRM	Customer relationship Management
ICT	Information and Communication technology

Part I

SYNOPSIS

1 INTRODUCTION

The global software industry has emerged as a significant business segment in the last few decades [19], with five tech companies achieving the highest market value in the world (2017): Apple, Alphabet, Microsoft, Amazon and Facebook [34]. At the same time, the products and services of an increasingly broad spectrum of companies are becoming more dependent on software. As a result of this development, global Research & Development spending on software increased by 65% between 2010 and 2015 [116], and there is no slow-down in sight. Technologies like AI and IoT promising significant efficiency gains and profits guarantee continued investments in software-related innovations. Technology has begun to replace not only industrial workers but also information workers at a growing rate.

As a result of these developments, the success of software more frequently dictates the success of a company. The timing of a product launch is an important factor for product and company success. Delayed product launches have been found to cause, for instance, lower product margins, increased cost and decreases in firms' market value [49]. However, software projects are often unpredictable. They tend to overrun their schedules and budgets [30, 53, 44, 136], usually by a significant margin [101], causing hundreds of billions of euros in losses annually [38, 100, 106].

Considering the importance of understanding the time and cost required to develop software, practitioners and researchers have tried to estimate these parameters since the beginning of software's history, with the earliest studies dating back to the 1960s. Since then, software cost estimation has developed into a significant body of knowledge with hundreds of different estimation techniques and methodologies [97, 67] and is being included in all major project management and process improvement frameworks, including CMMI [17], PMBOK [56] and IPMA ICB [54].

1.1 WHAT IS A COST ESTIMATE?

Simply speaking, a software project cost estimate is a prediction of how long a project will take to implement and how much it will cost. A classical view of the estimation process is presented in Figure 1.1, where a chosen methodology is applied to create an estimate for effort, duration and cost, or loading, of a software based on the requirements and other cost drivers such as developers' experience and skill-level, programming language, size of the project and type of software to be developed. Almost always, the size of the software is estimated first, which is then converted into effort, schedule and budget.

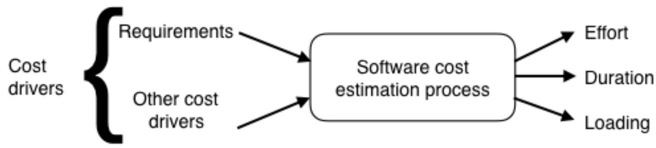

Figure 1.1: Classical view of software estimation process. Adapted from [140].

The reality is more complex. Many times, an estimate is confused with the concepts of target, commitment and plan. Typically, when business management asks for an estimate, they are actually interested in getting a commitment or plan to hit a certain target. These misconceptions need to be rectified to ensure that an estimate is utilised in a meaningful way and errors are avoided. The requirements are also seldom clear or consistent when the estimation commences, and there are other inputs than cost drivers that need attention, such as financial and other resource constraints, software process and overall company software architecture. A more precise SCE process is presented in Figure 1.2.

Estimates are used to prepare a project plan and control a project toward its targets. Project control activities include making decisions related to estimation outputs, such as adding or removing requirements or staffing or architectural changes. Project execution involves dealing with a plethora of unforeseen events, which usually invalidate the inputs, or assumptions, used for pre-

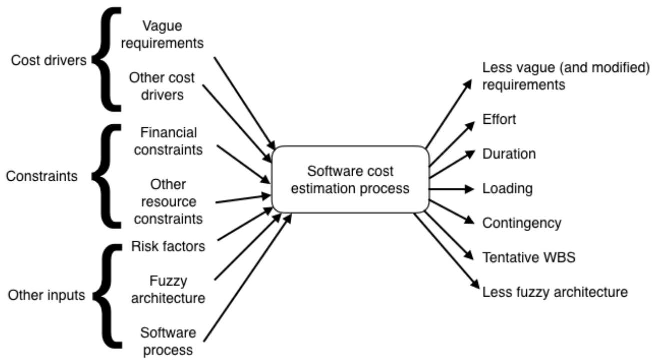

Figure 1.2: Actual software estimation process. Adapted from [140].

paring the estimate. This typical project control flow is illustrated in Figure 1.3.

Changes in the estimation assumptions call for re-estimation to gauge the impact of the changes on the estimate and whether the project can be controlled to meet its goals. Another reason for re-estimation is the gathering of new and more precise information regarding estimation inputs as the project progresses. Utilising this information enables refined and more accurate estimations, which are valuable for maintaining a reliable project status and controlling the project. This improved understanding of estimation inputs and their impact on the estimation accuracy can be illustrated with a cone of uncertainty [13], presented in Figure 1.4. According to the cone, the initial estimate is inaccurate and subject to an estimation error of 4x magnitude. When the project progresses under competent project management, the cone starts narrowing fast, indicating improved accuracy of the estimates.

As presented above, SCE deals with a lot of uncertainty, and estimates are inaccurate by nature, especially in the beginning of a project. This brings us back to the purpose of estimation and the definition of an estimate. Considering the uncertainties and changes described above, it would not make sense, or even be possible, to try to predict the outcome of the project accurately. Thus, the real value of an estimate is to help executives set good targets and help project management make good plans and take appropriate action to control the project toward achieving its

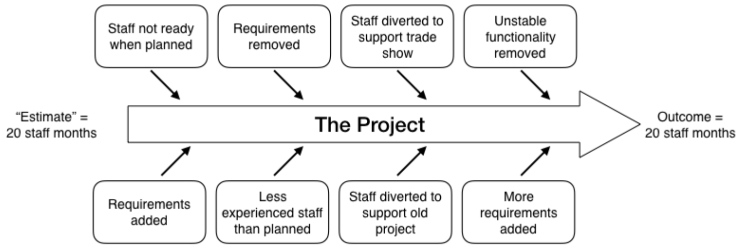

Figure 1.3: Project control flow. Adapted from [97].

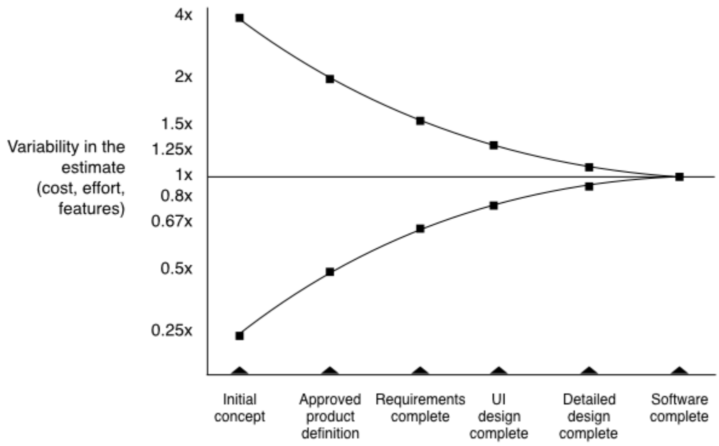

Figure 1.4: Cone of uncertainty; variability in the estimate by project phase. Adapted from [12].

goals. Based on this realisation, Steve McConnell [97] has defined a good estimate as follows:

> *A good estimate is an estimate that provides a clear enough view of the project reality to allow the project leadership to make a good decision about how to control the project to hit its targets.*
>
> — Steve McConnell

1.2 ESTIMATION METHODOLOGIES

The actual estimation methodologies are often divided into two categories in the literature, model-based and expert judgement-based [101]. In both categories, most of the steps are explicit, i.e., they are clear and can be described in detail. Examples of these explicit steps include breaking down the work into smaller pieces and combining these pieces into a whole. The difference between the categories emerges from steps for quantifying the size of software. In expert judgement-based methods, an expert, typically a software developer, assesses the size of a software artefact, while in model-based methods, the size is estimated based on input parameters, which convert into lines of code or function points based on a mechanical formula. Input parameters used by model-based methods may include several screens or reports and their complexity. However, it must be noted that estimating the complexity of a report, for example, is not deterministic and includes an element of expert judgement. In expert judgement-based methods, one or more experts estimate the size of a software artefact, like a screen or report, based on their judgement. This part of the estimation process is based on intuition, or a cognitive process, and cannot be described in detail. Popular model-based processes include COCOMO [13] and SEER–SEM [58], while popular expert judgement-based processes include Planning Poker [43] and Work Breakdown Structure [97, p. 117].

Neither of the method categories produce good results consistently. On the contrary, systematic overruns persist [30, 53, 44]. However, there are signs that the improvement of model-based methods is slowing down after decades of work, while judgement-based methods still seem to have significant potential for improvement [63, 67]. Research results also show that model-based processes cannot outperform judgement-based methods systematically, and judgement-based methods seem to produce better results in most situations [61]. Jørgensen [68] has proposed that this advantage of judgement-based methods would originate from experts' higher degree of knowledge of the estimation subject and their flexibility in processing this information. Models cannot incorporate all contextual parameters in a meaningful way. Modelling the relationships between the high number of parameters is complex and time-consuming, and understanding the importance of the parameters is critical. While parametric models cannot process contextual information effectively, judgement-based models are exposed to human biases, as the estimates are based on a thinking process. However, as mentioned above, despite these weakness in judgement-based processes, they outperform parametric models in most situations. Therefore, it is not surprising that software engineers have adopted the use of judgement-based processes to a high degree [71, 52, 80].

1.3 ESTIMATION BIASES AND DISTORTIONS

As expert judgement-based processes have gained popularity within the software industry, the research focus for improving the methodologies has also expanded. One important focus area is human biases distorting estimation. This is understandable, since the quantification of the software size or effort is based on human thinking processes. Reportedly, biases can emerge from various sources such as customer expectations [71], lack of separation between a bid ('price-to-win') and realistic effort usage [72], variations in wording, future opportunities or irrelevant information [66]. In addition, the selection of estimators affects estimation success. Estimators with more relevant experience [70, 64], more realistic track records on previous estimates, higher development skills and self-perceptions that are less than averagely optimistic [65] tend to produce more realistic estimates.

Cognitive biases are not the only factors distorting estimation. Many practitioners have argued that factors emerging from organisational contexts are equally important as estimation process-related factors [67, 94]. Indeed, some recent studies have found that organisational factors and even intentional distortions are common. For example, in a study by Magazinius, Börjesson and Feldt [94], 13 of 15 interviewees reported having experienced some type of estimate distortion. Published research papers report that organisational phenomena such as negotiations [95], hidden agendas [73] and selling ideas [94] distort estimation. Thus, estimation is a complex process combining many factors such as explicit estimation methodologies, cognitive processes of human individuals and distortions emerging from organisational contexts. In this thesis, organisational factors, or organisational phenomena, refer to factors connected to the organisational environment, structures and processes, along with the behaviours of individuals acting in the organisational context.

1.4 MOTIVATION AND RESEARCH GAP

As described in the sections above, organisational factors significantly affect software cost estimation. From this perspective, increased interest toward organisational factors is unsurprising. Increased focus can also be justified indirectly through the success of using estimation methodologies. Methodologies are shown to produce accurate results when used properly [113, 115]. Nevertheless, many software projects seem to be unable to use estimation methods effectively [94, 68, 51]; however, in project management, the use of methodologies has been found to have a positive impact on project success [149, 141, 24]. Thus, the problems that result in estimation errors seem to occur partially because estimation methodologies are used more ineffectively by organisations than reported in previous works.

While software projects may have difficulties in using proper methodologies [94, 68, 51], the situation is considerably better, for example, in the area of project management, as only 5% of projects do not use any PM tools [141]. Because cost estimation is an inseparable part of all projects and the cause of an overrun in software projects may reside in software cost estimation, project management or other areas [21, 98, 104], the difference in the extent of use of methodologies between SCE and PM is surprising, especially because commonly used industrial project management and process improvement frameworks, such as CMMI, PMBOK and IPMA ICB, promote the importance of estimation and the use of methodologies. The use of proper methodologies is proven to have a positive effect on the outcomes of both SCE and PM [85, 149, 148]; however, only PM professionals utilise these valuable tools and methods to any great extent. This observation suggests that organisational factors inhibit successful application of estimation methodologies in SCE.

As scientific literature or industrial advice do not provide clear explanations for the gap in the extent of use of methodologies between SCE and PM, one assumption is that the difference arises from organisational priorities and does not seem to be related to the availability of proven cost estimation methodologies. Project management is widely linked to the execution of corporate strategy, but SCE seems to have very little visibility among top management. Also, while project management research has paid close attention to non-technical factors such as top management support, communication, skills and learning, SCE research has mostly focused on developing and improving estimation techniques. This is an important observation, indicating that the explanation for the difference in the extent of use of SCE and PM methodologies could reside within the research areas omitted from the study of SCE.

Organisational factors and their impact on project success are thoroughly studied and well-understood in the field of project management; top management support and actions especially have been found as some of the most important factors affecting project success [90, 35, 149, 107]. The effects of top management support in project management correlate highly with overall project success [147, 35, 122], and some reports propose that TMS is the most important project success factor [147]. PM literature also lists various ways for top management to support projects [148]. Estimation inaccuracies emerging from management actions have also been reported to some extent in the domain of SCE. For example, management pressure [94], lack of management tracking of the estimation performance and removal of padding from estimates by managers have been found to affect estimation [95, 86]. However, top management support in SCE has not been studied systematically, leaving a significant gap in the body of knowledge.

Our work focuses on filling the gap related to top management participation in SCE by studying the impact of organisational factors, and specifically top management support, on successful use of estimation processes. Improving the understanding of the real-world dynamics may provide practitioners with valuable tools for improving SCE in organisations, and the gap between the advice provided by the industrial project management frameworks and the low extent of use of methodologies could be narrowed. The results could also provide further evidence that organisational issues are equally as important as technical ones for effective SCE and generate new theories about the reasons for continuing overruns in estimation.

1.5 RESEARCH OBJECTIVE AND QUESTIONS

Based on the described problems and research gaps, the objective of this research is:

To contribute to practices and theories on SCE by explaining the role of top management in SCE, specifically addressing how top management support SCE and determining the positive or negative impacts of this support.

Derived from the objective, the questions for this research are:

RQ1 How does top management support SCE?

While the first question is largely concerned with how and to what extent top management participates in SCE, the second question focuses on the actual positive or negative impacts of top management's actions, with the next question being:

RQ2 What are the impacts of top management support for creating a good cost estimate for a software project?

Satisfying the research objective and answering the questions requires a two-fold research focus: 1) the current practice of top management participation in SCE, i.e., behaviours for participation and their extent of use, and 2) analysis of the impact of top management actions.

1.6 OVERVIEW OF ORIGINAL PUBLICATIONS

In the following sections, we will describe each of the original publications in brief. Table 1.1 provides a mapping between original publications and the research questions they address.

1.6.1 *Paper I: Top management support in software cost estimation – A study of attitudes and practice in Finland*

In this paper, we identify a list of 16 practices from the literature that are likely to represent top management support for software cost estimation. We use survey research and statistical methods to verify that the practices measure the same construct, TMS for SCE.

Table 1.1: Mapping between original publications and the research questions they address.

	Paper I	Paper II	Paper III	Paper IV	Paper V
RQ1	X			X	X
RQ2	x	x	X	X	X

Notes: X, major contribution to RQ; x, minor contribution to RQ.

The list is used to capture the frequency of use and experienced importance of top management support practices in a questionnaire from 116 software professionals. We also study statistical correlations between frequency of use, experienced importance and project success to determine which support practices affect project success positively. The key finding of the paper is that top management invests a significant amount of attention in SCE; however, the extent of use and experienced importance of support practices do not correlate strongly with each other or project success. In practice, software professionals invite senior managers to participate in SCE, for which the paper provides a list of participation practices.

1.6.2 *Paper II: Accounting testing in software cost estimation – A case study of the current practice and impacts*

This paper focuses on cost estimation related to software testing tasks. We present in-depth findings from two case projects with 11 interviewees and report on the current practice for estimating software testing and the impact of used practices on SCE and project success. We emphasise the differences between software and testing estimation and show how these influence testing success. Finally, we examine the role of testing planning, attitudes connected to testing and other phenomena as explaining factors for the differences between software and testing estimation and discuss the impact on the estimation process. The results reported in the paper show that companies easily deviate from their standard procedures, when estimating testing, which may even lead to severe estimation errors. The deviations can be explained by negative attitudes towards testing. Furthermore, the paper shows that the extant literature has sparsely addressed estimation of software testing.

1.6.3 *Paper III: The impact of a delayed software project on product launch coordination – A case study*

In this article, we study the impact of a delayed software project on the related product launch through a case study involving one delayed software project. We present how management's delay-containment actions influence launch-related sales and marketing activities and their motivational factors. Impacts of sales and marketing actions are observed through changes in the scope, cost and quality of the planned activities, moderated by motivational factors. We also explain the role of reliable scope and schedule information throughout the project and describe how management actions may influence this. Finally, we provide evidence that pressure and volatility, among other things, connected to a delayed project hinder reliable and meaningful estimation. The results show that the delays may increase the cost of a product launch, as well as decrease the scope and quality of the launch activities. These impacts are influenced by key personnel's motivational factors.

1.6.4 *Paper IV: Top management support for software cost estimation – A case study of the current practice and impacts*

In this paper, we paint a complete picture of top management's role in SCE based on multiple case studies. We study real-life top management support practices for SCE and how they appear in organisations. The identification of practices is based on the list of 16 support practices developed in Paper I, which we validate and develop further in this article. We also report on the time and effort top management invests into participation in SCE. Additionally, we identify concrete artefacts, persons and items affected by top management actions and analyse the impact of top management actions on estimation and project success. The results show that top management takes no, or very little, direct actions to participate in SCE. However, projects can conclude successfully regardless of the low extent of participation. Top management actions may also induce bias in estimation, influencing project success negatively. This implies that senior managers must recognise the importance of seeking realism and avoid influencing the estimation in any direction.

1.6.5 *Paper V: The role of organisational phenomena in software cost estimation: A case study of supporting and hindering factors*

This article focuses on the organisational context of software cost estimation and describes organisational factors either supporting or hindering creation of a meaningful cost estimate. We focus broadly on organisational properties and mechanisms and report on factors emerging from human behaviour. We explain the impact of top management actions, SCE and SW process maturity, communication and attitudes in connection to supporting and hindering factors. To study SCE maturity, we develop a model for measuring SCE maturity in an organisation. CMM is used to measure software process maturity. The results suggest that the role of the top management is important in creating prerequisites for meaningful estimation, but their day-to-day participation is not required for successful estimation. Top management may also induce undesired distortion in estimation. Estimation maturity and estimation success seem to have an interrelationship with software process maturity, but there seem to be no significant individual organisational factors, which alone would make estimation successful. Our results validate several distortions and biases reported in the previous studies, and show that the SCE research focus has remained on methodologies and technical issues.

1.7 OVERVIEW OF RESEARCH CONTRIBUTIONS

This thesis investigates top management participation in SCE; the contributions of the thesis are divided into two parts: participation practices and impacts of participation. First, we identify participation practices and validate their relevance by measuring frequency of use, experienced importance and top management investment of time and effort into practices. Second, we investigate the impacts of participation practices on SCE and project success and report on delay management actions' impact on product launch. Figure 1.5 presents an overview of the contributions and how the research questions are linked to them.

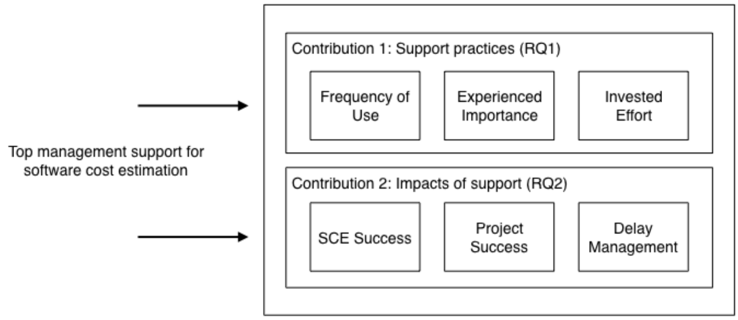

Figure 1.5: Contributions of the thesis.

Through the contributions of the thesis, we establish a concept top management participation in SCE and describe the current practice related to top management participation. We also show how top management actions influence estimation, projects and product launch success. The contributions are introduced in the following sections.

1.7.1 *Top management participation practices for SCE*

The first contribution of the thesis is an explanation of how top management participates in software cost estimation. Our approach is to first identify a list of potential participation practices from literature and then capture the frequency of use and experienced importance of these practices. Based on these metrics, we use statistical methods to show that the list coherently measures the same construct, top management participation in SCE. Then, we validate the results by showing how the practices are applied in a real-life setting.

The list of potential participation practices, the basis of the contribution, is adopted from the area of project management, where top management participation is thoroughly studied [148]. This list is complemented with best practices from the area of SCE to account for the specific nature of SCE [97]. The resulting list of potential participation practices contains 17 candidates. The frequency of use and experienced importance is captured in a survey study from 116 software professionals in Finland, and the consistency of the list is validated using statistical methods. Except for one practice, the list measures the same construct, top management participation in SCE, resulting in a list of 16 practices. The candidate removed from the list was concluded, as a result of a peer discussion, to have an unclear proposition, thus, not being valid from the research point of view. The results show that the average experienced importance is higher than the average frequency of use, with averages of 2.54 and 3.04 for frequency of use and experienced importance, respectively. The results did not show a correlation for individual practices between these two. This part of the research is presented in Paper I.

The next step in our approach is to investigate how these 16 practices are applied in organisations in a real-life setting. A multiple-case study in three organisations helped us gain an in-depth understanding of the question. Paper II, Paper IV and Paper V report the results from this research. The results deviate significantly from the preliminary results reported above. We found strong evidence for only five of the 16 support practices. Additionally, we discovered two new practices, which were present in all studied projects. We also show that top management invests very little time in their participation in SCE. However, although the extent of direct top management participation is low, we show that many of their actions are indirectly connected to SCE by creating prerequisites for successful estimation. For example, we show that top management provides direction for the project and ensures adequate resources and good communication in the organisation.

1.7.2 *Impacts of top management participation in SCE*

Our second contribution regards the impact of top management actions on a software project. In our first approach in Paper I, we investigated the statistical correlation between the 16 top management support practices and project success. Two practices were found to be significant with respect to project success. In our second approach, we conducted a multiple-case study in three organisations to examine the real-life consequences of top management actions on SCE and project success. The third approach focused on top management reactions to software project delays and the impact of those reactions on launching the software product in question.

In the second approach, we examined the effects of TMS on people and project artefacts. These were mapped to impacts on SCE and project success. The study discovered two direct TMS practices for SCE, 'Top management studies and approves the estimate' and 'Top management ensures the involvement of the project manager during the estimation stage', and five indirect practices related to resource provisioning, goal setting and demonstrated importance. Thus, top management's role was found to be mostly related to creating prerequisites for successful estimation instead of participating in estimation directly. We also found that top management can cause direct negative effects on estimation and the related software project. For example, expectations expressed by top management may become anchors [5] affecting estimation, and a lack of shared vision may cause the team to base the estimate and work on incorrect or incomplete requirements. These results are reported in Papers II, IV and V.

The third approach focused on the impact of top management actions on a product launch in the case of a delayed software project. In our single case study-based research, we studied top management reactions to project escalation, the impacts of containment measures on sales and marketing tactics and motivational factors related to the launch team. Impacts on sales and marketing tactics were observed through the lens of scope, cost and quality of actions. The results show that top management may try to align the situation to the business objectives quickly, although project realism would not allow this. On the other hand, it may create uncertainty about the schedule and scope of the project, which may have broad negative impacts on project launch activities. Our results suggest that establishing a reliable schedule and scope and therefore reducing uncertainty would be the most effective action to recover from delays and their negative impact. Paper III reports these results.

1.8 STRUCTURE OF THE THESIS

The thesis consists of two parts. Part I provides a research summary and explains the concepts and background information, while Part II presents the peer-reviewed original publications. Part I is structured as follows: Chapter 2 presents background and related work, Chapter 3 describes the research methodology and discusses the reliability and validity, Chapter 4 discussed the results of this thesis and their implications, as well as provides suggestions for future research, and Chapter 5 concludes the thesis.

2 BACKGROUND AND RELATED WORK

In this chapter, we provide a background for this thesis and review the related work. First, we establish the importance of successful software project deliveries to motivate our work properly. Then, we review the current practice for creating a good software cost estimate and common reasons why estimation may fail. We also present critical factors for successful project management. Finally, the role of top management in software projects is reviewed.

2.1 ROLE OF SOFTWARE

Software is a catalyst for many profound changes in the global economy. Gartner stated that IT spending will reach $3.7 trillion in 2018 [39]. PwC's The 2016 Global Innovation 1000 study reports that global R&D spending on software increased by 65% between 2010 and 2015, and R&D allocation for software is expected to increase an additional 43% by the end of the decade [116, 117]. The same study reports that companies investing more on software grow faster than their competitors. Non-tech companies are joining the software revolution [130]. John Deere says that they will soon employ more software than mechanical engineers. Pizza Hut is deploying robotic waiters, and people can order a pizza with a tweet. Car manufacturers and ride companies are developing autonomous cars, and Volkswagen's R&D spending is the highest in the world [116]. A Siemens' executive stated that everything they do—including all products and all services—is software-driven [131]. Software enables product differentiation to a higher degree and enhances the customer experience. At the same time, software is revolutionising the job market and the management and culture of companies. Therefore, it is clear that business success is becoming increasingly dependent on software.

Although software drives innovation and creates opportunities, benefits of software are not realised automatically. Software creates new business models, accelerates the innovation cycle and makes the marketplace more competitive [135, 33]. Thus, software needs to be done right, and there are many considerations when launching a software-driven product or service. The timing of market introduction is critical for a product's success [18]. Customers are increasingly asking for high-quality software products, leaving the companies with less time for development [47]. However, launching a product involving software depends on the outcomes from the software project, which are often late and difficult to predict [30, 53]. A vast number of studies have been conducted showing the relationships between speed to market, quality, costs and profitability (e.g., [127, 135]. Timing and good management of key aspects of the launch, such as marketing plans and overall launch direction, have been pinpointed as critical success factors for launch success [28], and it is generally known that marketing and technological execution proficiency are significant predictors of new product success [46]. Thus, firms that are able to develop software quickly on a predictable schedule are likely to gain an advantage over the competition.

2.2 IMPACT OF OVERRUNS AND ESTIMATION ERROR

Software projects are difficult endeavours. According to several studies on software project overruns, most projects (60%–80%, Table 2.1) are completed overbudget and/or late, with an average cost overrun of 30%–40% [101]. Approximately 20% of all projects are cancelled due to various challenges such as lack of senior management involvement, too many scope and requirement changes and high project overrun [32]. Expressed in financial terms, the annual losses from software projects are measured in the billions of euros [38, 100, 106]. The biggest single failures include the electronic care record system in the UK, the dental service system in Sweden and the combat support system in the US, where the cost was over EUR 1 billion before cancellation [144]. Thus, the direct cost impact of failure can be significant.

Often, the direct cost of a failure is not the only, or even the most significant, negative impact of an overrun. The literature suggests that the consequences of being late to the market include lower profit margins, higher production costs and decreased firm market value [49]. Scholars have also argued that speed

Table 2.1: Estimation accuracy results [101].

Study	Jenkins	Phan	Heemstra	Lederer	Bergeon	Standish
Cost overrun	34% (median)	33% (mean)			33% (mean)	89% (mean)
Projects completed over budget	61%		70%	63%		84%
Projects completed under budget	10%			14%		
Schedule overrun	22%					
Projects completed after schedule	65%		80%			84%
Projects completed before schedule	4%					

is associated with high-quality products [89]. In addition to impacts on firms' financial performance and competitiveness, overruns may have destructive impacts on operative and project dynamics such as apologising to key customers, fixing problems arising from quick and dirty workarounds and preparing interim releases for trade shows or demos [97]. If the software were ready on time, these situations could be avoided.

Failures and overruns are seldom related to one specific reason such as underestimating a project, poor project execution or scope volatility [93, p.13]. However, good estimation should foresee and account for challenges related to a project, assuming good project execution, to produce a reliable estimate. A reliable estimate is key to the success of a project in terms of financial success, customer satisfaction and other criteria. Too high or too low estimates have a biasing effect on the business case, meaning that projects may be approved or rejected on false premises, possibly leading to the situations described above. Generally, business cases justifying starting software projects are based on several important assumptions such as time-to-market, features, customer experience, cost and internal coordination of work. Most of these parameters are related to the schedule. If the schedule is overrun, the assumptions fail, causing various problems in the project. Thus, accurate estimation is important for the successful planning and control of a project, including budgeting, coordination of all activities and delivering quality and functionality.

Most executives are willing to accept higher costs, longer schedules and reduced functionality if they gain predictability in return [97, 115]. Businesses need to make commitments to customers, investors and the marketplace, and these commitments along with systematic planning and execution are supported by predictability. Although keeping promises is highly valued by executives, projects should not be overestimated. Additionally, overestimation may lead to rejecting viable projects on false premises and increased cost [42, 109]. However, executives' opinions and other arguments favour overestimation over underestimation.

2.3 REASONS FOR ESTIMATION ERROR

Since overruns are a systemic problem for the software industry, it is natural that sources of estimation error are thoroughly studied, resulting in an extensive body of knowledge. Regrettably, this body of knowledge has not been able to remedy the situation. However, without this knowledge, the situation would most likely be even worse, considering the explosion of global software development and growing project sizes. Knowledge of sources of errors provides valuable pointers for estimation process improvement.

Reasons for estimation error are many. To aid in reviewing sources of estimation error, we have divided them into the following categories, based on McConnell [97, pp. 33-34]:

- Inaccurate information about the project being estimated
- Inaccurate information about the capabilities of the organisation that will perform the project
- Inaccuracies arising from the estimation process
- Inaccuracies arising from human bias related to the estimation process
- Intentional biases arising from political issues and management

Additionally, chaos and variability in projects are common reasons for overruns [96]. However, this is primarily a concern for project control, not for estimation. Even the best estimators and estimates cannot predict the outcome of an out-of-control project. Thus, we consider this source of estimation error to be out of the scope of this review.

Before addressing the actual sources of errors, a reminder of the cone of uncertainty [13] is appropriate: in the beginning of a project, estimates are very inaccurate and the ranges are broad.

When the project progresses, emerging information allows the estimates to be tightened up. Thus, re-estimation should be applied throughout the project, as planned. However, the cone of uncertainty is not a source of estimation error, it just describes the levels of accuracy warranted in different phases of the project.

The following sections review the sources of estimation error by category. In addition to this brief review, there are several other causes for estimation error.

2.3.1 *Inaccurate information about the project*

Understanding the software and related requirements is necessary for accurate estimation. The cone of uncertainty describes how accuracy increases as a function of the project phase, i.e., the accuracy correlates with the amount of information available. However, the cone does not narrow by itself. Increased accuracy and information requires persistent and firm work from the buyer, users, project management, analysts and software developers to achieve this greater understanding. There is no room for sloppy investigation of requirements, missing activities, lack of user involvement or poor specifications if the intention is to tighten the range.

Larenjeira [83] said that the accuracy increases through decomposition of the complexity of the software. In their often-quoted study, Lederer and Prasad [87] reported that users' lack of understanding of their own requirements, frequent requests for changes by users and poor or imprecise problem definition are significant sources of estimation error. Van Genuchten [139] reported that causes for estimation error include requirements of insufficient quality or lateness, specs of delivered software of insufficient quality, underestimated complexity of application and more problems than expected with performance requirements or memory constraints. Stutzke [132] indicated that changes to product requirements and design become more expensive as the projects proceed. The cost of requirement errors may be 100 times higher if fixed after the system is in operation instead of fixing them at the start of the project.

To summarise, an estimate can only be as accurate as the understanding of the software warrants. Failure to establish a desired level of understanding of requirements leads to estimation errors.

Thus, all of a project's stakeholders should pay close attention to establishing and maintaining a shared understanding of what the software should accomplish. Additionally, all changes should be dealt with immediately, rather than postponing them.

2.3.2 *Inaccurate information about the capabilities of the organisation*

The reason for variability in or poor understanding of requirements may arise from the capabilities of an organisation. For example, analysts may be inexperienced or unskilled and are therefore unable to decompose and refine the requirements as planned. There are many other kinds of issues arising from organisational capabilities causing estimation error. Lederer and Prasad [85] concluded that the inability to anticipate the skills of project team members is a significant source of error. Subramanian and Breslawski [133] reported that the project manager's confidence in the estimation model, managerial experience and programmer or team member experience affect estimation accuracy, and Van Genuchten [139] found that too little experience with development environments and more inexperienced people on a team than expected can cause errors.

Generally, a lack of organisational capabilities means that an organisation is not equipped to accomplish what needs to be done to deliver the software as planned. The lack of capabilities can be related to various factors ranging from the experience of project managers or analysts to their experience with development tools, programming languages, testing methods, integration frameworks, development methods or project management methods. Even the whole business area may be unfamiliar. Oftentimes, estimates are based on an assumption that all capabilities will be available for the project. Equally often, this is not the case, resulting in errors. Therefore, it is crucial to understand which people, processes and technologies are involved in the project and their levels of experience and to account for this in the estimation.

2.3.3 *Inaccuracies arising from the estimation process*

Software cost estimation research has been methodology-centric for decades, thus it is obvious that methods contribute significantly to estimation accuracy. However, they are also a source for estimation error. Hihn and Habib-agahi [51] report that 83% of estimates are prepared primarily using informal analogy—without using a planned and defined process. This has been found to be connected to estimation errors [85]. Heemstra [48] reported that 35% of organisations do not make proper cost estimates and recommended using SCE methods and ensuring estimator commitment. Although the studies are old, according to our personal experience, ad-hoc or off-the-cuff estimates are not uncommon today. The first step is to pursue an accurate estimate and choose a proper methodology for creating one.

The number of estimation methods, or processes, is measured in hundreds, making an exhaustive review of different inaccuracies impossible. However, we have selected examples for illustrating problems arising from the estimation process. Lederer and Prasad [87] reported that errors are caused by lack of an adequate methodology or guidelines for estimating, inability to tell where past estimates failed, lack of setting and review of standard durations for use in estimating and insufficient analysis when developing estimates. Also, overlooked tasks, insufficient user-analyst communication and understanding and lack of adequate guidelines for estimating cause inaccuracies [85]. Jørgensen [62] reported that a variation in and poor communication of what is meant by the effort estimate may easily contribute to effort and cost inaccuracies and cause planning and budgeting problems for software projects. McConnell [97] continued the list with unexpected events and overlooked tasks (such as data conversion, mentoring of new team members and installation), projects estimated by a developer as opposed to a PM and projects estimated by a person not participating in the project.

2.3.4 *Inaccuracies arising from human bias*

As mentioned in the Introduction, human biases are an important source of error because practically all estimation methodologies involve human judgement and thinking processes. Considering the extent of overruns reported, it is unsurprising that optimism is an important factor distorting estimates [112]. Moløkken-Østvold and Jørgensen [102] have continued with this theme and found that people in technical roles are even more optimistic than people in non-technical roles. Several other studies concur [139]. Optimism is an example of self-induced bias, but a human mind can also be affected by exposing it to pressure, irrelevant information, future opportunities, expectations and cognitive anchors, among others [72, 69, 5, 66].

A significant amount of effort has been put into finding ways to avoid bias. Means for avoiding bias include selecting persons less prone to bias and distractions as estimators. As concluded in the Introduction, estimators with more relevant experience [70, 64], more realistic track records on previous estimates, higher development skills and self-perceptions that are less than averagely optimistic [65] tend to produce more realistic estimates. Lederer and Prasad [88] also proposed that reviewing the estimation accuracy should be included in performance reviews to shun bad practices.

Another method of reducing bias is training the organisation and altering the estimation process so that estimators are less exposed to pressure, irrelevant information and other distractions. For example, management should avoid anchoring their expectations prior to estimation [5] and should pursue realism [60]. There is excessive pressure in almost all large projects [59], so it is easy to understand that the temptation to please the management is high [60]. Furthermore, obvious countermeasures have been proposed for almost all reported sources of bias: remove irrelevant information before handing it over for estimation, do not show customer expectations of the price for estimators and so on. As the sources of biases are many, it is ultimately important to understand the phenomena and critically review potential influences.

2.3.5 *Political issues and management*

Cost estimating is a political activity [84]. It may be intentionally influenced by different stakeholders from upper management to users to fit the estimates into their personal agendas [84, 112, 94]. For example, if management and estimators want to complete projects on time, management encourages high estimates, so estimators pad estimates, and if they want to do more projects, management encourages low estimates and estimators shrink estimates. People with the greatest power on a software project, including upper management, IT management and software professionals, are in the best position to play politics. Although this political colouring of estimation has been recognised for decades [84, 112], surprisingly few studies have focused on the phenomenon [67], even though otherwise meticulous estimation work can easily be voided by distortions.

As with other sources of estimation inaccuracies, the range of individual causes of error is broad. Lederer and Prasad [87] reported that pressures from managers, users or others to influence the estimate, reduction of project scope or quality to stay within the estimate and removal of padding from the estimate by management are common. Magazinovic and Pernstål [95] showed that estimates are affected by budget and management goals. Additionally, Magazinius et al. [1] found that selling ideas, personal agendas, negotiations, hiding activities in other activities to buy extra time, increasing estimates to drop undesired functionality out of the score and job securing occur in projects [94]. The distortion is not limited to the planning phase of a project, as tracking costs can be intentionally incomplete or misleading [95, 2, 41] and it is typical for IS management to fail to examine estimates carefully [87].

2.3.6 *Frequency of occurrence*

The occurrences of different estimation inhibitors have been addressed in several studies over the decades, as summarised in Table 2.2. All the categories reviewed in the previous sections are represented, however, certain topics are more common than others. For example, changes and overlooked tasks repeat in several studies. Also, topics related to understanding the project, organisational capabilities and the estimation process itself repeat in the research findings, while findings related to human bias and politics are rare.

Although politics and human bias-related inhibitors are rarely reported, this does not imply that they would not affect the estimation and may also be significant inhibitors for estimation. According to our experience, but not reported in this thesis, many of the previously reported inhibitors are interlinked. For example,

1 Ana Magazinius has published also under the name Ana Magazinovic, thus Magazinius and Magazinovic refer to the same person.

we have witnessed situations where unexpected and overlooked tasks, lack of understanding of the requirements and feedback problems may have been caused by actions of senior management. Effects of management actions have not been thoroughly studied in the area of SCE, and causal relationships and consequential and indirect effects are not well-understood.

In the area of project management, the impact of politics is well-recognised. Atkinson [7] concluded that quality is measured based on attitudes, and project results are measured against subjective perceptions. Thus, the original goals are also set based on people's subjective interpretations of the desired outcome when only the least-solid information is available. Shenhar et al. [125] suggested that project managers "need to see the big picture" or "be aware of the results expected", which describe how complex and subjective goal setting and measurement are in project management. Buchanan [16] reported that 90% of practitioners state that utilising politics is necessary to succeed, and 84% say that they would use politics when necessary. Generally, political behaviour is common in all organisational levels and is well-documented [16]. Considering the examples drawn from the extensive body of knowledge of organisation politics, it is likely that software cost estimation is more of a political activity than anticipated. Thus, the consequential and indirect effects of politics are likely broader than is currently reported, which may manifest through the reported non-politics–related reasons.

2.4 TOP MANAGEMENT SUPPORT AND ROLE

Leaders at all levels establish unity of purpose and direction and create conditions in which people are engaged in achieving the organisation's objectives [55]. Top management, or senior management, refers to the group of senior executives and decision makers responsible for the overall strategic direction of the organisation [143]. Thus, organisations resemble their leaders in many ways; the leaders shape organisations with their behaviours. Typically, these behaviours are beneficial to endeavours within an organisation [29]. For the previous reasons, top management's role and actions have been thoroughly studied throughout the decades from different perspectives. For example, in project management, leadership-related topics have become the most popular

Table 2.2: Results from studies on perceived cost estimation inhibitors [93].

Study	Top perceived cost estimation inhibitors
Phan et al. (1998)	Optimism
	Frequent changes
	Duration
van Genuchen (1991)	More time spent on other work than planned
	Complexity of application underestimated
Lederer and Prasad (1995a)*	Frequent requests for changes by users
	Users' lack of understanding for their own requirements
	Overlooked tasks
Subramanian and Breslawski (1995)*	Requirement change/addition/definition
	Programmer/team member experience
	Staff turnover
Jørgensen and Moløkken-Østvold (2004)*	No systematic feedback
Interviews	Poor project planning and overlooked tasks
	Poor requirements specification
Jørgensen and Moløkken-Østvold (2004)*	Unexpected and overlooked tasks
Questionnaires (qual analysis)	Change requests from clients
	Simpler task/more experienced developer than expected
Jørgensen and Moløkken-Østvold (2004)*	Project estimated by developer
Questionnaires (stat analysis)	Project estimated by a person outside the project
	Client focus on time-to-delivery, not cost or quality
Morgenshtern et al. (2007)	High level of uncertainty
Magazinovic and Perstål (2008)	Feedback problems
	Shared resources between projects
	Negotiations
	Requirement uncertainty and change

Notes: The three highest ranked inhibitors are presented for studies where the number of inhibitors is too high for all to be included, marked with *.

theme for research, with an increasing popularity from decade to decade (Table 2.5). In total, top management role-related research papers form a formidable body of knowledge that aids in studying top management roles in connection to less-studied areas such as top management roles in software cost estimation.

Top management support is typically divided into two separate constructs, involvement and participation, although the terminology for these constructs may vary [57]. Involvement and participation refers to psychological state and behaviour and activities performed, respectively [10]. In their terminology, Jarvenpaa and Ives [57] divided TMS into attitudinal and behavioural interpretations. Attitudinal interpretations cast TMS as a set of favourable attitudes, involvement, commitment and 'opinions and desires', whereas the behavioural interpretations present TMS as a set of direct managerial behaviours such as offering technical assistance or engaging in 'activities or substantive personal interventions'.

Dong et al. [29] listed 19 IT system-related studies where top management participation has been investigated. The studied areas include IT adoption, IS success, EDI success, data warehousing success and use of technological innovation, among others. With a few exceptions (e.g., [23, 138]), the presented TMS measures were attitudinal. Measures were of the nature, "...supports information system", "...understands the importance", "...is aware of the benefits" and "...strongly encourages". Positive effects of TMS were widely reported, however, not in all cases. In the two papers mentioned above, Compeau and Higgins [23] reported that top management availability for equipment selection, HW and SW difficulties and specialised instruction did not affect computer efficacy. Thong et al. [138] reported that the CEO's hands-on participation in IS implementation was not related to user satisfaction. Thus, behavioural TMS and positive impacts emerging from it are rare. Provisioning of resources is also closely connected to TMS. While this is probably an attitudinal practice in most cases, it may also be a behavioural practice. Nevertheless, resource provisioning is also a widely studied concept in various areas and has been found to contribute positively to project success, IS success and other areas (e.g., [1, 14, 124]).

In the field of project management, TMS is considered one of the most important, if not the most important [147], success factors. Fortune and White [142] reported in their study of 236 respondents working in projects in various countries and industries, 63% of which were managing projects, that support from senior management was the second most important success factor for projects. The most important was clear goals and the third most important was adequate budget/resources, both of which are tightly coupled to the power and role presented by senior management in organisations. Madanayake et al. [92] conducted a qualitative case study with five case projects to identify critical top management support practices in the software sector. They presented a list of 16 support practices categorised into three roles assumed by top management (Table 2.3). Top management roles have also been studied from the risk perspective. Arnuphaptrairong [6] summarised in his review of 12 studies related to software risks that lack of management commitment and support is the second most significant risk for software projects.

Although TMS seems to be mostly attitudinal, the project management literature has also identified behavioural types of top management support practices that contribute positively to project success [14, 148, 149]. These practices provide assistance to senior management on how they should use their valuable time. Success factors have been identified separately for different countries and industry segments, including the software industry [92, 149]. Zwikael's list [149], based on quantitative survey research, consists of 10 top management support practices contributing positively to project success:

1. appropriate project manager assignment,
2. refreshing project procedures,
3. involvement of the project manager during the initiation stage,
4. communication between the project manager and the organisation,
5. existence of a project success measurement,
6. supportive project organisational structure,
7. existence of interactive inter-departmental project groups,
8. organisational project resource planning,
9. PMO involvement and
10. use of standard project management software.

McLeod and MacDonell [99] described the broad role of top management, which is perceived to play into the scope of software system development. Top management is expected to ensure adequate financial and human resources and that the project is aligned with organisational goals and strategies. Top management should also influence attitudes, resolve political conflicts and create a positive context for change, among other things. This spectrum of power exercised by top management creates plenty of opportunities for influencing success factors or reasons

Table 2.3: Analysis of top management actions.

Strategy	Facilitate	Lead
Workout a sustainable business model	Supply resources	Accept ownership and gain better understanding of project work
Have clear business objectives and state them	Make necessary information available	Review project plans
Provide challenging work	Retain key employees	provide guidance
Balance project assignments	Liaise with customer	Watch status
Build support in the organizational model	Boost employee morale	—
Participate in scope definitions	—	—

for failure. Poon and Wagner [114] suggested that TMS is a meta-factor encompassing other success factors. This proposition is also supported by Young and Jordan [147]. The concept of meta factors is interesting, especially related to TMS, and should be studied further in connection to SCE. It is also worth noting that no practice in the previous review addresses cost estimation or overruns specifically, although these continue disappointing and frustrating senior management year after year.

2.5 TROUBLE ESCALATION

Many software projects end up in trouble for overrunning their budgets significantly. When this happens, top management, project team members and other stakeholders need to decide what to do with the runaway project. Surprisingly, it appears to be difficult to terminate a troubled project. Keil [76] explained that many times, projects seem to take on a life of their own and are allowed to continue to absorb resources until they are finally terminated. This phenomenon of not being able to terminate a troubled project is called escalation of commitment, or simply escalation.

Escalation occurs for various reasons. Staw and Ross [128] divided these reasons into project-related, psychological, social or organisational reasons. Keil [79] suggested that cultural reasons also impact escalation, as well as reluctance to report bad news [126]. Keil et al. [78] also reported that "sunk cost" ("so much money has been spent already") and "project completion" ("it is so close") effects make de-escalation difficult. Schmidt and Cavaltone [121] suggested that new product development projects are especially prone to escalation since they are rarer and people grow enamoured with them. These projects are also allowed to continue to launch commercialisation phases, even if performance reviews are poor and costs increase in these phases.

Therefore, many projects are allowed to continue longer than necessary before termination or proceed to conclusion even though the status analysis would advocate termination. The situation is so serious that Lyytinen and Robey [91] claimed that organisations fail to learn from their failures, and failure becomes a new norm. The effects of overrun projects are not limited to sunk cost only. The literature suggests that the consequences of being late to the market are significant, causing lower profit margins, higher development and production costs and decreasing the firm's market value [49]. Scholars have also argued that speed keeps costs in control, is associated with high-quality products [89] and helps to ensure early entrant advantages and overall profitability (e.g., [22]). Suboptimal project decisions can also harm brands and erode the credibility of an organisation in the eyes of its stakeholders [120].

When the time for blowing the whistle comes, it is usually done by the senior management or IT management [77]. Early warning signs for escalation include lack of top management support, lack of documented requirements and weak project managers [74]. When the stakeholders, typically senior management, take action to de-escalate, the typical actions include redefining the project, improving project management and changing project leadership if the project is not terminated [77]. To summarise, top management holds a key role when projects escalate. They make the conscious or unconscious decisions to let the projects escalate or take actions to de-escalate. These top management behaviours significantly influence not only the project success but also the product and organisation success.

2.6 SCE AND PM RESEARCH FOCUS

The first software cost estimation studies date back to the 1960s [103, 105], and thousands of research papers have been published since [15, 68], resulting in a notable body of knowledge. The most prominent areas of interest have been estimation methods and size measures, as presented in Table 2.4. The recent study of SCE research trends shows also that the research focus has remained consistently on estimation methodologies and techniques between 1996 and 2016, the emerged research areas being 'size metrics', 'estimation by analogy', 'tools for estimation', 'soft computing techniques' and 'expert judgement' in five topic solution [123]. The researchers seem to believe that the answer to the continuing estimation errors resides in the methodologies; the secret formula just has not been found yet. At the same time, organisational issues have been the focus of only 16% of studies (Table 2.4).

Considering the reported reasons for estimation error (Table 2.2), the research focus is confusing. The practitioners report that issues like unstable requirements, feedback problems and poor planning are among the most common reasons for estimation error. These topics are primarily organisational matters unrelated to methodologies or size metrics. Many practitioners concur with this observation [67, 97]. Additionally, Jørgensen and Shepperd [67] pointed out that only eight out of 304 articles were in-depth case studies and only three evaluated the background of the estimation processes. Thus, focus on an in-depth understanding of real-life phenomena behind estimation has been low, and it is possible that the dynamics behind the estimation inhibitors may not be thoroughly understood.

At the same time, while SCE researchers put their effort into improving methodologies, the situation is vastly different in the area of project management. Kolltveit, Karlsen and Grønhaug [82] analysed 562 project management-related articles published in the International Journal of Project Management between 1983 and 2003 and categorised articles in six perspectives, as presented in Table 2.5. Task and Transaction perspectives deal mainly with non-organisational issues, while the other four perspectives mainly focus on organisations. As Table 2.5 reveals, the research focus has undergone a major shift from methodology-heavy areas

Table 2.4: Distribution of published SCE articles among research topics. [68]

Perspective	-1989	1990-1999	2000-2004	Total
Estimation method	73 %	59 %	58 %	61 %
Size measures	12 %	24 %	16 %	20 %
Organisational issues	22 %	15 %	14 %	16 %
Uncertainty assessment	5 %	6 %	13 %	8 %
Calibration of models	7 %	8 %	4 %	7 %
Production function	20 %	4 %	3 %	6 %
Measures of estimation performance	5 %	5 %	6 %	5 %
Data set properties	0 %	1 %	2 %	1 %
Other	0 %	2 %	1 %	1 %

toward leadership- and business-related topics. In the first observation period (1983–1987), 68% of the studies reported findings primarily related to Task and Transaction perspectives, while the share of these categories decreased to only 18% by 2003–2004. This shift of focus seems reasonable, since organisational issues are reported to be even more important factors to project success than technical ones [90, 35, 148, 107]. Top management support has even been suggested as the most important factor affecting project success [147], which corresponds with the largest share of leadership perspective-related papers.

Thus, project management research seems to converge with the reported problem areas, while SCE research continues improving methods and measures, although the reported problems reside elsewhere. Counterintuitively, the extent of use of methodologies seems to be much higher in project management. Fortune and White [142] reported that only 5% of projects do not use any PM tools, while Hihn and Habib-agahi noted that only 17% of estimators used proper estimation methodologies [51]. Although not reported in research papers, it seems reasonable to assume that the leadership and business focus in project management has increased the awareness, extent of use and effective application of methodologies, and, again counterintuitively, a tight methodo-

Table 2.5: The distribution of published PM articles among different perspectives. [82]

Perspective	1983 -1987	1988- 1992	1993- 1997	1998- 2002	2003- 2004	Total
Task	49%	34%	32%	23%	12%	29%
Leadership	8%	16%	25%	28%	33%	23%
System	23%	25%	18%	19%	15%	20%
Stakeholder	1%	3%	1%	5%	6%	3%
Transaction	19%	9%	6%	10%	6%	10%
Business	0%	13%	17%	15%	29%	15%

logy focus does not increase use or improve effectiveness of the application of methodologies.

2.7 BACKGROUND RELATION TO RESEARCH QUESTIONS

Overruns are still common in software projects regardless of significant effort invested in improving estimation processes. Although the review of causes for estimation error show that many are related to human factors, the research focus has remained on methodologies. When accounting for the reported frequency of occurrence of estimation inaccuracies, it should be observed that in most cases, human factors have a significant role. Thus, the assumption that estimation error would decrease significantly by inventing a new, better estimation model or methodology seems challenging, as well as the idea that humans would be able to follow any methodology precisely and cold-bloodedly. On the contrary, humans are easily misguided, and they sometimes distort results deliberately [69, 5, 66, 84, 112, 94]. There is also strong evidence that even simple and easy-to-use methodologies produce good results if used effectively [113, 115], supporting the importance of human factors. Therefore, a fair course of action is to seek improvement for SCE from other areas than methodologies.

This thesis focuses on improving estimation practices and accuracy by focusing on top management roles in software cost estimation. Senior management roles are broad in organisations, and their actions have a significant impact on operations. The software estimation error causes and top management role-related literature reviewed in this chapter provide a good reference framework for studying our research topic. Thus, the first research question is:

RQ1 How does top management support SCE?

Project management, top management, risk management and many other disciplines indicate top management support as one of the most impactful factors. Leadership has become the most studied topic in project management. Thus, top management participation seems to be a good direction when expanding SCE into non-technical research topics. Since TMS in SCE has received little attention so far, this thesis broadly explores the impacts of top management actions in an organisation, reflecting the phenomena reported in the literature and addressing the second research question:

RQ2 What are the impacts of top management support for creating a good cost estimate for a software project?

As presented above, this thesis connects the research questions to the previously established theory from other domains. However, the extant literature utilised in this thesis, as well as the scope of study, are by no means exhaustive. The goal has been to draw from the existing theories, connect them to our efforts and use them to aid in adding new findings and theories to the body of knowledge as well as to make suggestions for further research.

3 METHODOLOGY

Three empirical studies, one quantitative and two qualitative, were conducted to address the research questions. This section describes the used research strategy and methodologies of the studies and discusses the validity of the studies and results.

3.1 RESEARCH STRATEGY

The research questions of this thesis are exploratory in nature, discovering what is happening, seeking new ideas and generating hypotheses and new research areas [118]. They explain the practices associated with top management participation in SCE, the extent of use of the practices and the impacts of practices on SCE. Answers to these questions add to existing knowledge regarding top management participation in SCE, which is an understudied topic in the field of software estimation. Furthermore, this thesis adds knowledge from the in-depth organisational stance, rather than measuring aspects of top management participation. Thus, the philosophical perspective of the studies is interpretive [108].

The interpretive perspective aims to understand the dynamics of humans taking action in social settings and acknowledges that organisations and social structures do not behave rationally and objectively. Interpretive perspective is the most appropriate for this thesis, since top management as a topic is generally politically loaded [16]. Previous research also reports on politics, personal agendas, human bias and other behaviours that can be considered non-objective or non-rational and their influence on SCE. We have also witnessed many behaviours distorting and influencing SCE, either intentionally or unintentionally, and therefore believe that SCE cannot be improved effectively without seeking an in-depth understanding and interpretations of observed phenomena including motivational factors, social actions and interactions between related stakeholders.

The conducted research investigates specifically top management participation in SCE, thus a clear definition for top management is necessary both for conducting the research and interpreting the results. The literature presents various definitions for top management. In the Background section, we mention that top management, or senior management, refers to the group of senior executives and decision makers responsible for the overall strategic direction of the organisation [143]. This definition reflects the widely used theoretical construct of *dominant coalition* [26], which generalizes the term top management. The term dominant coalition refers to a group of powerful people, who sets the goals and strategic direction of a company. However, the empirical definition varies. Carpenter, Geletkanycz and Sanders [20] present a list of definitions for a top management team from over 30 research papers. This variance in definitions is acknowledged also by researchers, and e.g. [110] propose that 'rather than assuming titles and positions as indicators of involvement, the first task ... is to identify which players are involved and why'. We have assumed this approach in our work, and from our work's viewpoint, top management refers to the furthest up level of management, who is involved in the project and related estimation at hand. Hence, in smaller firms top management may include the CEO and management team, while in large corporations top management may refer to business unit or department heads, and their leadership teams. There are also many terms referring to top management. In our work, we use *senior management*, *senior executives* and *upper management* as synonyms for top management.

Finally, this thesis is based on empirical investigation. The reported findings are based on empirical evidence, which are gained through direct and indirect observation and experience. A survey and two case studies were conducted as part of the research. The survey initiated the research, acting as a pre-study for the following case studies to assure that important issues were not foreseen [145]. The more in-depth studies conducted in relation to this thesis were case studies.

3.2 RESEARCH PROCESS

This thesis consists of three studies, five publications and two research questions, as presented in Figure 3.1. Before the actual research commenced, a literature review was conducted in the beginning of 2014 to gain an understanding of the most recent findings and current practices in the areas of software estimation, project management and top management participation. All studies were conducted as part of the N4S research program [27] between 2014 and 2016. The used research methodologies are summarised in Table 3.1.

STUDY 1 is a survey of a descriptive and explorative nature [9] reporting the views of 114 Finnish software professionals regarding the extent of use and experienced importance of top management support practices for SCE. In addition to a pre-defined list of 16 support practices, the respondents were also able to propose other practices. The statistical correlation between the extent of use and experienced importance of support practices and project success was tested. The respondents worked in various roles in the studied projects including senior managers, developers and project managers. The studied projects varied from small to large, and the involved organisations included all sizes of firms from SMEs to large multinational corporations. The findings of Study 1 are reported in Paper I and contribute to RQ1 and RQ2.

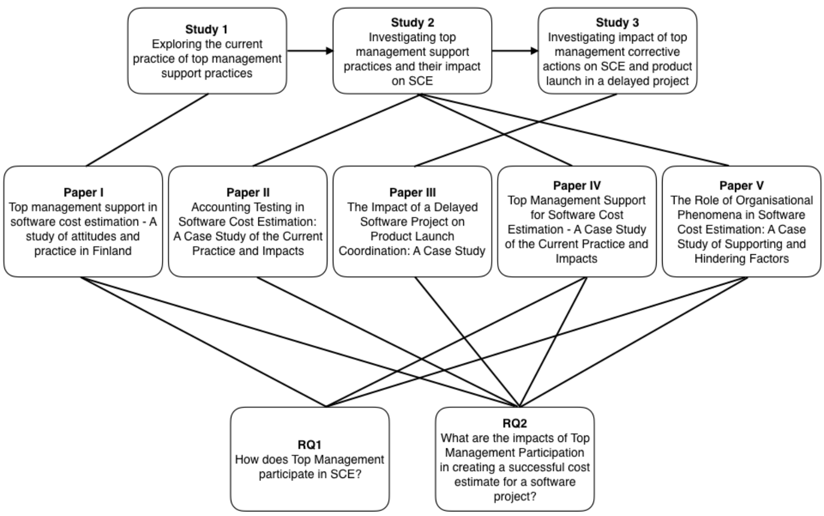

Figure 3.1: Overview of the conducted research.

Table 3.1: Summary of research methodologies

	Study 1	Study 2	Study 3
Strategy	Survey	Multiple case study	Single case study
Approach	Quantitative	Qualitative	Qualitative
Purpose	Exploratory	Exploratory	Exploratory
Data collection method	Web-based questionnaire	Semi-structured interviews Documentation review	Semi-structured interviews Documentation review
Data analysis method	Multivariate statistical	Qualitative (NVivo 10)	Qualitative (NVivo 10)
Sample size	114 respondents	3 projects 15 people 18 documents	1 project 7 people 115 documents
Reporting	Paper I	Paper II, IV and V	Paper III

STUDY 2 is an explorative follow-up study for Study 1 deepening the understanding of top management support practices for SCE and investigating the impact of top management actions on SCE. The study is a multiple-case study following the replication logic [146] and involving 15 senior software developers from three organisations and projects. One of the organisations was an SME and two were large corporations. The sizes of the studied projects were 20, 44 and 200 person months, respectively, with a duration of less than one year. The findings of Study 2 are reported in Papers II, IV and V and contribute to RQ1 and RQ2.

STUDY 3 is a single case study focused specifically on top management actions related to a delayed software project. The study further deepens the results of Study 2 in a setting where top management involvement is typically high and reports on the impact of top management actions on SCE and product launch. Study 3 involved seven interviewees, and 121 project related documents were investigated. The case company and case project were the same as in Study 2 (SME), but Study 3 was an independent study with a different research plan and interviewees. The planned duration of the project was three months, but after seven schedule extensions, the product was finally released 12 months after the project start. The findings of Study 3 are reported in Paper III and contribute to RQ2.

3.3 DATA COLLECTION

This section describes the data collection methods related to the three studies.

STUDY 1 was a survey study using a questionnaire of 17 questions for data collection. The goal of the questionnaire was to

1. collect information about the respondent and the project in question,
2. identify the extent of use of TMS practices in SCE and
3. identify software professionals' opinions on the importance of different TMS practices in SCE.

As typical for surveys, this study also collected information about attitudes and behaviour [75] in retrospect [111]. The recipients were instructed to answer the questionnaire only if they had been actively involved in the estimation of a software project within the past 24 months and to base their responses on their most recent estimated project, including abandoned projects. Responses were received for approximately three months, from the end of May to the end of August 2014.

Before the actual survey, a pilot survey was sent to 30 respondents. Based on the feedback from the 16 pilot survey responses, 20 changes were made to the questionnaire. The main survey was implemented as an anonymous, internet-based questionnaire, and a link to the questionnaire was sent to 1,109 software professionals. The recipients of the invitation were selected based on the mailing lists provided by the Project Management Institute Finland chapter and Dimecc Oy (Dimecc Oy is a non-profit Finnish organisation that coordinates national and international research programmes). Of the recipients, 114 completed the entire questionnaire (10.28% response rate, partial responses were excluded from data analysis). The questionnaire was anonymous to maximise the number of responses.

The first two of the 17 questions were screening questions determining whether the respondent was eligible to participate in the questionnaire. The next four questions determined the role of the respondent in SCE. Questions 7–10 measured the experienced importance and frequency of use of TMS practices in SCE. The respondents were also asked to add their own suggestions for practices. The rest of the questions determined the characteristics of the estimated project.

STUDY 2 was a multiple-case study using replication logic [146] in which the data collection was conducted over seven weeks in June and July 2015. The cases were selected to generate rich information about the phenomena being studied. We focused on large and small companies, selecting higher and lower maturity organisations and exemplary and challenged projects. The case companies and projects are different in their industrial domains, sizes and processes. The final decision to include each project in the study was made based on a discussion with a company representative confirming that the project was likely to add new perspectives to the study.

The unit of analysis was a single software cost estimate. The study was focused on the experiences gained during the preparation of the cost estimate. The primary data collection methods were semi-structured interviews [118] and a review of documentation. An interview protocol consisting of questions related to top management participation in SCE was created following the guidelines of Runeson and Höst [119]. In total, 15 people were interviewed from three organisations, and 18 documents were reviewed (Table 3.2). The documents included typical project documentation such as cost estimates, project plans, meeting minutes and status reports to gain a better understanding of the procedures and SCE methods used. The case studies were completed one at a time to allow for the reflection and refinement of the research and interview questions [81]. All interviews (but not key informant interviews) related to a single case study were conducted on the same day, except for one interview for the last case study. The one-hour interviews were conducted by two researchers, and the discussion was recorded. One researcher acted as the interviewer and the other took notes. The recordings were transcribed and sent to the interviewees for review. All case subjects participated in the study voluntarily and anonymously, and the collected data were treated as confidential. The case companies chose to remain anonymous.

Each interview day was preceded by a key informant interview day during which background information about the case was collected from a person in a central role in the case study area. The key informant interviews addressed the following topics:

1. Project background, size, status and success

2. Project team members and their roles
3. Estimation methodology and success
4. Software development methodology
5. Software process maturity, capabilities and track record

The semi-structured interviews were based on a predefined list of questions. Additionally, we employed the list of 16 top management support practices for studying top management participation practices. Any interesting facts and observations that were mentioned led to additional questions being asked on that subject. The interview instrument was developed by three researchers and adapted slightly for the individual case studies. The interview instrument consisted of the following main areas:

1. Introduction
2. Personal, team and project background
3. Current state of SCE in the organisation
4. Experiences of the organisational phenomena affecting SCE
5. Ending (uncovered topics)

STUDY 3 was a single case study collecting in-depth experiences of one delayed software project. The case project was selected for three main reasons: First, the project was delayed, which was a precondition for studying the impacts of a delay on SCE and a product launch. Second, the new product was highly expected in the company as it was strategically important. Thus,

Table 3.2: Interviewees of the research.

Study 2: Small Global	Study 2: Large Multinational	Study 2: Tech Giant
Product Owner (KI)	Project Manager (KI)	Program Manager (KI)
Senior Business Manager	Business Manager	Line Manager
	Testing Manager	Senior Manager
Senior Technology Manager	Requirements Engineer	Requirements Engineer
Project Manager	Software Developer	Head of Product Management
		Head of Programs

KI = Key informant for the study, interviewed twice

Table 3.3: Interviewees' roles in the company

Role	Department	Length (min)
CEO, Product Director	CEO, Products	55
Key account manager	Sales	35
Product marketing manager	Marketing	55
Sales director	Sales	35
Sales analyst	Sales	25
Product owner (PO)	Product management	50
Marketing director	Marketing	50

the company made an important investment into the product, which was followed carefully even by the top management, which improves the validity of the results. Third, the researchers were familiar with the company, and the employees were expected to speak honestly, even about difficult topics.

The study design was very similar to Study 2, following the guidelines by Runeson and Höst [119]. The primary data gathering method for this study was semi-structured interviews. In total, we interviewed seven persons and studied 121 meeting minutes and four plan documents. The interviewees and their role descriptions, as well as their departments in the company, are shown in Table 3.3. Prior to the interviews, an interview protocol with the interview instrument was created. The interviews were conducted by two researchers; one acted as the main interviewer and the other took notes. The interviews were recorded and carried out during one week in the beginning of January 2016. All interviews were transcribed, and the results were sent to the interviewees for review. All participated in the study voluntarily, and the company and interviewees wished to remain anonymous. The collected data were treated confidentially. As a secondary data source, we collected different project-related documents such as meeting minutes and marketing and sales plans.

3.4 DATA ANALYSIS

STUDY 1 aimed to establish the level of TMS for SCE and the level of experienced importance of TMS for SCE. Additionally, the correlation between these two and project success was analysed. The primary method for data analysis was statistical analysis. The conducted survey was based on a predefined list of 16 TMS practices for SCE for measuring the current practice and attitudes. The list's validity was evaluated according to three conventional techniques: Cronbach's standardised α, Spearman's rank correlation coefficients and an eigenvalue decomposition. Cronbach's standardised α was computed separately for the practices and attitudes. The internal consistency was studied further by generating Spearman's rho matrices for the use of practices and attitudes. One practice showed negative correlation, which, after a peer discussion, was excluded from the analysis as an unclear proposition. Finally, a scree plot was used to assess the number of latent dimensions in the data [134].

The correlation between the extent of use and experienced importance was measured with Pearson's r for the overall score and individual elements. Also, nonparametric Mann-Whitney U-tests were administered to test the correlation. Project- and respondent-related variables' influences on the use of practices and attitudes were tested with two separate ordinary least squares (OLS) regression models. The basic model setup was similar in both models: the sum variables of extent of use and attitudes were used as dependent variables, while respondent- and project-related background variables were used as independent variables. The tested independent variables were the following: ICT sector, firm size, project size, project duration, project status and estimate meaningfulness evaluation before the project, during the project and after the project. As a second regression modelling approach, a logistic regression model was fitted with a recoded project success variable as the dependent phenomenon. Finally, a nonparametric Kruskal-Wallis rank sum test was used to investigate the effects of the individual 16 items.

STUDY 2 used transcriptions of the interviews, coding of transcripts and documents and grouping the coded pieces of text to make conclusions. The overall process of analysis was conducted as outlined by [119]. During the coding phase, each interview transcript and collected document was reviewed statement by statement, and statements containing information about studied topics were assigned a code representing the findings category. Then, readily coded main categories were reviewed statement by statement to identify subcategories. After a couple of iterations, subcategories emerged from these two approaches. The performed analysis was of the inductive type, meaning that the patterns and categories of the analysis come from the data instead of being pre-defined. The application used for coding (NVivo 10) maintained the evidence trail from the coded pieces of text back to the documents, transcripts and interviewees automatically. After coding the data, the coded statements were grouped together to form initial hypotheses, or candidates, for conclusions. The process progressed iteratively. During the process of forming a hypothesis, interviewees were asked clarifying or additional questions, as necessary, to resolve any confusions and provide additional confidence for the hypothesis. The role of the collected documentation was mostly to provide background information and to support statements made by the interviewees.

STUDY 3 followed the same analysis practice as Study 2. However, while the role of the documentation was mostly related to understanding the background of the studied projects in Study 2, in Study 3, the documentation had an important role in discovering escalation-related practices and patterns and establishing a chronological order of events.

3.5 RELIABILITY AND VALIDITY

STUDY 1 used Cronbach's standardised α, Spearman's rank correlation coefficients and an eigenvalue decomposition for measuring the model's reliability, i.e., how well the list of 16 propositions measure top management participation in SCE. The obtained Cronbach's standardised α values were 0.92 and 0.88 for the extent of use and perceived importance, respectively. This is higher than required by the statistical literature [40]. Spearman's rank correlation coefficients show a clear correlation between the scores for the list items. Finally, the result of the scree plot test was that a single latent construct is present.

The correlation between the two latent constructs, extent of use and experienced importance, is moderate (Pearson's $r = 0.48$). The same also holds for the individual elements. When tested with nonparametric Mann-Whitney U-tests, the null hypothesis of equal distributions holds only for the three practices.

The analysis did not find statistically significant correlations between project- and respondent-related background variables and the sum variables extent of use and experienced importance. Furthermore, the analysis did not discover correlations between the sum variables extent of use and experienced importance and project success. Finally, the nonparametric Kruskal-Wallis rank sum test, which was used to investigate the effects of the individual 16 items on project success, found that only two items had statistically significant effects ($p<0.05$).

STUDY 2 AND STUDY 3 were qualitative studies in which the validity could be evaluated through four aspects of validity: reliability, construct validity, internal validity and external validity [145]. Reliability is concerned with the extent to which the data and analysis are dependent on the specific researchers. We took several measures to improve reliability. First, we prepared research protocols to plan and document the steps of the performed research. Second, the analysis was conducted using a software that acts as a data storage for all documents, keeps records of codings and related text inserts and preserves a full trail of evidence. Finally, all research materials are archived to provide a possibility to assess the reliability of the findings. Although several measures were taken to improve reliability, open-ended questions and related discussions are problematic from the reliability point of view, since the discussions were facilitated by specific researchers whose thinking processes are unique and not replicable.

Construct validity refers to the extent to which operational measures actually represent what the researcher has in mind and what is investigated according to the research questions [145]. Prior to Study 2 and Study 3, a quantitative study (Study 1) was conducted in which the construct validity of the themes addressed in the interview instruments was confirmed. The interview instruments were prepared based on these validated topics and further reviewed by the supervising professor. The interviews were also conducted by two researchers, ensuring that questions were addressed in an in-depth manner and provided insight to draw conclusions on the desired topics. The interviewees were also able to review our transcripts and conclusions to confirm mutual understanding of the topics.

Internal validity is of concern when causal relations are examined [145]. When the researcher investigates whether one factor affects an investigated factor, there is a risk that the investigated factor is also affected by a third factor. If the researcher is not aware of the third factor and/or does not know to what extent it affects the investigated factor, there is a threat to internal validity. Our primary measure to mitigate threats to internal validity is in-depth collection of data. All topics were discussed with the interviewees until we felt that the topic was thoroughly covered and an understanding of the topic was reached. Furthermore, we asked follow-up questions during the analysis phase as necessary. The paper manuscripts were also presented to the interviewees to resolve any possible problems related to the drawn conclusions.

External validity is concerned with the extent to which it is possible to generalise the findings and the extent to which the findings are of interest to other people outside the investigated case [145]. Considering that Study 2 is based on three projects and Study 3 is based on one project, they are exploratory in nature, and because the study topics have not been widely explored prior to these studies, the generalisability of the results is low. However, our studies provide in-depth information and detailed findings of the studied topics, so we believe that the transferability of the results to similar contexts should be fair.

Finally, we took several countermeasures against reactivity, researcher bias and respondent bias-related threats to validity [118]. Reactivity means that the presence of the researcher may influence the study and the behaviour of the study objects. Researcher bias refers to the preconceptions of the researcher that may influence how questions are asked and answers are interpreted. Finally, respondent bias originates from the respondents' attitudes toward the research, which may lead, for example, to withholding information or giving answers the respondents think the researcher is looking for. The countermeasures included prolonged involvement, data source triangulation, observer triangulation, methodological triangulation, theory triangulation peer debriefing, member checking, negative case analysis and audit trail [118] .

4 DISCUSSION OF RESULTS

The following section discusses and summarises the key results of this thesis for both research questions. This chapter also presents the academic and practical implications, addresses limitations of the conducted studies and makes suggestions for future research.

4.1 HOW DOES TOP MANAGEMENT SUPPORT SCE (RQ1)?

The first goal of this thesis was to understand the top management support practices for SCE (i.e., how does top management participate in SCE?). Study 1 and Study 2 addressed this question; the results of Study 1 were reported in Paper I, and the results of Study 2 were reported in Papers IV and V. Study 1 approached the question quantitatively by conducting a survey among Finnish software professionals serving different roles in software projects. The study used a list of 16 predefined practices for measuring TMS in SCE (Table 4.1, practices 1–16). Since TMS in SCE is a sparsely studied area, we identified relevant top management support practices for project management [149, 97] and adapted the identified practices for SCE. The list contained two kinds of practices, tangible actions and expressing understanding and support for SCE. According to employed validity measures, the list properly measured the desired constructs, the extent of use and experiences importance to TMS in SCE.

The results from Study 1 show that 10 out of 16 practices are more often present in estimating projects than absent. Using a scale from 1 to 4, the means for extent of use and experienced importance were 2.54 and 3.04, respectively. Six practices had a mean value lower than 2.5 (the scale midpoint) for extent of use, and all but two practices had a mean above 2.5 for perceived importance. Thus, the results clearly indicate that TMS for SCE would be high and that senior management would contribute to SCE in various ways. However, in Study 2, which was a case study, we used the same list of 16 practices for gaining an in-depth understanding of TMS for SCE. The results were vastly different from the results reported in Paper I. According to the results based on three projects and interviews of 16 software professionals, only five of the 16 practices were found to be exercised in connection to estimation Table 4.1. However, the study identified two additional practices: Top management studies and approves the estimate and Top management ensures adequate resources for estimation. Thus, seven practices in total were associated with TMS for SCE. Study 2 also confirmed that the time and effort invested by top management in SCE was low, and the respondents clearly stated that top management did not participate in estimation directly.

Considering the dissonance in the results between Study 1 and Study 2, a brief discussion is appropriate. In Study 1, our assumption was that the survey would have measured the extent of use of TMS practices as objective facts and the experienced importance as a subjective state [36] and that the results would describe the real-life situation in a generalisable manner. However, the results from Study 2 indicate that the validity of the extent of use-related results from Study 1 may be low for individual practices. Reasons for low validity may be that respondents did not have a good recollection of exercised support practices and have based their answers on 'gut feelings', or they tended to prefer seeing senior management participating in SCE and adjusted their answers accordingly. The possibilities are extensive. Although the validity for individual practices seems low, the overall presence of top management in SCE correlates with the results from Study 2. However, the validity of the results from Study 2 is likely to be higher than the validity of the results from Study 1, as is typical between surveys and case studies [45]. Retrospectively, the order of Study 1 and Study 2 should have been switched: surveys are not optimal for discoveries but are strong for testing hypotheses [8]. Thus, the theories should have been first generated with case study research and followed by verifying surveys [37].

As detailed above, top management participation in SCE is mostly indirect through creating a good environment for successful estimation and demonstrating the importance and understanding of estimation. The findings correlate with previous studies from other fields. Akkermans et al. [3] reported that the presence and attitudes of top management were identified as a root cause for driving performance in ERP implementation. Bingi

Table 4.1: 16 TM support practices for SCE used in Study 1.

	Practice	Confirmed	Category
1.	TM ensures existence of estimation procedures	–	
2.	TM ensures that the estimator has adequate skills	–	
3.	TM ensures improving estimation procedures	–	
4.	TM ensures the involvement of the project manager during the estimation stage	+	Direct participation
5.	TM ensures good communication between the estimator and the organisation	–	
6.	TM ensures that there are criteria for evaluating the meaningfulness of the estimate	–	
7.	TM ensures ongoing estimation skills training programmes	–	
8.	TM requires re-estimating during the project to get more accurate estimates	–	
9.	TM ensures that the estimate relies on documented facts rather than guessing and intuition	–	
10.	The IT executive studies and approves the estimate	–	
11.	TM recognizes that estimates are critical to this organization's success	+	
12.	TM is knowledgeable of estimation procedures	–	
13.	TM understands the consequences of an erroneous estimate to the project success	–	
14.	TM can distinguish between estimates, targets and commitments	–	
15.	TM recognizes that the estimates are inaccurate in the beginning of the project	–	
16.	TM takes the output of an estimate as given without debate	+	
17.	NEW: TM studies and approves the estimate	+	
18.	NEW: TM ensures adequate resources for estimation	+	

(+) indicates that the practice was also supported by Study 2; (–) indicates that no evidence was found in Study 2; NEW indicates that the practice was discovered in Study 2. Supported practices are categorised as outlined in Paper V.

et al. [11] noted that leadership from upper management and support and caring acts of project leaders would boost the morale of the team members and that the top contributor for a successful ERP implementation is strong commitment from upper management. Herington and Peterson [50] said that top management should set the stage in CRM initiatives for leadership, strategic direction and alignment of vision and business goals.

In software process improvement, management is encouraged to take responsibility, discuss SPI often and consider SPI as a method of increasing competitiveness [31]. Top management plays an important role in innovation in providing a supportive organisational climate [4]. In project management, Riaz and Mohamad [1] recognised that provisions of resources, communication, expertise, power and structural arrangements are important dimensions of top management support, having a significant positive relationship with project success. Bingi et al. [11] emphasised top management in situations where the outlay of capital investments is significant, and Stelzer and Mellis [129] reported that support from top management indicates their interest in SPI and the extent to which organisational resources are granted by top management for SPI implementation. All in all, top management employs very few hands-on measures and participation happens on a more strategic level.

To summarise, our studies show that top management participation in SCE is mostly related to resource provisioning, demonstrated importance and goal setting. This corresponds to top management roles in other areas such as project management, ERP or CRM implementation, innovation and SPI. Additionally, we found two direct participation practices that also demonstrate power by nature. Thus, top management support for SCE is generic and strategic by nature and has been found to be typical for managers in earlier studies [25, 137].

4.2 WHAT ARE THE IMPACTS OF TMS FOR SCE (RQ2)?

The second goal of the thesis was to investigate the impacts of top management participation on SCE for a software project. Study 1, Study 2 and Study 3 address this question. The results of Study 1 are reported in Paper I, the results of Study 2 are reported in Papers II, IV and V and the results of Study 3 are reported in Paper III. Study 1 used a quantitative approach for investigating the impact of 16 top management support practices in project success for extent of use and perceived importance of practices. The impact was studied for individual practices and for all practices. According to the results from Study 1, two practices were found to be significant with respect to project success:

1. "Top management ensures that the estimate relies on documented facts rather than guessing and intuition for the extent of use", and

2. "Top management recognises that estimates are critical to this organisation's success" for the experienced importance.

Study 2 approached the research question though a qualitative case study-based research design. The study discovered that top management's actions had very little direct impact on a software project. The exercised practice 'TM studies and approves the estimate' led to re-estimation and improved the estimates. Otherwise, top management actions did not have direct supportive impacts on people- or project-related artefacts. The study also found that top management actions may have direct negative impacts on estimation and software projects. For example, if the project-related expectations ('vision') are not clearly communicated, the project team may aim toward a wrong target and base the estimate on incorrect premises. Furthermore, expressing expectations, e.g., regarding the schedule or effort of a project, may create anchors [5] that can influence estimation. Also, interpreting estimates as commitments resulted in estimators giving upper bound estimates.

Many interviewees also disclosed their negative feelings toward estimation, originating from fear of failure, management interpreting estimates as commitments and requests for 'quick estimates'. Specifically, in the case of a delayed project investigated in Study 3, extending the schedule in small, unrealistic intervals as a result of experienced management pressure had harmful effects in an organisation. Consecutive extensions resulted in lost trust in the estimates and made work coordination difficult. This had widespread negative impacts not only on the project itself but also on market introduction of the project results and on planned benefits to be earned. Thus, it seems easier for top management to hinder successful estimation than to give direct support.

According to our findings, the nature of TMS for SCE is mostly indirect or attitudinal [57] in nature. 'Recognising', 'understanding', 'being able to distinguish' and 'taking without debate' are examples of attitudinal interpretations of TMS. In the scope of our findings, 'provisioning resources' is also attitudinal. Hence, top management's role in supporting SCE is a 'back-seat driver' instead of an 'active participant' [57]. Generally, top management provides an appropriate environment [4] and does not often participate in day-to-day SCE operations. This description of the top management role in SCE corresponds with previous research (e.g., [29, 92]).

While the direct TMS for SCE is low, the positive impact of indirect practices seems to be enough for a successful project outcome. The evidence suggests that these factors support estimation and are even crucial to it, as found in various studies from other areas (e.g., [29]). The importance of the practices was also confirmed by the interviewees and respondents. No results suggested that direct top management actions related to SCE would be necessary for successful estimation or projects. These attitudinal practices can be seen as oppositional to the harmful behaviours mentioned earlier, increasing the chances of successful estimation and improving motivation.

Top management participation may also have significant negative impacts on project and estimation success. The top management actions with negative impacts discovered in our studies are behavioural in nature without systematic or planned long-term patterns. Anchoring, interpreting estimates as commitments and demanding quick estimates were situational events with which top management caused negative impacts. Intuitively, people with power can easily make either good or bad by their actions. Unfortunately, doing harm appears to be easier than doing good, and opportunities for causing harm are practically endless. Buchanan [16] lists consequences of organisational politics to include frustration, anxiety and job dissatisfaction on an individual level and misuse of resources, creating communication barriers and wasting time on an organisational level. Some harmful behaviours are likely to be unintentional, others are intentional [94]. Our research suggests that top management hands-on participation is more likely to do harm than good.

Therefore, the role that top management has assumed in relation to SCE seems appropriate.

4.3 IMPLICATIONS FOR PRACTICE

The objective of this thesis was to explain the role of top management in SCE and to provide suggestions for improving estimation practices through a greater understanding of top management-related reasons for estimation error. The results presented in this thesis are based on three empirical studies: one survey and two in-depth case studies. The studies focused on collecting evidence of top management participation in SCE, specifically tangible top management participation practices and the effects of top management participation on a software project.

The results of Study 2, reported in Papers IV and V, indicate that top management creates prerequisites for successful estimation from the back-seat driver's role without the need for direct participation in estimation itself. I believe that this is an accurate and useful depiction of the reality, indicating that top management makes the best contribution to estimation and project success by understanding the importance and nature of estimations and by providing adequate resources for a reality-seeking and thorough estimation process. We also conclude, based on the evidence, that top management should avoid direct participation in estimation. Due to the power top management holds in organisations, their words and signals are easily interpreted as intentions. When organisations try to align with and comply to these assumed directions, the objectivity and realism of estimation is compromised. Thus, emphasising a supportive and realism-seeking environment with minimal potential bias is a good starting point for SCE improvement work when considering the role of top management.

Retrospectively, and especially in light of previous research, the results presented in this thesis seem unsurprising to me, my fellow researchers and fellow practitioners. However, considering the power of top management in organisations, it may not be easy to discuss their role in relation to SCE or other areas, as conflicts with top management may have consequences for one's career. I hope that this thesis is an eye-opener and facilitates constructive discussion of top management roles in SCE. Optimally, management would understand their place in SCE better and provide support and encouragement. I believe that many organisations would be able to benefit from these insights and improve their estimation accuracy.

4.4 FUTURE RESEARCH

The findings of this thesis draw an unsurprising picture of top management as a grey eminence creating an environment for successful estimation and staying behind the scenes. However, because of the power top management holds, its role is complex and multi-layered. I believe that top management may have a role in relation to some estimation inhibitors presented in Table 2.2. Top management roles and actions in software projects may be a meta-factor behind estimation inhibitors such as "frequent changes", "focus on time-to-delivery rather than cost or quality" or "poor requirement specification". Considering the significant negative effects of top management actions on estimation witnessed in our studies, this seems probable, and understanding top management's role in connection to individual inhibitors could enable improvements in estimation accuracy.

This thesis also shows that the results of surveys and in-depth case study research of the same topic may differ. The survey produced results that could not be validated in case study research. The reason for this discrepancy may be that it is difficult to judge whether a certain proposition in a questionnaire is truly significant in a real-life setting or if it is just a proud thought. I believe that top management-related research cannot be properly conducted without in-depth studies in real-life settings due to the complex nature of TMS. Thus, top management-related research should be primarily conducted via in-depth studies generating in-depth insight and with quantitative methods to validate already proven phenomena.

5 CONCLUSIONS

Software project overruns continue to be a systemic problem in the industry, even after decades of persistent improvement. Thus far, SCE research has focused on seeking reasons for overruns and estimation error from methodologies without much luck. Therefore, it seems likely that the reasons for estimation error originate from other sources. While SCE research focuses on methodologies, many other disciplines have given more attention to managerial issues, especially top management support. As a significant success factor in many areas, TMS has not been studied thoroughly in SCE. Based on empirical evidence, this thesis investigates the role of top management in SCE.

The evidence from our studies shows that top management support for SCE is mostly indirect, as in many other disciplines. Senior management focuses on creating a successful environment for software development and SCE instead of participating in the software process personally. For SCE, the key factors of top management support include adequate resources, demonstrating the importance of SCE and seeking realism. This indirect role is enough for successful estimation. On the other hand, top management may negatively impact estimation with its actions. For example, unclear expectations may cause the project team to aim for the wrong outcome, expressed expectations may become anchors biasing estimation and interpreting estimates as commitments may decrease estimators' motivation and cause them to give high estimates. Because of the power top management holds in organisations, their opinions and wishes easily cloud objective work toward realistic estimates. Some estimation errors inevitably originate from top management actions.

The results show that top management can both support and hinder estimation accuracy. The support is mostly conveyed through attitudinal behaviours such as seeking realism and demonstrating support for SCE, while the hindering factors typically originate from direct actions. The practical implication is that top management should avoid direct participation in, or even contact with, SCE and focus on sustaining a supportive and unbiased environment. By doing this, many projects should be able to avoid failures hurting firms' competitiveness. From the research perspective, the results provide evidence that people-related perspectives are an important factor in SCE, implying that a shifting focus from methodologies toward managerial topics is justified.

This thesis contributes to the practice and theories on SCE by explaining the role of top management in SCE. The primary insight is that top management can focus on creating prerequisites for successful estimation and avoid direct participation. The results also show that the impacts of top management actions typically originate from indirect and/or informal actions. This insight provides a direction for further research. Top management behaviour may act as a meta-factor related to sources of estimation error and should be investigated in more detail. Finally, we noted that survey research may give vastly different, and in our experience unreliable, results compared to in-depth case study research. Top management support seems to be too complex to be adequately understood through structured questionnaires. Thus, I recommend an in-depth approach for future studies.

BIBLIOGRAPHY

[1] Riaz Ahmed and Noor Azmi bin Mohamad. "Exploring the relationship between multi-dimensional top management support and project success: An international study". In: *Engineering Management Journal* 28.1 (2016), pp. 54–67 (cit. on pp. 30, 56).

[2] Jarmo J Ahonen, Paula Savolainen, Helena Merikoski and Jaana Nevalainen. "Reported project management effort, project size, and contract type". In: *Journal of Systems and Software* 109 (2015), pp. 205–213 (cit. on p. 27).

[3] Henk Akkermans and Kees van Helden. "Vicious and virtuous cycles in ERP implementation: a case study of interrelations between critical success factors". In: *European journal of information systems* 11.1 (2002), pp. 35–46 (cit. on p. 54).

[4] Eshaq M Al Shaar, Shadi Ahmed Khattab, Raed Naser Alkaied and Abdelkareem Q Manna. "The effect of top management support on innovation: The mediating role of synergy between organizational structure and information technology". In: *International Review of Management and Business Research* 4.2 (2015), p. 499 (cit. on pp. 56, 58).

[5] Jorge Aranda and Steve Easterbrook. "Anchoring and Adjustment in Software Estimation". In: *Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ESEC/FSE-13. Lisbon, Portugal: ACM, 2005, pp. 346–355. ISBN: 1-59593-014-0. URL: `http://doi.acm.org/10.1145/1081706.1081761` (cit. on pp. 17, 26, 36, 57).

[6] Tharwon Arnuphaptrairong. "Top ten lists of software project risks: Evidence from the literature survey". In: *Proceedings of the International MultiConference of Engineers and Computer Scientists*. Vol. 1. 2011, pp. 1–6 (cit. on p. 30).

[7] Roger Atkinson. "Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria". In: *International journal of project management* 17.6 (1999), pp. 337–342 (cit. on p. 28).

[8] P Attewell and JB Rule. "Survey and other methodologies applied to IT impact research: experiences from a comparative study of business computing". In: *The Information systems research challenge: survey research methods* 3.1 (1991), pp. 299–315 (cit. on p. 54).

[9] E Babbie. "Survey Research Methods. Wadworth Publishing Co". In: *Belmont California* (1990) (cit. on p. 41).

[10] Henri Barki and Jon Hartwick. "Rethinking the concept of user involvement". In: *MIS quarterly* (1989), pp. 53–63 (cit. on p. 29).

[11] Prasad Bingi, Maneesh K Sharma and Jayanth K Godla. "Critical issues affecting an ERP implementation". In: *IS Management* 16.3 (1999), pp. 7–14 (cit. on p. 56).

[12] Barry W Boehm, Ray Madachy, Bert Steece et al. *Software cost estimation with Cocomo II*. Prentice Hall PTR, 2000 (cit. on p. 6).

[13] Barry W Boehm et al. *Software engineering economics*. Vol. 197. Prentice-hall Englewood Cliffs (NJ), 1981 (cit. on pp. 5, 7, 22).

[14] Albert Boonstra. "How do top managers support strategic information system projects and why do they sometimes withhold this support?" In: *International Journal of Project Management* 31.4 (2013), pp. 498–512 (cit. on pp. 30, 31).

[15] Lionel C. Briand and Isabella Wieczorek. "Resource Estimation in Software Engineering". In: *Encyclopedia of Software Engineering*. John Wiley & Sons, Inc., 2002. ISBN: 9780471028956 (cit. on p. 34).

[16] David A Buchanan. "You stab my back, I'll stab yours: Management experience and perceptions of organization political behaviour". In: *British Journal of Management* 19.1 (2008), pp. 49–64 (cit. on pp. 28, 39, 58).

[17] Product Team CMMI. *CMMI for Development, Version 1.3 (CMU/SEI-2010-TR-033). Software Engineering Institute.* 2010 (cit. on p. 3).

[18] Roger J Calantone and C Anthony Di Benedetto. "The role of lean launch execution and launch timing on new product performance". In: *Journal of the Academy of Marketing Science* 40.4 (2012), pp. 526–538 (cit. on p. 20).

[19] Martin Campbell-Kelly. "Development and structure of the international software industry, 1950-1990". In: *Business and economic history* (1995), pp. 73–110 (cit. on p. 3).

[20] Mason A Carpenter, Marta A Geletkanycz and Wm Gerard Sanders. "Upper echelons research revisited: Antecedents, elements, and consequences of top management team composition". In: *Journal of management* 30.6 (2004), pp. 749–778 (cit. on p. 40).

[21] Narciso Cerpa and June M Verner. "Why did your project fail?" In: *Communications of the ACM* 52.12 (2009), pp. 130–134 (cit. on p. 9).

[22] Jiyao Chen, Richard R Reilly and Gary S Lynn. "The impacts of speed-to-market on new product success: the moderating effects of uncertainty". In: *IEEE Transactions on engineering management* 52.2 (2005), pp. 199–212 (cit. on p. 33).

[23] Deborah R Compeau and Christopher A Higgins. "Computer self-efficacy: Development of a measure and initial test". In: *MIS quarterly* (1995), pp. 189–211 (cit. on p. 30).

[24] T. Cooke-Davies. "The 'real' success factors on projects". In: *International journal of project management* 20.3 (2002), pp. 185–190 (cit. on p. 9).

[25] Lynn Crawford. "Senior management perceptions of project management competence". In: *International journal of project management* 23.1 (2005), pp. 7–16 (cit. on p. 56).

[26] Richard M Cyert, James G March et al. "A behavioral theory of the firm". In: *Englewood Cliffs, NJ* 2.4 (1963), pp. 169–187 (cit. on p. 40).

[27] *DIMECC N4S-Program: Finnish Software Companies Speeding Digital Economy*. 2017. URL: http://n4s.dimecc.com/en/ (visited on 14/11/2018) (cit. on p. 41).

[28] C Anthony Di Benedetto. "Identifying the key success factors in new product launch". In: *Journal of Product Innovation Management* 16.6 (1999), pp. 530–544 (cit. on p. 20).

[29] Linying Dong, Derrick Neufeld and Chris Higgins. "Top management support of enterprise systems implementations". In: *Journal of Information technology* 24.1 (2009), pp. 55–80 (cit. on pp. 28, 30, 58).

[30] Yogesh K Dwivedi, David Wastell, Sven Laumer, Helle Zinner Henriksen, Michael D Myers, Deborah Bunker, Amany Elbanna, MN Ravishankar and Shirish C Srivastava. "Research on information systems failures and successes: Status update and future directions". In: *Information Systems Frontiers* 17.1 (2015), pp. 143–157 (cit. on pp. 3, 7, 20).

[31] Tore Dybå. "An empirical investigation of the key factors for success in software process improvement". In: *IEEE Transactions on Software Engineering* 31.5 (2005), pp. 410–424 (cit. on p. 56).

[32] Khaled El Emam and A Günes Koru. "A replicated survey of IT software project failures". In: *IEEE software* 5 (2008), pp. 84–90 (cit. on p. 20).

[33] Robert G Fichman, Brian L Dos Santos and Zhiqiang Eric Zheng. "Digital innovation as a fundamental and powerful concept in the information systems curriculum." In: *MIS quarterly* 38.2 (2014) (cit. on p. 20).

[34] Forbes. *The World's Largest Public Companies 2017*. 2018. URL: https://www.forbes.com/global2000/list/ (visited on 11/11/2018) (cit. on p. 3).

[35] Joyce Fortune and Diana White. "Framing of project critical success factors by a systems model". In: *International Journal of Project Management* 24.1 (2006), pp. 53–65. ISSN: 0263-7863 (cit. on pp. 10, 35).

[36] Floyd J Fowler Jr. *Survey research methods*. Sage publications, 2013 (cit. on p. 54).

[37] Guy G Gable. "Integrating case study and survey research methods: an example in information systems". In: *European journal of information systems* 3.2 (1994), pp. 112–126 (cit. on p. 54).

[38] Dan Galorath. "Software Project Failure Costs Billions". In: *Better Estimation & Planning Can Help.[online] Available at: http://www. galorath. com/wp/software-project-failure-costs-billions-betterestimation-planning-can-help. php [Accessed 16 September 2014]* (2012) (cit. on pp. 3, 20).

[39] Inc. Gartner. *Gartner Says Global IT Spending to Reach $3.7 Trillion in 2018*. 2017. URL: https://www.gartner.com/en/newsroom/press-releases/2017-10-03-gartner-says-global-it-spending-to-reach-3-trillion-in-2018 (visited on 12/11/2018) (cit. on p. 19).

[40] Darren George and M Mallery. *Using SPSS for Windows step by step: a simple guide and reference*. Boston, MA: Allyn & Bacon, 2003 (cit. on p. 48).

[41] Robert L Glass, Johann Rost and Matthias S Matook. "Lying on software projects". In: *IEEE software* 25.6 (2008), pp. 90–95 (cit. on p. 27).

[42] EM Goldratt. *Critical chain: A business novel.[sl] North River Press Great Barrington*. 1997 (cit. on p. 22).

[43] James Grenning. "Planning poker or how to avoid analysis paralysis while release planning". In: *Hawthorn Woods: Renaissance Software Consulting* 3 (2002), pp. 22–23 (cit. on p. 7).

[44] Torleif Halkjelsvik and Magne Jørgensen. "From origami to software development: A review of studies on judgment-based predictions of performance time." In: *Psychological bulletin* 138.2 (2012), p. 238 (cit. on pp. 3, 7).

[45] Martyn Hammersley. *Routledge Revivals: What's Wrong With Ethnography?(1992): Methodological Explorations*. Routledge, 2018 (cit. on p. 54).

[46] Nukhet Harmancioglu, Cornelia Droge and Roger J Calantone. "Strategic fit to resources versus NPD execution proficiencies: what are their roles in determining success?" In: *Journal of the Academy of Marketing Science* 37.3 (2009), pp. 266–282 (cit. on p. 20).

[47] Michael Haug, Eric W Olsen and Luisa Consolini. *Software quality approaches: testing, verification, and validation: software best practice 1*. Vol. 1. Springer Science & Business Media, 2001 (cit. on p. 20).

[48] Fred J Heemstra. "Software cost estimation". In: *Information and software technology* 34.10 (1992), pp. 627–639 (cit. on p. 25).

[49] Kevin B Hendricks and Vinod R Singhal. "Delays in new product introductions and the market value of the firm: The consequences of being late to the market". In: *Management Science* 43.4 (1997), pp. 422–436 (cit. on pp. 3, 20, 33).

[50] D Herington and G Peterson. "Making sense of e-CM: setting the strategic agenda for sales automation". In: *Proceedings of DCI Customer Relationship Management Conference, Boston, MA*. 2000, pp. 27–29 (cit. on p. 56).

[51] Jairus Hihn and Hamid Habib-agahi. "Cost estimation of software intensive projects: A survey of current practices". In: *Proceedings of the 13th international conference on Software engineering*. IEEE Computer Society Press. 1991, pp. 276–287 (cit. on pp. 9, 25, 35).

[52] J Hill, LC Thomas and DE Allen. "Experts' estimates of task durations in software development projects". In: *International journal of project management* 18.1 (2000), pp. 13–21 (cit. on p. 8).

[53] D Laurie Hughes, Yogesh K Dwivedi, Antonis C Simintiras and Nripendra P Rana. *Success and failure of IS/IT projects: A state of the art analysis and future directions*. Springer, 2015 (cit. on pp. 3, 7, 20).

[54] IPMA, ICB. "IPMA Competence Baseline Version 3". In: *Nijkerk, The Netherlands* (2006) (cit. on p. 3).

[55] ISO. *Quality management principles*. 2018. URL: `https://www.iso.org/files/live/sites/isoorg/files/archive/pdf/en/pub100080.pdf` (visited on 13/11/2018) (cit. on p. 28).

[56] Project Management Institute. *A Guide to the Project Management Body of Knowledge: PMBOK Guide*. PMBOK® Guide Series. Project Management Institute, 2013. ISBN: 9781935589679. URL: `https://books.google.fi/books?id=FpatMQEACAAJ` (cit. on p. 3).

[57] Sirkka L Jarvenpaa and Blake Ives. "Executive involvement and participation in the management of information technology". In: *MIS quarterly* (1991), pp. 205–227 (cit. on pp. 29, 58).

[58] Randall Jensen. "An improved macrolevel software development resource estimation model". In: *5th ISPA Conference*. 1983, pp. 88–92 (cit. on p. 7).

[59] Capers Jones. *Assessment and control of software risks*. Yourdon Press, 1994 (cit. on p. 26).

[60] Magne Jørgensen. "A review of studies on expert estimation of software development effort". In: *Journal of Systems and Software* 70.1 (2004), pp. 37–60 (cit. on p. 26).

[61] Magne Jørgensen. "Forecasting of software development work effort: Evidence on expert judgement and formal models". In: *International Journal of Forecasting* 23.3 (2007), pp. 449–462 (cit. on p. 7).

[62] Magne Jørgensen. "Communication of software cost estimates". In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. ACM. 2014, p. 28 (cit. on p. 25).

[63] Magne Jørgensen, Barry Boehm and Stan Rifkin. "Software development effort estimation: Formal models or expert judgment?" In: *IEEE software* 26.2 (2009), pp. 14–19 (cit. on p. 7).

[64] Magne Jørgensen and Gunnar J Carelius. "An empirical study of software project bidding". In: *IEEE Transactions on Software Engineering* 30.12 (2004), pp. 953–969 (cit. on pp. 8, 26).

[65] Magne Jørgensen, Bjørn Faugli and Tanja Gruschke. "Characteristics of software engineers with optimistic predictions". In: *Journal of Systems and Software* 80.9 (2007), pp. 1472–1482 (cit. on pp. 8, 26).

[66] Magne Jørgensen and Stein Grimstad. "Avoiding Irrelevant and Misleading Information When Estimating Development Effort". In: *IEEE Softw.* 25.3 (May 2008), pp. 78–83. ISSN: 0740-7459. URL: `http://dx.doi.org/10.1109/MS.2008.57` (cit. on pp. 8, 26, 36).

[67] Magne Jørgensen and Martin Shepperd. "A Systematic Review of Software Development Cost Estimation Studies". In: *IEEE Transactions of Software Engineering* (2007) (cit. on pp. 3, 7, 8, 27, 34).

[68] Magne Jørgensen and Martin Shepperd. "A Systematic Review of Software Development Cost Estimation Studies". In: *IEEE Transactions of Software Engineering* 33.1 (2007), pp. 33–53 (cit. on pp. 7, 9, 34, 35).

[69] Magne Jørgensen and Dag IK Sjøberg. "Impact of effort estimates on software project work". In: *Information and software technology* 43.15 (2001), pp. 939–948 (cit. on pp. 26, 36).

[70] Magne Jørgensen and Dag IK Sjøberg. "Impact of experience on maintenance skills". In: *Journal of Software Maintenance and Evolution: research and practice* 14.2 (2002), pp. 123–146 (cit. on pp. 8, 26).

[71] Magne Jørgensen and Dag IK Sjøberg. "The impact of customer expectation on software development effort estimates". In: *International Journal of Project Management* 22.4 (2004), pp. 317–325 (cit. on p. 8).

[72] Magne Jørgensen and Dag IK Sjøberg. "The impact of customer expectation on software development effort estimates". In: *International Journal of Project Management* 22.4 (2004), pp. 317–325 (cit. on pp. 8, 26).

[73] Magne Jørgensen, Karl Halvor Teigen and Kjetil MoløKken. "Better sure than safe? Over-confidence in judgement based software development effort prediction intervals". In: *Journal of systems and software* 70.1-2 (2004), pp. 79–93 (cit. on p. 8).

[74] Leon A Kappelman, Robert McKeeman and Lixuan Zhang. "Early warning signs of IT project failure: The dominant dozen". In: *Information systems management* 23.4 (2006), pp. 31–36 (cit. on p. 33).

[75] Daniel Karlstrom, Per Runeson and Claes Wohlin. "Aggregating viewpoints for strategic software process improvement-a method and a case study". In: *IEE Proceedings-Software* 149.5 (2002), pp. 143–152 (cit. on p. 43).

[76] Mark Keil. "Pulling the plug: Software project management and the problem of project escalation". In: *MIS quarterly* (1995), pp. 421–447 (cit. on p. 32).

[77] Mark Keil and Daniel Robey. "Turning around troubled software projects: An exploratory study of the deescalation of commitment to failing courses of action". In: *Journal of Management Information Systems* 15.4 (1999), pp. 63–87 (cit. on p. 33).

[78] Mark Keil, Duane P Truex and Richard Mixon. "The effects of sunk cost and project completion on information technology project escalation". In: *IEEE Transactions on Engineering Management* 42.4 (1995), pp. 372–381 (cit. on p. 33).

[79] Mark Keil, Bernard CY Tan, Kwok-Kee Wei, Timo Saarinen, Virpi Tuunainen and Arjen Wassenaar. "A cross-cultural study on escalation of commitment behavior in software projects". In: *MIS quarterly* (2000), pp. 299–325 (cit. on p. 33).

[80] Barbara Kitchenham, Shari Lawrence Pfleeger, Beth McColl and Suzanne Eagan. "An empirical study of maintenance and development estimation accuracy". In: *Journal of systems and software* 64.1 (2002), pp. 57–77 (cit. on p. 8).

[81] Heinz K Klein and Michael D Myers. "A set of principles for conducting and evaluating interpretive field studies in information systems". In: *MIS quarterly* (1999), pp. 67–93 (cit. on p. 44).

[82] Børn Johs. Kolltveit, Jan Terje Karlsen and Kjell K Grønhaug. "Perspectives on project management". In: *International Journal of Project Management* 25.1 (Jan. 2007), pp. 3–9 (cit. on pp. 34, 36).

[83] Luiz A. Laranjeira. "Software size estimation of object-oriented systems". In: *IEEE Transactions on software engineering* 16.5 (1990), pp. 510–522 (cit. on p. 23).

[84] Albert L Lederer and Jayesh Prasad. "The validation of a political model of information systems development cost estimating". In: *Proceedings of the 1991 conference on SIGCPR*. ACM. 1991, pp. 164–173 (cit. on pp. 26, 27, 36).

[85] Albert L. Lederer and Jayesh Prasad. "Nine Management Guidelines for Better Cost Estimating". In: *Communications of the ACM* 35.2 (Feb. 1992), pp. 51–59. ISSN: 0001-0782 (cit. on pp. 9, 24, 25).

[86] Albert L Lederer and Jayesh Prasad. "Causes of inaccurate software development cost estimates". In: *Journal of systems and software* 31.2 (1995), pp. 125–134 (cit. on p. 10).

[87] Albert L Lederer and Jayesh Prasad. "Causes of inaccurate software development cost estimates". In: *Journal of systems and software* 31.2 (1995), pp. 125–134 (cit. on pp. 23, 25, 27).

[88] Albert L. Lederer and Jayesh Prasad. "Software management and cost estimating error". In: *Journal of Systems and Software* 50.1 (2000), pp. 33–42 (cit. on p. 26).

[89] Bryan A Lukas and Ajay Menon. "New product quality: intended and unintended consequences of new product development speed". In: *Journal of Business Research* 57.11 (2004), pp. 1258–1264 (cit. on pp. 21, 33).

[90] L.F. Luna-Reyes, J. Zhang, J.R. Gil-Garćia and A.M Cresswell. "Software developments development as emergent socio-technical change: a practice approach". In: *European Journal of Software developments* 14.1 (2005), pp. 93–105 (cit. on pp. 10, 35).

[91] Kalle Lyytinen and Daniel Robey. "Learning failure in information systems development". In: *Information Systems Journal* 9.2 (1999), pp. 85–101 (cit. on p. 33).

[92] Ochini Madanayake, Shirley Gregor, Colleen Hayes and Steven Fraser. "What we need: Project managers' evaluation of top management actions required for software development projects." In: *ECIS*. 2009, pp. 1247–1258 (cit. on pp. 30, 31, 58).

[93] Ana Magazinius. ”Exploring Software Cost Estimation Inaccuracy – Informal Use of Estimates”. PhD thesis. Chalmers University of Technology and Göteborg University, 2012 (cit. on pp. 21, 29).

[94] Ana Magazinius, Sofia Börjesson and Robert Feldt. ”Investigating intentional distortions in software cost estimation — An exploratory study”. In: *Journal of Systems and Software* 85.8 (Feb. 2012), pp. 1770–1781 (cit. on pp. 8–10, 26, 27, 36, 59).

[95] Ana Magazinovic and Joakim Pernstål. ”Any other cost estimation inhibitors?” In: *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*. ACM. 2008, pp. 233–242 (cit. on pp. 8, 10, 27).

[96] Steve McConnell. *Rapid development, 1996* (cit. on p. 22).

[97] Steve McConnell. *Software Estimation: Demystifying the Black Art*. Redmond, Washington, USA: Microsoft Press, 2006 (cit. on pp. 3, 6, 7, 15, 21, 22, 25, 34, 53).

[98] Laurie McLeod and Stephen G MacDonell. ”Factors that affect software systems development project outcomes: A survey of research”. In: *ACM Computing Surveys (CSUR)* 43.4 (2011), p. 24 (cit. on p. 9).

[99] Laurie McLeod and Stephen G MacDonell. ”Factors that affect software systems development project outcomes: A survey of research”. In: *ACM Computing Surveys (CSUR)* 43.4 (2011), p. 24 (cit. on p. 31).

[100] John McManus and Trevor Wood-Harper. ”A study in project failure”. In: *British Computer Society* (2008) (cit. on pp. 3, 20).

[101] Kjetil Moløkken and Magne Jørgensen. ”A Review of Software Surveys on Software Effort Estimation”. In: *Proceedings of International Symposium on Empirical Software Engineering*. IEEE, 2003, pp. 220–230 (cit. on pp. 3, 6, 20, 21).

[102] Kjetil Moløkken and Magne Jørgensen. "Expert estimation of web-development projects: are software professionals in technical roles more optimistic than those in non-technical roles?" In: *Empirical Software Engineering* 10.1 (2005), pp. 7–30 (cit. on p. 25).

[103] Burt Nanus and Leonard Farr. "Some Cost Contributors to Large-scale Programs". In: *Proceedings of the April 21-23, 1964, Spring Joint Computer Conference*. AFIPS '64 (Spring). Washington, D.C.: ACM, 1964, pp. 239–248 (cit. on p. 34).

[104] Mohd Hairul Nizam Nasir and Shamsul Sahibuddin. "Critical success factors for software projects: A comparative study". In: *Scientific research and essays* 6.10 (2011), pp. 2174–2186 (cit. on p. 9).

[105] E. A. Nelson. *Management Handbook for the Estimation of Computer Programming Costs*. Technical Documentary Report No. ESD-TR-67-66. Bedford, Massachusetts: United States Air Force, 2002 (cit. on p. 34).

[106] R Ryan Nelson. "IT project management: infamous failures, classic mistakes, and best practices". In: *MIS Quarterly Executive* 6.2 (2007), pp. 67–78 (cit. on pp. 3, 20).

[107] Teri Okoro. "Diverse Talent: Enhancing Gender Participation in Project Management". In: *Procedia — Social and Behavioral Sciences* 226 (2016). Proceedings of the 29th IPMA World Congress WC2015 (28-30 September – 1 October, Panama), pp. 170–175. ISSN: 1877-0428 (cit. on pp. 10, 35).

[108] Wanda J Orlikowski and Jack J Baroudi. "Studying information technology in organizations: Research approaches and assumptions". In: *Information systems research* 2.1 (1991), pp. 1–28 (cit. on p. 39).

[109] C Northcote Parkinson. *Parkinson's Law on the Pursuit of Progress*. Penguin Books, 1968 (cit. on p. 22).

[110] Andrew M Pettigrew. "On studying managerial elites". In: *The Value Creating Board*. Routledge, 2008, pp. 179–207 (cit. on p. 40).

[111] Shari Lawrence Pfleeger. "Experimental design and analysis in software engineering, part 5: analyzing the data". In: *ACM SIGSOFT Software Engineering Notes* 20.5 (1995), pp. 14–17 (cit. on p. 43).

[112] Dien Phan, Douglas Vogel and Jay Nunamaker. "The search for perfect project management". In: *Computerworld* 22.39 (1988), pp. 95–100 (cit. on pp. 25–27, 36).

[113] Bill Pitterman. "Telcordia Technologies: The Journey to High Maturity". In: *IEEE Software* 17.4 (July 2000), pp. 89–96. ISSN: 0740-7459 (cit. on pp. 9, 36).

[114] PoPo Poon and Christian Wagner. "Critical success factors revisited: success and failure cases of information systems for senior executives". In: *Decision support systems* 30.4 (2001), pp. 393–418 (cit. on p. 32).

[115] Lawrence H Putnam and Ware Myers. *Five core metrics: the intelligence behind successful software management*. Pearson Education, 2013 (cit. on pp. 9, 21, 36).

[116] PwC. *Companies shifting more RD spending away from physical products to software and services: 2016 Global Innovation 1000 Study*. 2016. URL: `http://www.pwc.com/us/en/press-releases/2016/pwc-2016-global-innovation-1000-study-press-release.html` (visited on 12/11/2018) (cit. on pp. 3, 19).

[117] PwC. *The 2018 Global Innovation 1000 study*. 2017. URL: `https://www.strategyand.pwc.com/innovation1000` (visited on 12/11/2018) (cit. on p. 19).

[118] Colin Robson. *Real world research: A resource for social scientists and practitioner-researchers*. Second edition. Oxford, UK: Blackwell Publishing, 2002 (cit. on pp. 39, 44, 50, 51).

[119] Per Runeson and Martin Höst. "Guidelines for conducting and reporting case study research in software engineering". In: *Empirical Software Engineering* 14 (2 2009), pp. 131–164. ISSN: 1382-3256. URL: `http://dl.acm.org/citation.cfm?id=1519313.1519324` (cit. on pp. 44, 46, 48).

[120] Jeffrey B Schmidt. "Gate decisions: The key to managing risk during new product development". In: *The PDMA handbook of new product development* (2004), pp. 337–348 (cit. on p. 33).

[121] Jeffrey B Schmidt and Roger J Calantone. "Are really new product development projects harder to shut down?" In: *Journal of Product Innovation Management* 15.2 (1998), pp. 111–123 (cit. on p. 33).

[122] Roy Schmidt, Kalle Lyytinen and Paul Cule Mark Keil. ”Identifying software project risks: An international Delphi study”. In: *Journal of management information systems* 17.4 (2001), pp. 5–36 (cit. on p. 10).

[123] Sumeet Kaur Sehra, Yadwinder Singh Brar, Navdeep Kaur and Sukhjit Singh Sehra. ”Research Patterns and Trends in Software Effort Estimation”. In: *Information and Software Technology* 91 (2017), pp. 1–21. ISSN: 0950-5849 (cit. on p. 34).

[124] Rajeev Sharma and Philip Yetton. ”The contingent effects of management support and task interdependence on successful information systems implementation”. In: *MIS quarterly* (2003), pp. 533–556 (cit. on p. 30).

[125] Aurmi J Shrnhur, O Levy and D Dvir. ”Mapping the dimensions of project success”. In: *Project management journal* 28.2 (1997), pp. 5–13 (cit. on p. 28).

[126] H Jeff Smith and Mark Keil. ”The reluctance to report bad news on troubled software projects: a theoretical model”. In: *Information Systems Journal* 13.1 (2003), pp. 69–95 (cit. on p. 33).

[127] Michael A Stanko, Francisco-Jose Molina-Castillo and Jose-Luis Munuera-Aleman. ”Speed to market for innovative products: blessing or curse?” In: *Journal of Product Innovation Management* 29.5 (2012), pp. 751–765 (cit. on p. 20).

[128] Barry M Staw and Jerry Ross. ”Behavior in escalation situations: Antecedents, prototypes, and solutions.” In: *Research in organizational behavior* (1987) (cit. on p. 32).

[129] Dirk Stelzer and Werner Mellis. ”Success factors of organizational change in software process improvement”. In: *Software Process: Improvement and Practice* 4.4 (1998), pp. 227–250 (cit. on p. 56).

[130] Strategy+Business. *How Non-Tech Companies Are Joining the Software Revolution – in Pictures*. 2016. URL: `https://www.strategy-business.com/pictures/How-Non-Tech-Companies-Are-Joining-the-Software-Revolution-in-Pictures` (visited on 12/11/2018) (cit. on p. 19).

[131] Strategy+Business. *Software-as-a-Catalyst*. 2016. URL: `https://www.strategy-business.com/feature/Software-as-a-Catalyst` (visited on 12/11/2018) (cit. on p. 19).

[132] Richard D Stutzke. *Estimating software-intensive systems: projects, products, and processes*. Pearson Education, 2005 (cit. on p. 23).

[133] Girish H Subramanian and Steven Breslawski. "An empirical analysis of software effort estimate alterations". In: *Journal of Systems and Software* 31.2 (1995), pp. 135–141 (cit. on p. 24).

[134] Barbara G Tabachnick and Linda S Fidell. "Principal components and factor analysis". In: *Using multivariate statistics* 4 (2001), pp. 582–633 (cit. on p. 47).

[135] David J Teece. "Business models, business strategy and innovation". In: *Long range planning* 43.2-3 (2010), pp. 172–194 (cit. on p. 20).

[136] The Standish Group International. *The CHAOS Manifesto: Think Big and Act Small*. 2013 (cit. on p. 3).

[137] Janice Thomas, C Delisle, Kam Jugdev and Pamela Buckle. "Selling project management to senior executives: the case for avoiding crisis sales?" In: *Project Management Journal* 33.2 (2002), pp. 19–28 (cit. on p. 56).

[138] James YL Thong, Chee-Sing Yap and KS Raman. "Top management support, external expertise and information systems implementation in small businesses". In: *Information systems research* 7.2 (1996), pp. 248–267 (cit. on p. 30).

[139] M. Van Genuchten. "Why is software late? An empirical study of reasons for delay in software development". In: *IEEE Transactions on Software Engineering* 17.6 (1991), pp. 582–590 (cit. on pp. 23, 24, 26).

[140] Mark Vigder and Anatol W Kark. *Software cost estimation and control*. Citeseer, 1994 (cit. on pp. 4, 5).

[141] Diana White and Joyce Fortune. "Current practice in project management – An empirical study". In: *International journal of project management* 20.1 (2002), pp. 1–11 (cit. on p. 9).

[142] Diana White and Joyce Fortune. "Current practice in project management – An empirical study". In: *International journal of project management* 20.1 (2002), pp. 1–11 (cit. on pp. 30, 35).

[143] Margarethe F Wiersema and Karen A Bantel. "Top management team demography and corporate strategic change". In: *Academy of Management journal* 35.1 (1992), pp. 91–121 (cit. on pp. 28, 40).

[144] Wikipedia. *List of failed and overbudget custom software projects*. 2018. URL: `https://en.wikipedia.org/wiki/List_of_failed_and_overbudget_custom_software_projects` (visited on 12/11/2018) (cit. on p. 20).

[145] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012 (cit. on pp. 40, 49, 50).

[146] Robert K. Yin. *Case Study Research: Design and Methods*. Third edition. Thousands Oaks, California: SAGE Publications, Inc., 2003 (cit. on pp. 42, 44).

[147] R. Young and E. Jordan. "Top management support: Mantra or necessity?" In: *International journal of project management* 26.7 (2008), pp. 713–725 (cit. on pp. 10, 30, 32, 35).

[148] Ofer Zwikael. "Top management involvement in project management: A cross country study of the software industry". In: *International Journal of Managing Projects in Business* 1.4 (2008), pp. 498–511. eprint: `http://dx.doi.org/10.1108/17538370810906228` (cit. on pp. 9, 10, 15, 31, 35).

[149] Ofer Zwikael. "Top management involvement in project management: Exclusive support practices for different project scenarios". In: *International Journal of Managing Projects in Business* 1.3 (2008), pp. 387–403 (cit. on pp. 9, 10, 31, 53).

Part II

ORIGINAL PUBLICATIONS

PI

TOP MANAGEMENT SUPPORT IN SOFTWARE COST ESTIMATION: A STUDY OF ATTITUDES AND PRACTICE IN FINLAND

By Jurka Rahikkala & et al

Originally published in *International Journal of Managing Projects in Business 8.3 (2015): 513-532.*

Top management support in software cost estimation

A study of attitudes and practice in Finland

Jurka Rahikkala
Vaadin Ltd, Turku, Finland, and
Ville Leppänen, Jukka Ruohonen and Johannes Holvitie
Department of Information Technology, University of Turku, Turku, Finland

Received 1 November 2014
Revised 22 January 2015
Accepted 24 January 2015

Abstract

Purpose – A cost estimate is considered to have a high impact on software project success. Because of this, different methodologies for creating an accurate estimate have been studied over decades. Many methodologies produce accurate results, when used properly. However, software projects still suffer from inaccurate estimates. The disparity may result from organisational hindrances. This paper focuses on top management support (TMS) for software cost estimation (SCE). The purpose of this paper is to identify current practices and attitudes of top management involvement in SCE, and to analyse the relationship between these two and project success.

Design/methodology/approach – A list of 16 TMS practices for SCE has been developed. A survey was conducted to capture the frequency of use and the experienced importance of support practices. Data has been collected from 114 software professionals in Finland. Correlations between the frequency of use, attitudes and project success were analysed.

Findings – Top management invests a significant amount of attention in SCE. The extent of use and experienced importance do not correlate strongly with each other or project success.

Research limitations/implications – The results may lack generalisability. Researchers are encouraged to validate the results with further studies.

Practical implications – Software professionals invite senior managers to participate in SCE. A list of practices for participating is provided.

Originality/value – This paper suggests a list of 16 TMS practices for SCE. The paper also reports on the extent of use and experienced importance of practices, and the correlations between these two and project success.

Keywords Senior management, Software cost estimation, Support practices

Paper type Research paper

1. Introduction

Projects exceeding their budgets and schedules are a widely recognised problem in the software industry (Standish Group, 1994; Moløkken and Jørgensen, 2003; Lederer and Prasad, 1992). A project delivering the desired results within the planned budget and schedule is an exception, not the rule. The consequences of this phenomenon are dramatic, resulting in hundreds of billions of euros in losses annually (Ewusi-Mensah, 2003; Charette, 2005). Considering the severity of the problem and its consequences, it is not surprising that the problem has been extensively studied from several angles, including project management (PM) and software cost estimation (SCE) (Van Genuchten, 1991; Lederer and Prasad, 1995).

Software projects are difficult endeavours. So, a project should be well organised in order to have the premises to succeed. The PM literature has identified a wide spectrum

The authors thank Digile Oy and PMI Finland chapter for providing mailing lists for the survey. Thanks also go to the whole Vaadin team for numerous valuable comments and suggestions.

International Journal of Managing Projects in Business
Vol. 8 No. 3, 2015
pp. 513-532

1753-8378
DOI 10.1108/IJMPB-11-2014-0076

of critical support factors (CSF), which contribute positively to project success (Fortune and White, 2006; Cooke-Davies, 2002). One of the most important CSFs is top management support (TMS) (Fortune and White, 2006; Young and Jordan, 2008; Schmidt *et al.*, 2001). The PM literature has also identified own specific CSFs for TMS, guiding how senior management should use its valuable time to support project success (Zwikael, 2008a, b; Boonstra, 2013).

The purpose of SCE is to provide meaningful information to support decision making (McConnell, 2006). Too high or too low estimates have a biasing effect on the business case, meaning that projects may be approved or rejected on false premises. Also, during the project phase, working towards incorrect goals can have negative effects on the project (Jørgensen and Sjøberg, 2001). Therefore the scientists and practitioners have developed an almost endless plethora of different methods for creating cost estimates (McConnell, 2006; Jørgensen and Shepperd, 2007). It has also been proven that when used properly, these methods will constantly produce adequately good results (Putnam and Myers, 2003; Pitterman, 2000).

The necessary information for delivering a successful software project seems to exist, yet most projects fail to meet the set expectations. The possible reasons are many, including leadership related problems. As a matter of fact, even 60 per cent of senior executives perceive that organisational issues are more important than the technical ones in information system development (Doherty and King, 2001). A logical conclusion would be that top management would promote and enforce the use of best practices both in PM and SCE. The literature clearly shows that TMS has been studied in the scope of PM, but surprisingly not in the scope of SCE.

The first objective of this paper is to develop a list of TMS practices for SCE, which is then used for studying TMS in SCE further. The developed list is based on PM CSFs (Zwikael, 2008a, b), which are adopted in the SCE scope, and selected best practices presented in the SCE literature (McConnell, 2006), which can reasonably be assumed to have a positive result in the SCE outcome. The other objectives of this paper are: to investigate to what extent and how top management is involved in SCE, to study what is the perceived importance of presented support practices and to analyse the relationships between the extent of use, perceived importance and project success. Understanding better the extent of top management involvement and perceived importance of different support practices can help senior managers, project managers and researchers to justify paying more attention on organisational issues related to SCE, and specifically on TMS.

1.1 SCE maturity

Software projects are often late or do not finish at all (Standish Group, 1994). There is not one reason for being late, but it is a result of many reasons, including PM related issues and non-meaningful estimates (Van Genuchten, 1991; Lederer and Prasad, 1995). When considering estimation related problems, it is not likely that failing cost estimates would be a result of lack of suitable estimation methods. For example, Jørgensen lists 12 different estimation approach categories (Jørgensen and Shepperd, 2007), and the number of actual estimation techniques is measured probably in hundreds. Regardless of the modest track record of SCE, projects also do succeed and deliver the wanted results on time, within budget and scope (Putnam and Myers, 2003; Pitterman, 2000). There is also evidence that some organisations succeed constantly, proving that when used appropriately, estimation and PM can produce accurate and desired results (Putnam and Myers, 2003; Pitterman, 2000). On the other hand, even as many as 83 per cent of estimators seem not to use any estimation methodologies, or they rely on ad-hoc or memory-based methodologies (Hihn and Habib-agahi, 1991). This is proven to be connected to budget and schedule overruns (Lederer and Prasad, 1992). When considering PM, White and Fortune reported that only 5 per cent of all organisations do not use any PM methodologies (White and Fortune, 2002), and it has been found that the maturity of an organisation is highly correlated with project success (Harter *et al.*, 2000).

Although the number of estimation methods is more than sufficient, the overall organisational maturity of making SCE seems not to be on the same level as the maturity of executing projects. This is surprising, because both the estimate and the project execution influence the project success (Van Genuchten, 1991; Lederer and Prasad, 1995). TMS has been studied a lot in PM. It has been found that, e.g. paying attention to choosing the right project manager and ensuring that the appropriate PM tools and methodologies are in place contribute positively on project success (Zwikael, 2008a, b). However, the literature does not provide any advice to how senior management should support SCE. Actually, it has been found that no practice discourages the use of guessing as an estimating method (Lederer and Prasad, 2000). This disparity may indicate that the difference between organisations' estimation maturity and PM maturity is a result of differences in the management focus.

1.2 Top management involvement in PM

The primary duty and responsibility of a project manager is to make sure that the project delivers on time, cost and scope (PMI, 2013). However, the senior management is accountable for appointing project managers, making sure that the work is performed as a project in the first place, and ensuring that the PM maturity and methodology are on an adequate level. When a project has been established, the project manager and the project itself need support from top management. Senior management, e.g. provides direction for the project and helps in resolving any issues or conflicts. Senior management also influences PM capabilities and the performance of an organisation through TMS processes (Zwikael, 2008a, b).

The effect of TMS in project success has been found to correlate highly with the overall project success. As a matter of fact, TMS has been found to be one of the most significant factors contributing to project success (Fortune and White, 2006; Young and Jordan, 2008). PM literature lists various ways for top management to support projects. For example, Zwikael has identified a list of 17 relevant support practices of how top management could support IT projects (Zwikael, 2008a). Zwikael's list of 17 support practices is presented below:

(1) project-based organisation;

(2) existence of project procedures;

(3) appropriate project manager assignment;

(4) refreshing project procedures;

(5) involvement of the project manager during initiation stage;

(6) communication between the project manager and the organisation;

(7) existence of project success measurement;

(8) supportive project organisational structure;

(9) existence of interactive inter-departmental project groups;

(10) organisational projects resource planning;

(11) organisational projects risk management;

(12) organisational projects quality management;

(13) on-going PM training programmes;

(14) PM office involvement;

(15) extent of use of standard PM software;

(16) use of organisational projects data warehouse; and

(17) use of new project tools and techniques.

This paper utilises the fact that both SCE and PM affect project success. TMS has been successfully studied in scope of PM, but not in scope of SCE. Considering the universal nature of TMS, it is likely that the same or similar top management CSFs, which contribute positively to project success during a project, contribute positively on SCE success as well.

Based on the previous, a list of 16 support practices is developed and used for studying top management involvement in SCE. The selected support practices are adopted from Zwikael's list of support practices in IT PM, and from other best practices presented in SCE literature. This list of 16 support practices is used in a questionnaire to study the current practice and perceived importance of top management involvement in SCE. The purpose of the questionnaire is to understand TMS in SCE better.

2. Research configuration

A list of predefined support practices was developed to capture the current practice and opinions about top management involvement in SCE. An internet-based questionnaire was used for data collection. A fourfold Likert scale was used for evaluating both dimensions, the maximum values indicating that SCE is always used (1 = "never used" to 4 = "always used") or considered to be of high importance (1 = "no importance"… to 4 = "high importance"). Statistical methods were used for analysing the relationships between the current practice, attitudes and project success.

2.1 Methodology

In order to capture the current practice of TMS in SCE, and the opinions of software professionals regarding the importance of TMS practices, a list of 16 TMS practices in SCE has been developed. The list is adapted from Zwikael's PM related top management CSFs (Zwikael, 2008a) and other relevant TMS practices presented by McConnell (2006). The practices presented on the developed list can reasonably be assumed to be relevant also in the SCE scope. The practices represent conceptually two different kinds of practices: tangible doing (practices 1-10) and basic awareness of SCE (practices 11-16). The list of 16 support practices is presented below:

(1) top management ensures existence of estimation procedures;

(2) top management ensures that the estimator has adequate skills;

(3) top management ensures improving estimation procedures;

(4) top management ensures the involvement of the project manager during the estimation stage;
(5) top management ensures good communication between the estimator and the organisation;
(6) top management ensures that there are criteria for evaluating the meaningfulness of the estimate;
(7) top management ensures on-going estimation skills training programmes;
(8) top management requires re-estimating during the project to get more accurate estimates;
(9) top management ensures that the estimate relies on documented facts rather than guessing and intuition;
(10) the IT executive studies and approves the estimate;
(11) top management recognises that estimates are critical to this organisation's success;
(12) top management is knowledgeable of estimation procedures;
(13) top management understands the consequences of an erroneous estimate to the project success;
(14) top management can distinguish between estimates, targets and commitments;
(15) top management recognises that the estimates are inaccurate in the beginning of the project; and
(16) top management takes the output of an estimate as given without debate.

2.2 Data collection and questionnaire design

A questionnaire of 17 questions was designed for data collection. The questionnaire was to:

(1) collect information about the respondent and the project in question;
(2) identify the extent of use of TMS practices in SCE; and
(3) identify software professionals' opinions on the importance of different TMS practices in SCE.

A pilot survey was conducted prior to the main survey. In the pilot survey, an invitation to answer the questionnaire was sent to 30 software professionals, of which 16 responded. Based on the feedback from the pilot survey, 20 changes were made to the questionnaire. The main survey was implemented as an anonymous internet-based questionnaire, and a link to the questionnaire was sent to 1,109 software professionals. The recipients of the invitation were selected based on the mailing lists provided by Project Management Institute Finland chapter and Digile Oy (Digile Oy is a non-profit Finnish organisation, which coordinates national and international research programmes). The recipients were instructed to answer the questionnaire only if they had been actively involved in the estimation of a software project within the past 24 months, and to base their responses on their most recent estimated project, including abandoned projects. The reason for anonymity was to maximise the number of responses.

Of the all recipients, 114 completed the whole questionnaire (10.28 per cent response rate). In total, 291 recipients started answering the questionnaire, but only 184 of them had participated SCE during the past 24 months; 70 of those 184 respondents failed to answer all questions, and their incomplete answers were not included as part of the survey.

The first two of the 17 questions were screening questions determining whether the respondent is eligible for taking part in the questionnaire. The next four questions determined the role of the respondent in SCE. Questions 7-10 measured the experienced importance and frequency of use of TMS practices in SCE. The respondents were also asked to add their own suggestions for practices. The rest of the questions determined the characteristics of the estimated project. The list of the questions is presented below:

(1) Have you already answered this survey? If you answer Yes, you will be forwarded to the end of the questionnaire.

(2) Have you participated SCE during the past 24 months? If you answer No, you will be forwarded to the end of the questionnaire.

(3) Which of the following best describes your role in SCE?

(4) Which of the following best describes your role in making the go/No Go decision about the estimated project?

(5) Which of the following best describes your role in the estimated project?

(6) Which of the following best describes your job role?

(7) Indicate the Frequency of the ways of top management involvement in SCE in your organisation.

(8) Specify other ways of top management involvement in SCE, which were not presented in the previous question, and indicate the Frequency.

(9) According to your subjective opinion, indicate the Importance of the ways of top management involvement in SCE from the estimation success point of view.

(10) Specify other ways of top management involvement in SCE, which were not presented in the previous question, and indicate the Importance from the estimation success point of view according to your subjective opinion.

(11) In your opinion, indicate when the estimate meaningfulness was evaluated.

(12) Indicate the industry sector of the latest estimated project. Indicate either your own organisation's sector, or client organisation's sector, if the project was developed for a client.

(13) How many people are currently employed in your organisation?

(14) How many people were involved in the latest estimated project?

(15) What was the length of the last estimated project in months?

(16) What is the completion status of the last estimated software project?

(17) How successful was the last estimated project?

2.3 Respondent and project demographics

This section introduces respondent related demographics, including the projects at hand. All questions were answered by 114 respondents ($n = 114$).

2.3.1 Industry sector. The respondents were asked to indicate the industry sector of the estimated project. ICT accounted for 33 per cent (38) of responses. The complete breakdown is shown in Figure 1.

2.3.2 Number of employees. The number of employees was classified into five categories, as shown in Figure 2. Most of the respondents, 33 per cent (38), worked in organisations with 50 or less employees. The breakdown is presented in Figure 2.

2.3.3 Numbers involved. The number of people involved in the estimated project was classified into five ranges. When asked, most of the respondents, 38 per cent (43), indicated that the number of persons involved was between six and ten. The breakdown is shown in Figure 3.

2.3.4 Project duration. Most of the projects lasted 12 months or less. In total, 34 per cent (39) of the projects lasted between one and six months and 34 per cent (39) of the projects lasted between seven and 12 months. The categorised breakdown can be seen in Figure 4.

2.3.5 Role of respondent. The invitation to answer was sent to software professionals, of which 22 per cent (25) identified themselves as project managers, 16 per cent (20) as software developers, team leaders or architects, 30 per cent (34) as senior managers, 7 per cent (8) as directors and 16 per cent (18) as managers. The full breakdown is presented in Figure 5(a).

2.3.6 Role in project. Respondents were also asked to indicate their role in the estimated project. Totally, 24 per cent (28) answered they were project managers, 11 per cent (13) software developers and 4 per cent (4) non-technical project team members. In total, 25 per cent (29) were steering group members and 27 per cent (31) were project owners; 8 per cent (9) answered that they had some other role, like coach,

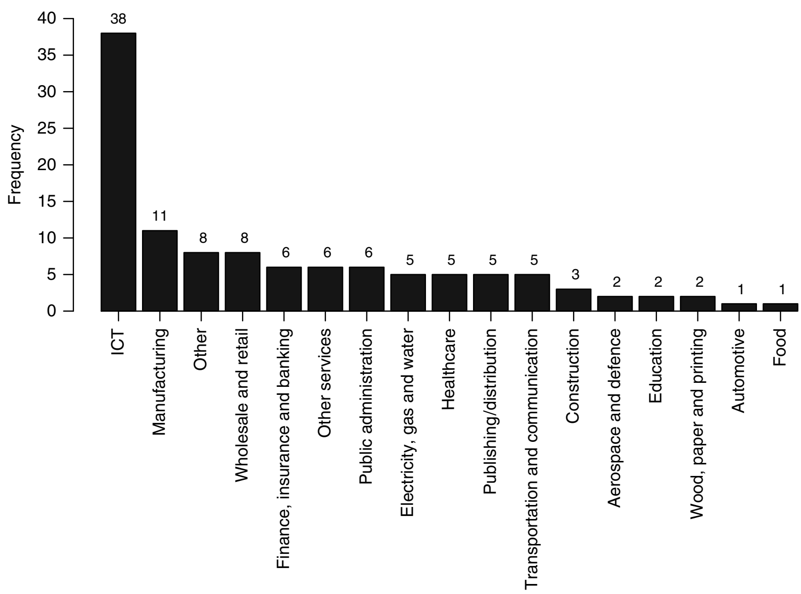

Figure 1.
Industry sector breakdown

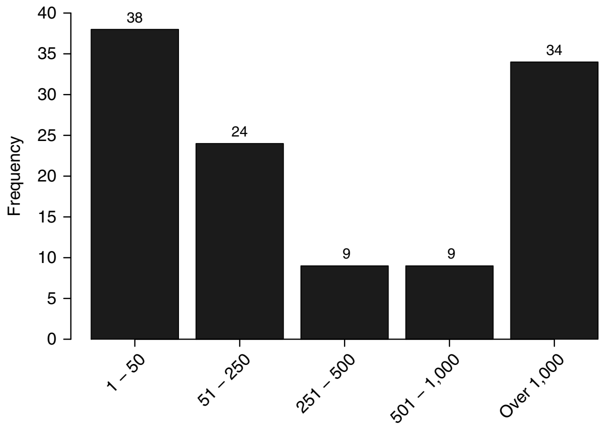

Figure 2.
Number of employees

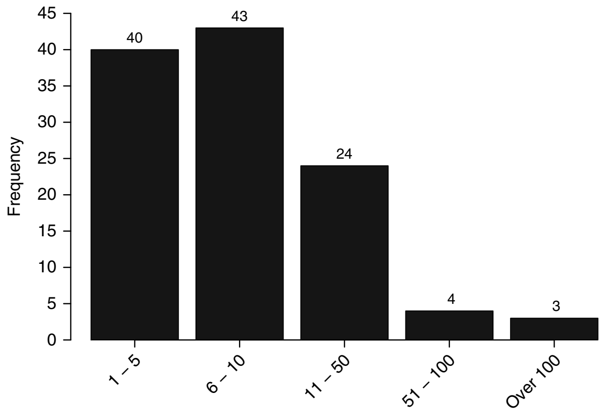

Figure 3.
Number of people involved in the project

quality manager or security officer. None of the respondents answered that they were not part of the project itself. The roles are presented in Figure 5(b).

2.3.7 Role in SCE. When requested to indicate their role in SCE, 33 per cent (38) of the respondents stated that they had the overall responsibility of preparing the estimate, 19 per cent (22) were involved in estimating certain functionalities, 40 per cent (46) reviewed the estimate and 7 per cent (8) had some other role in the estimation, like process improvement coach or they needed the estimate for the business case preparation. The roles are also shown in Table I.

2.3.8 Role in decision making. In all, 26 per cent (30) of the respondents indicated that they made the final decision about approving or rejecting the project, while 25 per cent (28) indicated they had some decision-making power, but someone else made the final decision; 34 per cent (39) recommended the project to the decision makers and 11 per cent (13) had their opinion considered, but they did not have formal decision

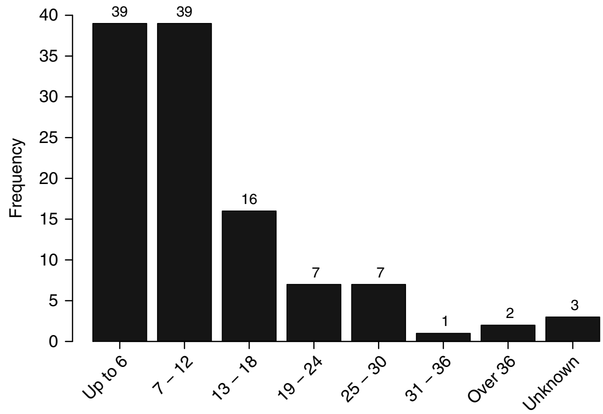

Figure 4.
Project duration

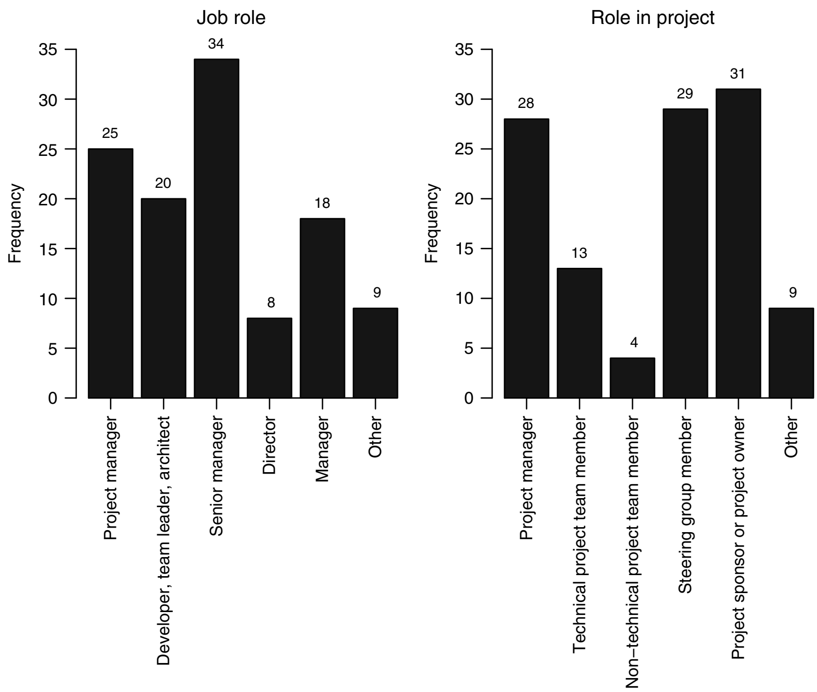

Figure 5.
(a) Job role; (b) role in project

making power. Only 4 per cent (4) indicated that they did not participate in decision making. The roles can be seen in Table II.

2.3.9 Completion status. Exactly half, or 50 per cent (57), of the respondents answered that the estimated project was still in progress. In total, 36 per cent (41) of the projects were delivered, and the rest of the projects did not get approved, were terminated or had not been started yet. The full breakdown is available in Figure 6(a).

2.3.10 Project success. The respondents were offered four alternatives to estimate project success: "complete success", "quite successful", "poor success" and "terminated/complete failure". In total, 73 per cent (83) of the respondents were able to indicate the success of the project, while 27 per cent (31) indicated that the information is not available; 14 per cent (12) of those who indicated the success of the project reported that the project was "complete success", while 73 per cent (61) reported "quite successful", 10 per cent (8) "poor success" and 2 per cent (2) "terminated/complete failure". This result corresponds somewhat with the other reported cost and budget overrun rates (Moløkken and Jørgensen, 2003). However, it must be noted that only 36 per cent (41) of the estimated projects had yet been concluded, so many of the respondents may have indicated the success of the project prematurely. The complete breakdown is shown in Figure 6(b).

2.3.11 Evaluation of estimate meaningfulness. The survey defined a meaningful estimate as follows, adapted from McConnell, (2006):

> A meaningful estimate is an estimate that provides a clear enough view of the project reality to allow the project leadership to make good decisions about how to control the project to hit its targets.

The respondents were asked to indicate in which phases of the project the meaningfulness of the estimate was evaluated in their opinion. The question was asked for three different phases: before the project, during the project and after the project. The respondents were offered five alternatives: disagree, tend to disagree, tend to agree, agree and too early to answer. For all phases the alternative agree was the most popular answer. The percentages for "agree" before, during and after the project were 57 per cent (62), 48 per cent (50) and 38 per cent (36) of those who already could answer the question, respectively. The full breakdown is presented in Table III.

Table I. Role in software cost estimation

Role in software cost estimation	Frequency	%
I had the overall responsibility for preparing the estimate	38	33
I estimated the effort for certain functionalities	22	19
I reviewed the estimate	46	40
Other	8	7

Table II. Role in decision making

Role in decision making	Frequency	%
I made the final decision	30	26
I had decision making power, but someone else made the final decision	28	25
I made the recommendation for the decision makers, but I did not have formal decision making power	39	34
My opinion was considered, but I did not have formal decision making power	13	11
I did not participate in decision making	4	4

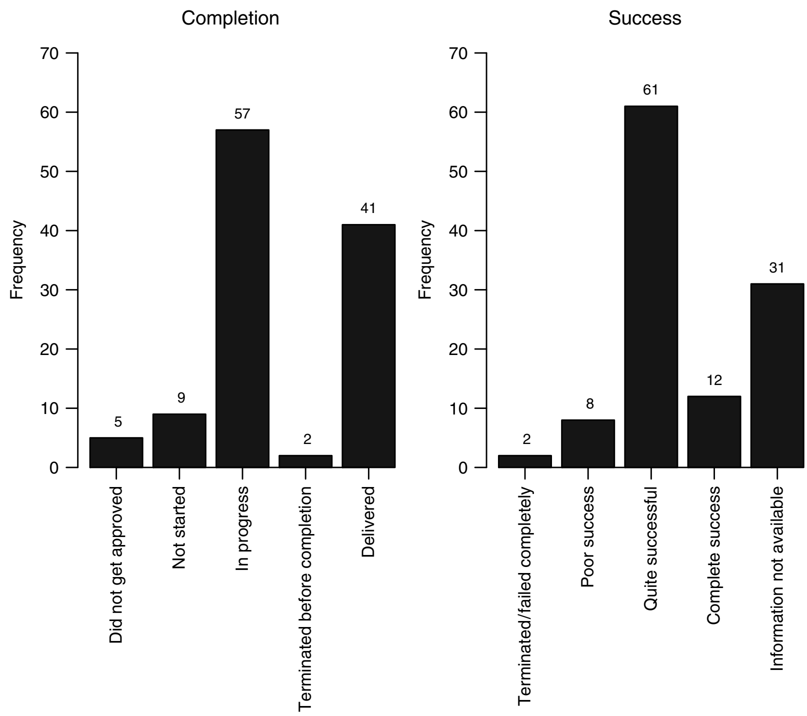

Figure 6.
(a) Project completion status; (b) project success

Table III.
Evaluation frequency of cost estimate meaningfulness at different project phases

Meaningfulness evaluation point	Disagree	Tend to disagree	Tend to agree	Agree	Too early to answer	Total
Before the project	4	22	21	62	5	114
During the project	5	14	36	50	9	114
After the project	14	22	24	36	18	114

3. Results and discussion

This section discusses the main survey findings about TMS in SCE. First, the validity of the list of 16 support practices for measuring the current practice and perceived importance is discussed. After that the correlation between these two dimensions of TMS is analysed. Third, the results from the causal analysis between the variables are presented, and finally, the extent of use and attitudes are compared, and the scores for the individual support practices are discussed.

3.1 Validity of support practices

The primary focus of this paper is to establish what is the level of TMS for SCE, and what is the level of experienced importance of TMS for SCE. A list of 16 TMS practices for SCE was developed and used for measuring the current practice and attitudes. The list's validity was evaluated according to three conventional techniques: Cronbach's standardised α, Spearman's rank correlation coefficients and an eigenvalue decomposition.

Cronbach's standardised α was computed separately for the practices and attitudes. The obtained values are 0.92 and 0.88, respectively. According to the literature, values over 0.9 indicate "excellent" internal consistency and values over 0.8 "good" internal consistency of the test items (George and Mallery, 2003), meaning that the items on the list are very likely to measure the same single theoretical construct. These values are also higher than required in the statistical literature (George and Mallery, 2003).

The internal consistency was studied further by generating Spearman's rho matrices for both the use of practices and attitudes (Figures 7 and 8).

Also these matrices show a clear correlation between the scores for the list items. However, the matrices show a negative correlation for one practice: "Top management takes the output of an estimate as given without debate". A peer discussion of the possible reasons for the deviating behaviour was conducted. The conclusion was that the proposition is not clear, and it can be interpreted either so that a higher score would imply a positive aspect of SCE, or so that a higher score would imply a negative aspect of SCE. For this reason, the proposition cannot be considered as valid from research point of view.

Finally, a scree plot was used for assessing the number of latent dimensions in the data (Tabachnick and Fidell, 2001). When the above two correlation matrices were used to extract the corresponding eigenvalues, it is clear that a single latent construct is present (Figure 9), as represented by the dominant eigenvalue.

Considering the results of the previous tests, in statistical terms the presented 16 support practices seem to measure the same phenomenon, TMS for SCE, both in terms of extent of use and experienced importance. In other words, the list is valid for measuring TMS for SCE.

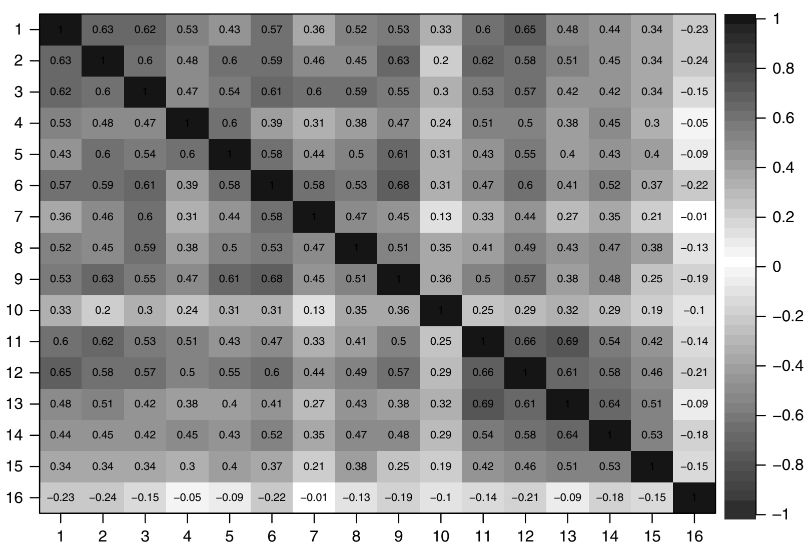

Figure 7.
Correlation between use of support practices (Spearman)

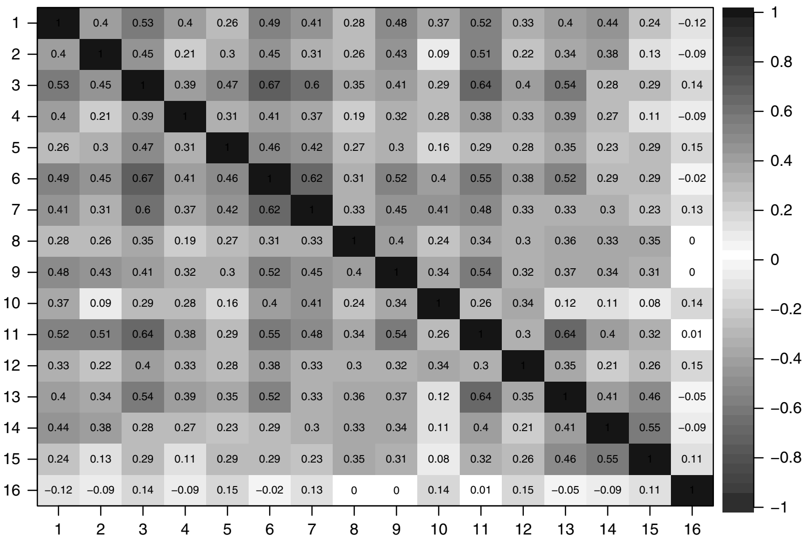

Figure 8.
Correlation between attitudes towards support practices (Spearman)

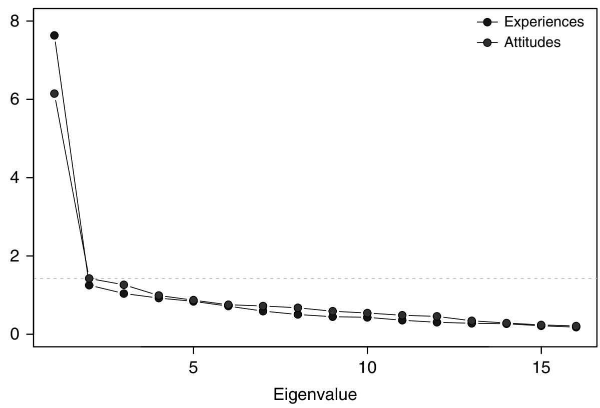

Figure 9.
Scree plot of test items

3.2 Correlation between current practice and attitudes

The correlation between the two latent constructs, extent of use and experienced importance, is moderate (Pearson's $r = 0.48$). The same also holds for the individual elements. When tested with nonparametric Mann-Whitney *U*-tests (Table IV), the null hypothesis of equal distributions holds only for the following three practices:

(1) top management recognises that estimates are critical to this organisation's success;

(2) top management ensures involvement of the project manager during the estimation stage; and

(3) top management is knowledgeable of estimation procedures.

Table IV. Use of practices, attitudes and correlation

Top management support practice	Frequency (mean)	Importance (mean)	Difference	U ($\rho < 0.05$)
Top management understands the consequences of an erroneous estimate to the project success	2.97	3.47H	0.50	4,361
Top management recognises that estimates are critical to this organisation's success	2.96	3.32H	0.37	5,012C
Top management ensures involvement of the project manager during the estimation stage	2.94	3.25	0.31	5,242C
Top management can distinguish between estimates, targets and commitments	2.84	3.41H	0.57	4,170
Top management recognises that the estimates are inaccurate in the beginning of the project	2.68	3.30	0.62	3,967
Top management ensures existence of estimation procedures	2.67	3.18	0.52	4,500
Top management ensures that the estimator has adequate skills	2.66	3.32H	0.66	4,153
Top management is knowledgeable of estimation procedures	2.57	2.86L	0.29	5,399C
Top management ensures good communication between the estimator and the organisation	2.56	3.20	0.64	4,005
The IT executive studies and approves the estimate	2.51	2.71L	0.20	–
Top management ensures that the estimate relies on documented facts rather than guessing and intuition	2.44	3.05	0.61	4,269
Top management requires re-estimating during the project to get more accurate estimates	2.38	3.10	0.72	3,758
Top management ensures that there are criteria for evaluating the meaningfulness of the estimate	2.36	3.04	0.68	3,930
Top management ensures improving estimation procedures	2.26	2.98	0.72	3,821
Top management takes the output of an estimate as given without debate	2.15	2.04	−0.11	
Top management ensures on going estimation skills training programmes	1.74	2.47L	0.74	3,664
Mean	2.54	3.04	0.50	
SD	0.33	0.37		

Notes: H, one of the four practices with the highest perceived importance; L, one of the three practices with the lowest perceived importance; C, frequence and importance correlate

It is notable that all of these three practices were ranked high, both in terms of use of practices and attitudes. In other words, not only the attitudes towards them are positive, but they have also received top management attention in practice.

When considering the first item, it seems understandable that the first step in supporting SCE is to be aware and recognise the importance of SCE for the organisation. In the scope of PM, awareness, recognition, commitment and other related concepts are listed among the most important success factors (White and Fortune, 2002; Young and Jordan, 2008). It is also intuitively understandable that to work effectively with something, awareness and recognition of the concept are important. Reports of failures and high overrun rates (Ewusi-Mensah, 2003; Charette, 2005; Jiang, 2005; Standish Group, 1994) have also made top managers painfully aware of the organisational consequences of failed estimates.

Project manager's role in a project is central, with an overall responsibility of the project success (PMI, 2013). Considering that, it is not surprising that involving the project manager is both seen as important, as well as practiced often. Also the literature mentions early involvement of the project manager as a critical TMS process (Zwikael, 2008a).

While the first two items may seem obvious, the third item does not seem to correlate well with the other studies. For example Hihn and Habib-agahi (1991) report that even as many as 83 per cent of estimators seem not to use any estimation methodologies, or they rely on ad-hoc or memory-based methodologies. Also Lederer and Prasad (2000) report that there are no practices discouraging the use of guessing as an estimation methodology. One possible reason for the difference is that an estimation methodology and estimation procedure are different concepts. The estimation methodology generally refers to a task for producing an effort estimate for a certain piece of software, while the estimation procedure refers to a broader concept, considering things like qualification of the estimator, presentation of the estimate and re-estimation. However, this conceptual difference is not likely to explain the difference, since an estimation procedure should provide protection against off-the-cuff estimation and guessing (McConnell, 2006). Another alternative is that the estimation procedure is interpreted as a synonym for some other concept, like business case preparation, which could explain the high rate of application, and the correlation with the attitudes.

The rest 13 pairs, which did not correlate with each other, are all operative by nature, related to PM or cost estimation as a technical performance. Generally these kind of tasks are of little interest to senior managers (Thomas *et al.*, 2002; Crawford, 2005). The first correlating item relates to organisational success, and the second is strategic for the project, so it seems natural that these two items draw senior management's attention. This phenomenon may explain the decreased correlation between the extent of use and attitudes for the rest of the practices. From the research point of view, studying support practices with strategic elements further could be justifiable.

3.3 Causal analysis

In the first test, project and respondent related variables' influence on the use of practices and attitudes was tested with two separate ordinary least squares (OLS) regression models. The basic model setup was similar in both models: the sum variables of extent of use and attitudes were used as dependent variables, while respondent and project related background variables were used as independent variables. The tested independent variables were the following: ICT sector, firm size, project size, project duration, project status and estimate meaningfulness evaluation, before the project, during the project and after the project.

According to the results, none of the tested independent variables were statistically significant under the conventional 5 per cent significance level ($p < 0.05$). The adjusted R^2-values were both below 0.13, meaning that the models could explain a little over 10 per cent of the variance seen in the sum variables. Considering this, other factors seem to determine most of the extent of use or attitudes towards SCE.

As a second regression modelling approach, a logistic regression model was fitted with a recoded project success variable as the dependent phenomenon. That is, in Figure 6 the two categories, "quite successful" and "complete success" were merged into one. In addition to the independent variables in the OLS models, the sum variables, extent of use and attitudes, were added as independent variables. The computed pseudo-R^2 measure (McFadden, 1974) of the model was 0.10. None of the included independent variables were found to be statistically significant, indicating no influence on project success. This result deviates at least from Jones (1998), which reports that project size affects the probability of project success. However, the sample size in this study was relatively small, and over 80 per cent of the studied projects were shorter than 18 calendar months. This may explain the deviating result.

Finally, a nonparametric Kruskal-Wallis rank sum test was used to investigate the effects of the individual 16 items. Only the following two were found to be significant ($p < 0.05$) with respect to project success:

(1) "Top management ensures that the estimate relies on documented facts rather than guessing and intuition" for the extent of use; and

(2) "Top management recognizes that estimates are critical to this organization's success" for the experienced importance.

Regardless of the statistical significance, it could be premature to draw conclusions of these practices' impact on the project success. Further studies are needed to confirm whether these findings are really significant or just outliers. However, guessing and intuition have proven to be connected to schedule overruns (Lederer and Prasad, 1992), so relying on documented facts in SCE could, in the light of the previous research, be expected to have a positive impact on project success. Also, recognition of the importance of estimates should intuitively help in supporting SCE, and contribute positively to project success, as well.

3.4 General observations

As typical for survey setups like the one used in this study, the attitudes are more positive than the actual extent of use (Table IV). In this case, the average difference is 0.50 in favour of attitudes. Table IV also shows a moderate correlation between the extent of use and attitudes, correlating with the results of the statistical analysis. Four attitudes with the highest scores are marked with "H", and three with the lowest scores with "L". As can be seen, the attitudes with the highest scores are among the most practiced ones, and the attitudes with the lowest scores among the least practiced.

When considering the lower ends of the lists, "skills training programmes", "improving estimation procedures" and "re-estimation" are related to learning and improvement. Also Zwikael (2008a) reports similarly low scores for learning related practices in software development organisations. A low score for "having a criteria for evaluating the meaningfulness of the estimate" may suggest that the software companies are not that interested in the quality of the estimate. In terms of attitudes, "the IT executive studies and approves the estimate" received a low score. This may indicate low confidence towards the IT management. The respondents also reported that in the majority of the estimated projects, "ensuring that the estimate is based on documented facts rather than guessing" was practiced "never" or "occasionally". This corresponds well with the findings in Lederer and Prasad (2000).

"Top management understands the consequences of an erroneous estimate to the project success" got the highest score in both dimensions of TMS. As discussed earlier, a high score seems understandable, because project failures and overruns are reported widely. The two most used practices confirm that the software professionals are very well aware of the importance of an estimate, and possible consequences of an erroneous estimate. Furthermore, "top management can distinguish between estimates, targets and commitments" is related to realism and clear goals, which are considered to be of very high importance also in White and Fortune (2002).

The respondents also suggested 51 different other means for TMS in SCE, of which eight could be categorised under "Top management ensures that all stakeholders have been consulted". This is close to "Top management ensures good communication between the estimator and the organisation", but it addresses more directly the need to present the estimate to all stakeholders when the estimate is ready, instead of ensuring good communication during the estimation work. This suggestion seems reasonable, and should be studied further in the future. Otherwise the comments were related more to PM and business case preparation than to SCE.

3.5 Implications for practice and research

The analysis conducted in this study did not reveal significant relationships between TMS in SCE and project success, and therefore this paper cannot give any strong recommendations. However, the study clearly shows that top management invests a significant amount of its valuable time in supporting SCE. The list of practices and results presented in this paper may still help senior managers to better prioritise their doings related to SCE. While not statistically proven, it can be reasonably assumed that the practices with the high extent of use and perceived importance are more important from the project success point of view than the practices with low scores. The results may also promote the importance of SCE among top managers and help them to start actions towards more meaningful and accurate estimates.

From the research and theory point of view, this paper has concentrated specifically on the construct "TMS in SCE". Prior work has extensively studied different estimation methods and TMS's role in PM. This study continues the work by addressing the tight coupling of an estimate, project and TMS. The results show that top management significantly gives their attention to SCE through different support practices, which all are related to the same construct, TMS in SCE.

3.6 Further research

The analysis conducted in this paper was not able to identify a clear list of support practices, the use of which would increase the probability of project success. The identification of this list is probably the most interesting research question, because top management's time is a scarce resource, and it should be invested effectively. A new study with a larger sample of concluded projects should be conducted to get more reliable results. One natural next step would also be to repeat this survey in other countries than Finland. The presented list of support practices is also likely to be incomplete, and it contains mostly operative practices perhaps of a low interest among the senior managers. Further research could make a more indepth case and interview based study to validate the results of this study, and to identify other possible support practices, preferably more strategic of nature. A more qualitative study is also likely to reveal other interesting aspects of top management involvement in SCE, which were not revealed by this survey. Specifically top management's viewpoint to software estimation and supporting it should be studied. This kind of a study would be likely to give valuable information about different organisational behaviours or other issues, which are supporting or inhibiting top management to demonstrate its support to SCE. Project manager and software professional viewpoints to TMS in SCE as separate items may also help to understand the matter as a whole better.

3.7 Limitations

This paper used a questionnaire to capture the extent of use and perceived importance of different TMS practices for SCE. However, this study is the first of its kind, and as all surveys, it is prone to bias. For example, the items on the list of preselected support practices may have been interpreted differently by the respondents, and a different kind of questionnaire setup could have changed the results. Also, all the respondents were working in Finland, and support practices and their demonstration may vary from country to country. Furthermore, the sample size of concluded projects was small, decreasing the reliability of statistical tests between the practices and project success. Further studies are needed to validate the results presented in this paper.

4. Conclusions

TMS has been studied extensively in the scope of PM, and found to have a positive impact on project success. This paper extends the scope of TMS research to also cover SCE. This paper clearly shows that top management is investing a lot of its attention in SCE. This paper also shows that the attitudes towards top management participation are very positive, top management participation is seen to be of a high importance. Considering this, understanding better the impact of different practices could help senior managers to invest their time more effectively and increase the probability of project success.

This paper has built a list of 16 TMS practices, adopted from Zwikael (2008a) and McConnell (2006). The list has been used in a questionnaire to measure the extent of use and experienced importance of an important PM support factor, TMS (Fortune and White, 2006; Young and Jordan, 2008; Schmidt *et al.*, 2001), in SCE. The paper reports the extent of use and the experienced importance of the support practices, and focuses on analysing the relationship between these two and project success.

The list of 16 support practices presented in this paper has been found to measure the construct "Top management participation in SCE" well. Table IV shows the extent of use and attitudes for each individual practice. As can be seen, top management demonstrates their support through these processes often, the average frequency being 2.54 on a scale from 1 to 4. The attitudes were also positive, the indicated average importance was 3.04. Table IV also shows that only for three of the 16 practices the extent of use and experienced importance correlate with each other. For some reason, most of the presented support practices do not get the top management's attention to the same extent as the practices are experienced important. This study also tested the impact of the extent of use and the experienced importance of the practices on project success, trying to identify the most important support factors. The following two were found to have a positive influence on project success:

(1) "Top management ensures that the estimate relies on documented facts rather than guessing and intuition" for the extent of use; and

(2) "Top management recognizes that estimates are critical to this organization's success" for the experienced importance.

However, these findings may also be outliers, and further research should be conducted to gain more reliable understanding of the impact of different support practices on project success.

References

Boonstra, A. (2013), "How do top managers support strategic information system projects and why do they sometimes withhold this support?", *International Journal of Project Management*, Vol. 31 No. 2, pp. 498-512.

Charette, R.N. (2005), "Why software fails", *IEEE Spectrum*, Vol. 42 No. 9, p. 36.

Cooke-Davies, T. (2002), "The 'real' success factors on projects", *International Journal of Project Management*, Vol. 20 No. 3, pp. 185-190.

Crawford, L. (2005), "Senior management perceptions of project management competence", *International Journal of Project Management*, Vol. 23 No. 1, pp. 7-16.

Doherty, N.F. and King, M. (2001), "An investigation of the factors affecting the successful treatment of organisational issues in systems development projects", *European Journal of Information Systems*, Vol. 10 No. 3, pp. 147-160.

Ewusi-Mensah, K. (2003), *Software Development Failures*, MIT Press, Cambridge, MA.

Fortune, J. and White, D. (2006), "Framing of project critical success factors by a systems model", *International Journal of Project Management*, Vol. 24 No. 1, pp. 53-65.

George, D. and Mallery, P. (2003), *SPSS for Windows Steps by Steps: A Simple Guide and Reference 11.0 Update*, Allyn & Bacon, Boston, MA.

Hihn, J. and Habib-agahi, H. (1991), "Cost estimation of software intensive projects: a survey of current practices", *Proceedings of the 13th International Conference on Software Engineering, IEEE*, *Computer Society Press*, *Austin, TX, 13-17 May*.

Jones, C. (1998), *Estimating Software Costs*, McGraw-Hill Inc., New York, NY.

Jørgensen, M. and Shepperd, M. (2007), "A systematic review of software development cost estimation studies", *Software Engineering, IEEE Transactions on*, Vol. 33 No. 1, pp. 33-53.

Jørgensen, M. and Sjøberg, D.I.K. (2001), "Impact of effort estimates on software project work", *Information and Software Technology*, Vol. 43 No. 15, pp. 939-948.

Lederer, A. and Prasad, J. (1992), "Nine management guidelines for better cost estimating", *Communication of the ACM*, Vol. 33 No. 2, pp. 51-59.

Lederer, A.L., and Prasad, J. (1995), "Causes of inaccurate software development cost estimates", *Journal of Systems and Software*, Vol. 31 No. 2, pp. 125-134.

Lederer, A.L. and Prasad, J. (2000), "Software management and cost estimating error", *Journal of Systems and Software*, Vol. 50 No. 1, pp. 33-42.

McConnell, S. (2006), *Software Estimation-Demystifying the Black Art*, Microsoft Press, Washington, DC.

McFadden, D. (1974), "Conditional logit analysis of qualitative choice behavior", in Zarembka, P. (Ed.), *Frontiers in Econometrics*, Academic Press, New York, NY, pp. 105-142.

Moløkken, K., and Jørgensen, M. (2003*)*, "A review of software surveys on software effort estimation", *Empirical Software Engineering, 2003, ISESE 2003, Proceedings, 2003, International Symposium on, IEEE*, pp. 223-230.

Pitterman, B. (2000), "Telcordia technologies: the journey to high maturity", *IEEE Software*, Vol. 17 No. 4, pp. 89-96.

PMI (2013), *A Guide to the Project Management Body of Knowledge (PMBOK® Guide),* Project Management Institute Inc., Newtown Square.

Putnam, L. and Myers, W. (2003), *Five Core Metrics*, Dorset House, New York, NY.

Schmidt, R., Lyytinen, K., Keil, M. and Cule, P. (2001), "Identifying software project risks: an international delphi study", *Journal of Management Information Systems*, Vol. 17 No. 4, pp. 5-36.

Standish Group (1994), *The Chaos Report*, The Standish Group, Boston, MA.

Tabachnick, B. and Fidell, L. (2001), *Using Multivariate Statistics*, Allyn and Bacon, Boston, MA.

Thomas, J., Delisle, C., Jugdev, K. and Buckle, P. (2002), "Selling project management to senior executives: the case for avoiding crisis sales", *Project Management Journal*, Vol. 33 No. 2, pp. 19-28.

Van Genuchten, M. (1991), "Why is software late? An empirical study of reasons for delay in software development", *Software Engineering, IEEE Transactions on*, Vol. 17 No. 6, pp. 582-590.

White, D. and Fortune, J. (2002), "Current practice in project management – an empirical study", *International Journal of Project Management 2002*, Vol. 20 No. 1, pp. 1-11.

Young, R. and Jordan, E. (2008), "Top management support: mantra or necessity?", *International Journal of Project Management*, Vol. 26 No. 7, pp. 713-725.

Zwikael, O. (2008a), "Top management involvement in project management: a cross country study of the software industry", *International Journal of Managing Projects in Business*, Vol. 1 No. 4, pp. 498-511.

Zwikael, O. (2008b), "Top management involvement in project management: exclusive support practices for different project scenarios", *International Journal of Managing Projects in Business*, Vol. 1 No. 3, pp. 387-403.

Further reading

Harter, D.E., Krishnan, M.S. and Slaughter, S.A. (2000), "Effects of process maturity on quality, cycle time, and effort in software product development", *Management Science*, Vol. 46 No. 4, pp. 451-466.

About the authors

Jurka Rahikkala is the Founder and the COO of a global open source company, Vaadin Ltd. He has a long experience in leading international software projects and working as a Senior Manager in multinational companies. His professional interests and areas of expertise include software project business cases, cost estimation and project leadership. He received his Masters in Computer Science from the University of Turku in 1998, and now he is preparing his PhD thesis. Jurka Rahikkala is the corresponding author and can be contacted at: jurka.rahikkala@gmail.com

Ville Leppänen, PhD, works currently as a Software Engineering Professor in the University of Turku, Finland. He has over 100 scientific publications. His research interests are related broadly to software engineering ranging from software engineering methodologies, practices and tools to security and quality issues and to programming language, parallelism and algorithmic design topics.

Jukka Ruohonen is a Doctoral Student at the University of Turku. His research interests include software engineering, applied statistics and applied use of machine learning techniques, among other things.

Johannes Holvitie, MSc (Tech.), is a Doctoral Student at the University of Turku. His research focus and doctoral dissertation topic is Technical debt as a factor in efficient and sustainable software development.

PII

ACCOUNTING TESTING IN SOFTWARE COST ESTIMATION: A CASE STUDY OF THE CURRENT PRACTICE AND IMPACTS

By Jurka Rahikkala, Sami Hyrynsalmi and Ville Leppänen

Originally published in *Proceedings of the 14th Symposium on Programming Languages and Software Tools (SPLST'15)*

Accounting Testing in Software Cost Estimation: A Case Study of the Current Practice and Impacts

Jurka Rahikkala[1,2], Sami Hyrynsalmi[2], Ville Leppänen[2]

[1] Vaadin Ltd, Turku, Finland
jurka.rahikkala@vaadin.com
[2] Department of Information Technology, University of Turku, Finland
{sthyry, ville.leppanen}@utu.fi

Abstract. Testing has become an integral part of most software projects. It accounts for even as high a share as 40% of the overall work effort. At the same time software projects are systematically exceeding their effort and schedule forecasts. Regardless of the overruns and big share of testing, there is very little advice for estimating testing activities. This case study research assesses the current practice of estimating testing activities, and the impact of these practices on estimation and project success. Based on the interviews with 11 stakeholders involved in two case projects and examination of project documentation, this study shows that companies easily deviate from their standard procedures, when estimating testing. This may even lead to severe estimation errors. The deviations can be explained by negative attitudes towards testing. Furthermore, this study shows that the extant literature has sparsely addressed estimation of software testing.

Keywords: Software Cost Estimation, Software testing, Project Management

1 Introduction

Software cost estimation (SCE), or effort estimation, is an art which is not well handled by the software industry (see e.g. Rahikkala et al., 2014). The famous *Chaos Reports* by Standish Group International have for decades claimed that nearly two thirds of software products either failed to meet their budget and timetable goals or they were never delivered (The CHAOS Manifesto, 2013). While these numbers have been heavily criticized (see Laurenz Eveleens and Verhoef, 2010), high failure percentages have been shown also by other research and consultancy institutes (c.f. Moløkken and Jørgensen, 2003; Mieritz, 2012). Nevertheless, these numbers clearly show the dilemma of software cost and effort estimations: they often fail.

In this study, our topic is an increasingly important aspect of SCE: estimating the effort of software testing activities. The software testing activities have been found to take from 20%–25% (Haberl et al., 2012; Lee et al., 2013; Buenen and Teje, 2014) to even 40% or more (Ng et al., 2004) of the total cost of the project. However, there is still a lack of knowledge of SCE methods as well as proper tools and practices for software testing estimation. For example, from the studies included into Jørgensen and Shepperd's (2007) systematic literature review on software cost estimation, only a handful seems to discuss estimation of software testing activities.

Thus, there is a gap in extant literature on the effect of software testing effort estimation. The objective of this paper is to 1) gain in-depth knowledge of the current practice for accounting testing in SCE, and 2) understand the impact of the used practices on SCE and project success.

To answer the research questions, a qualitative research approach is used. We study software cost estimation practices in two projects in two case companies. Based on the study of 11 interviews and 17 project related documents, this paper contributes to the scientific literature by reporting on the current practice of estimating testing effort, and the impact of use of these practices on estimation and project success. Understanding the role of testing effort estimation in software projects better may help project managers, other software professionals and researchers to pay more attention on testing and related estimation.

The rest of this paper is structured as follows. Section 2 briefly presents the background of software cost estimation studies. It is followed by a description of research design and the case study subjects. The fourth section presents the results and Section 5 discusses the practical and scientific implications, and addresses the limitations of the study as well as the future research directions. The final section concludes the study.

2 Background

Software cost and effort estimation has a long tradition in the field of computing disciplines. For example, Farr and Nanus (1964) and Nelson (1966) addressed the cost estimation of programming five decades ago. Since then a multitude of software cost estimation methods and models have been presented (c.f. Boehm et al., 2000; Briand and Wieczorek, 2002; Jørgensen and Shepperd, 2007). However, as discussed in the introduction, the software industry has an infamous history with the success of software cost and effort estimations. Despite the decades of research, projects often run over their budgets and timetables (Moløkken and Jørgensen, 2003).

Software testing is showed to be hard to estimate and predict. In the survey of Australian companies by Ng et al. (2004), only 11 companies out of the 65 studied were able to meet their testing budget estimates. Most of the companies spent more than 1.5 times the resources they had estimated. Furthermore, three of the studied companies estimated the cost of testing twice as high as was the actual cost. At the same time, most of the companies reported using approximately 10-30% of their initial budget to the testing activities.

The extant literature has classified presented estimation methods in several ways. For example, Boehm, Abts, and Chulani (2000) divided the existing methods into six different categories: Model-based, Expertise-based, Learning-oriented, Dynamics - based, Regression-based and Composite techniques. However, for the sake of simplicity, we consider only two major types of software cost estimation methods: Model-based and Non-model-based (Briand and Wieczorek, 2002). The first category, in general, consists of methods that have a model, a modelling method, and an application method. A classical example of these kinds of software cost estimation methods is COCOMO by Boehm (1981). The methods of the second category consist of one or more estimation techniques. These methods do not involve any models, only estimations. For example, bottom-up expert estimations belong to this category.

Some methods for software test effort estimation have been presented. For example, Calzolari, Tonella and Antoniol (1998) presented a dynamic model for forecasting the effort of testing and maintenance. The model is based on a view that testing is a similar activity to predating a prey (i.e. software bug) in nature. Engel and Last (2007) have also presented a model for estimating testing effort based on fuzzy logic. In contrast to modelling-focused techniques, Benton (2009) recently presented the IARM-estimation method for software cost and effort. The model is remarkably simpler than presented formal models and it emphasizes expert evaluation by team members; however, there is a lack of validation and verification of the proposed model.

To summarize, software testing activities seem to be rather hard to estimate. While there is a lack of studies on software testing estimations, Ng et al. (2004) showed that most of surveyed Australian companies do not hit their estimates. Furthermore, there is a rather wide understanding on the size of software testing of the whole software development project. While some studies suggest the use of only one-fourth of the budget (e.g., Haberl et al., 2012; Lee et al., 2013; Buenen and Teje, 2014), there have been claims of even half of the budget (see e.g. Ammann and Offutt, 2001). Nevertheless, in our literature review, we sparsely found any studies focusing on developing or validating software cost estimation techniques for testing. This is a noteworthy disparity between the cost caused by testing activities and the academic interest towards the topic.

3 Research methodology

In the following, we will first present the research approach and design of this study. It is followed by a description of case study subjects.

3.1 Research design

This study is based on a qualitative research approach (Cresswell, 2003). We use a case study research strategy and interviews as the main tools of inquiry. The qualitative research approach was selected to allow us to get an in-depth understanding about the phenomenon under the study lens. The case study research strategy was used as the researchers have no control over the study subject (Yin, 2003). As Patton (2001) states, the case studies are able to shed light on phenomena occurring in a real-life context. This study is *exploratory* of type, finding out what is happening, seeking new ideas and generating hypotheses and ideas for new research (Robson, 2002). The research uses a multiple case study design following a replication logic (Yin, 2003). The case study organisations were selected based on the impression of the SCE maturity of the organisation gained during the initial discussions with the organisation representatives. The unit of analysis is a single software cost estimate. The study is focused on the experiences gained during the preparation of the cost estimate.

This study about estimation of software testing consists of two case companies that we call as *Small Global* and *Large Multinational*. The companies wished to remain anonymous in this study. From both companies, we selected one software implementation project that has either ended or is near its ending. The first project was developed for a customer, thus the estimate was used for pricing the project, in addition to other planning purposes. The other is a software tool development project, which is aimed for a mass market. In this project, the estimates were especially used for estimating the release date of the product.

For the selected case study projects, we interviewed different stakeholders involved in the projects. The interviewees' roles varied from developers and testers to project managers and senior executives. In addition to the interviews, we collected various documents related to the project. For example, we reviewed project plans, design documents and minutes of weekly meetings related to the projects.

We created an interview protocol consisting of several questions related to software cost estimation and testing. The protocol was created following the guidelines by Runeson et al. (2012). The interviews were conducted as semi-structured (Robson, 2002). In the interviews, while following the protocol, new ideas were allowed to be brought up and discussed with the interviewee. The interviews were conducted by two researchers where one acted as the main interviewer. An interview session was approximately one hour in length and the discussion was recorded. The recordings were transcripted and sent to the interviewees for review. All case subjects participated in the study voluntarily and anonymously, and the collected data was treated as confidential.

For the analysis of data, we used nVivo 10. All transcripted interviews, notes done during the interviews, in addition to the auxiliary materials, were imported into the software. The analysis was conducted in a series of steps (Robson, 2002). First the texts were coded by the researchers, whereafter iteration followed, until the conclusions were reached.

3.2 Case subjects

The Small Global is a software producing firm of about 100 persons, headquartered in Finland. The company's line of business consists of selling consultancy and support services in addition to software products to businesses. The company is global; it has customers and offices in several countries. The selected project, referred to as *Developer Tool,* is a typical software product development project in the company. In the project, the aim was to produce a visual design tool for developing applications. The project followed a Waterfall-style software engineering method: it was strictly planned up-front but the actual development work was divided into sprints. The interviews conducted in the company are shown in Table 1.

The Developer Tool project started with a prototype where technical challenges and possible development stacks were studied. After the prototype project, a project aiming at the release of version 1.0 was planned. The management named a project manager and a product owner for the product. The product owner crafted a design document for the product, and based on that document, the project management created a project plan with time estimates. Initially, the project was estimated to take three months with a team of four people.

The project missed its first deadline, a beta version release, approximately after the half-point of the estimated duration of the project. First, the project team narrowed the content plan of the product in order to hit the planned release date. Later, the initial content was returned to the plan and deadlines were postponed in the future. When the problems were noted, some developers were changed and new ones were introduced. Currently, the project is expected to finish after approximately nine months of development with the team of approximately four persons.

Table 1. Interviews done in this research

Interview code	The role of interviewee
	Small Global
Interview V_1	Key informant interview (Product owner)
Interview V_2	Senior manager of *Developer Tool*
Interview V_3	Product owner of *Developer Tool*
Interview V_4	Senior manager of *Developer Tool*
Interview V_5	Project manager of *Developer Tool*
	Large Multinational
Interview T_1	Key informant interview (Project manager)
Interview T_2	Project manager of *Operational Control System*
Interview T_3	Senior manager of *Operational Control System*
Interview T_4	Software test manager of *Operational Control System*
Interview T_5	Requirements expert of *Operational Control System*
Interview T_6	Senior developer of *Operational Control System*

The other case study subject comes from The Large Multinational, which is a large multinational company also headquartered in Finland. The Large Multinational produces software and consultancy for a wide area of business sectors. The selected project, referred to as *Operational Control System*, is a typical software development project for the company. The resulted software is a business intelligence reporting system for following certain control activities. The software was ordered by a long-term customer of the company. The interviews conducted in the company are shown in Table 1.

Also this project followed a Waterfall-like software development process: the requirements were elicited before the features of the product were agreed upon with the customer. The product was estimated only after the specification documentation was ready and accepted by the customer. The estimation was done by the developers and testers themselves. The Large Multinational uses a structured sheet for estimations that the workers filled individually by themselves. The project manager prepared the final estimates by using the expert estimates as the base.

The Operational Control System project was planned according to certain preconditions: the customer had a fixed budget and limited time for development. Only after both the customer and the software vendor had agreed upon the content with a limited set of unknown features, the development started. The development work continued rather straightforwardly from design and implementation to testing and delivery. The project lasted 10 months; the size of the project was approximately 30 man-months, of which 25% was used for testing and quality assurance. While there were major changes asked by the customer in the midway of the project, it hit the targets by keeping the budget and the timetable. In certain areas, there were estimation mismatches: in one software testing area, there was a significant overrun (+76% of initial estimate); however, in the implementation of a certain feature, an expert level developer was able to underbid the estimate (-77%) that was created with the presumption of a general level developer.

4 Results

This section presents the findings identified during the analysis of data, as described in the research methodology chapter. The findings are grouped into the following five categories:

1. General
2. Estimation practices
3. Testing plan
4. Attitudes
5. Estimation challenges

4.1 General

In both projects testing was strongly related to effort and schedule overruns. In the Operational Control System project the overrun of the data import testing was nearly 80%, although the overall project managed to keep the original budgets. In the Developer Tool project, omitted testing tasks resulted in significant effort and schedule overruns, being roughly 100% in July 2015. Regardless of the significant testing related overruns, the management in both projects report that the actual amount of work would probably not have been affected by more accurate estimates. In other words, the actual testing work was necessary, although bigger than anticipated.

The overrun of the testing effort in the Operational Control System lead to discussions with the customer, and likely to decreased customer satisfaction. In the Developer Tool, the project has received a late project status because of the overruns. This has lead to some negative late project symptoms, like decreased development team motivation, increased number of status meetings, frequent re-estimation and more discussions about the project scope (McConnell, 2006).

Both organisations have a long experience of software projects and related testing. The unit responsible for the Operational Control System has dedicated testing engineers and teams, while in the Developer Tool project the testing was conducted partially by dedicated testing engineers, partially by the development team. In both cases customer representatives or pilot users participated in user testing. Following the Testing Maturity Model (TMM) presented by Burstein et al. (1998), the testing maturity was on level 3 and level 2 in Operational Control System and Developer Tool, respectively, in the scale from 1 (low maturity) to 5 (high maturity).

4.2 Estimation practices

There were standard procedures for the estimation of the testing effort in both of the companies. The procedures required that the project plan, functional specification and testing plan must be ready before the effort estimation. Expert estimation was used for estimating testing tasks in both projects, i.e. experts estimated the effort of the testing tasks based on their professional competence. In the Operational Control System project a spreadsheet based work breakdown structure was used for preparing the final estimate, and the effort was also compared with the historical project data. There was also a guideline based on the historical data that the testing effort varies between 50% and 200% of implementation effort, depending on the application area to be tested. The expert estimation of testing tasks was conducted by the actual testing engineer in the Operational Control System project and by the product owner in the Developer Tool, while feature implementation tasks were estimated by the actual software engineers in both projects. In the Operational Control System, the actual testing engineers were known by the time of estimation, which was not the case in the Developer Tool project. Finally the estimates were reviewed in the project steering groups.

The fact that the actual testing engineer estimated the testing tasks was seen to have a positive impact on the estimation results in the Operational Control System project, as well as the experience of the estimator. The Operational Control System team members report also that knowing the actual persons who will be doing the testing helps in preparing estimates. That is, the amount of work hours needed depends on the persons doing the work.

4.3 Testing Plan

Regardless of the requirement by the standard estimation procedures, neither of the projects had a testing plan ready, when the implementation phase of the project was started. In other words, the exact scope of the testing was not defined in detail before testing was estimated. In the Operational Control System project the testing effort was accounted in the project plan, and the testing manager reports that there was a shared vision of the testing based on the previous projects. In the Developer Tool project only automated regression testing was accounted in the project plan, but user testing and manual testing were omitted. In both projects the testing plan was created in a later phase of the project. The reason for not finalizing the testing plan before starting the implementation was a hurry to proceed with the implementation, not to save money.

Interviewees in the Developer Tool report that there was a significant difference in the expectations of the project results between the development team and the management. The management expected a finalized, user tested product, while the development team's expectation was more like a regression tested proof-of-concept, where the user testing would have been postponed until the next version of the product. The overruns in the Developer Tool project are specifically related to this difference in expectations. Although the testing activities are seen as necessary, a senior manager in the Developer Tool states that a testing plan would have helped to discover testing related problems earlier and to plan testing activities better. A proof-of-concept project was done prior to the actual Developer Tool project to test all key technological assumptions, but that did not cover testing. Retrospectively a senior manager speculates that testing should have been part of the proof-of-concept to avoid testing related surprises.

In the Operational Control System the project manager reported that the first estimate for testing did not fit in the project schedule, and it was discovered that the estimate was prepared in five minutes. The project manager had also challenged the high effort for testing activities, being not able to understand the basis for the estimate. Furthermore, the project and testing managers report that the difficulties in testing were related to e.g. unavailability of test data and higher complexity of the data import logic than expected.

4.4 Attitudes

In the Operational Control System, the testing manager describes that testing is generally considered as of low importance, and that the competence and historical data is not valued as it is valued in the implementation. This feeling contradicts with the general comments where professional competence and use of historical data are strongly connected to estimation success. The Operational Control System project owner emphasizes that the testing effort must be on a reasonable level considering the application area and that it must add value to the project. The project owner continues that Large Multinational has invested significant amount of time and money for improving testing capabilities within the past two years. This communicates about the experienced importance of testing. However, according the product owner, changing established practices will take time. Based on the interviews, changing attitudes seem to take time, like changing any other capabilities. In both projects the technical staff considered especially automated regression testing to have a high importance. In the Developer Tool project, a senior manager describes that developers' attitudes towards manual testing and user experience testing are negative.

Other findings in this research support the feeling that testing is not considered equally important as implementation. For example, the testing plans were not ready when the project was estimated or testing started, and in one project the actual testers did not estimate the testing work or the actual testers were not known at the time of estimation. The Developer Tool projects's product owner reports that testing related tasks are first omitted if the schedule is tight and something needs to be dropped out from the scope. This lower level of importance may prevent testing as a function from evolving, including estimation of testing. The testing manager in the Operational Control System pointed out that the attitudes towards testing are not encouraging even in universities: *"There was only one course of software testing out of two hundred offered"*. Additionally, the testing manager commented that testing is seen as a task for those who are not good enough for ordinary programming.

4.5 Estimation challenges

In the Operational Control System project, the testing manager considered estimation of testing as more difficult than estimating implementation. The primary reason for this is the high number of dependencies: problems with data or specifications, or high number of bugs will reflect to testing. For example, if the test data is delayed by one week, the testing team is still there, but not doing what they were planned to do. The number of found bugs was relatively high in the Operational Control System, which cumulated extra work because of manual testing practices. Also the Developer Tool project manager emphasised the role of external parties in estimation. For example, the unavailability of external testing engineers or users for user testing may delay the project. Other interviewees could not make a difference in difficulty, but a senior manager in the Developer Tool pointed out that there is generally less experience of estimating testing.

Automated testing was widely seen to make regression testing more predictable, because then the regression would not cause additional testing work. The project manager in the Developer Tool emphasises the importance of the right degree and phase of testing, otherwise the maintenance of tests can generate extra work. A senior executive of the project continues that indecisiveness in processing user feedback may cause overruns, and underlines the importance of decisiveness and time boxing in user testing.

5 Discussion

This section discusses the main case study findings, and presents the related practical and theoretical implications. This section also addresses the limitations of this study, and gives pointers for further research.

This study has captured in-depth experiences from two *typical* project organisations, who have established software development practices for one project, based on their internal guidelines. As is typical for these kinds of organisations, the maturity and optimisations of the practices are not on an exemplary level, because the projects exist for only a certain period of time, and the contents of the projects vary. The primary need of these organisations is to get the basic things right in every project, not to optimise the last details.

5.1 Implications for practice

This study clearly shows that the basic principles for estimating testing related tasks are the same as for estimating implementation tasks, but there is a strong tendency to take shortcuts and postpone tasks beyond their planned deadlines. The most significant finding is that the testing plan was not delivered in either project by the time of estimation. This can be compared to estimation of implementation with incomplete or no specifications, which is connected to overruns (Standish group, 1994; Lederer and Prasad, 1995). Especially in the Developer Tool project this was the reason for overruns, and the evidence suggests also that some surprises could have been avoided also in the Operational Control System project with a finalised testing plan.

Also, the practical estimation tasks were subject to taking short-cuts in the Developer Tool. While the implementation tasks were estimated by a developer and the initial project team was known, the testing was estimated by the product owner for an unknown team. This is problematic from the estimation point of view, because the productivity of developers is known to vary significantly (Kemayel, Mili and Ouederni, 1991).

The attitudes towards testing and its estimation seem to be contradictory: on one hand testing cannot be omitted, but on the other hand it seems not to be a full-fledged part of the software project. This research has found that the testing personnel perceives that testing is considered as unimportant, and the management emphasises that the cost needs to be reasonable. The tendency to take shortcuts supports this perceived unimportance. The extant literature clearly shows that the attitudes of the senior management influences strongly the change or improvement of a certain area (Stelzer and Mellis, 1998; Wiegers, 1998), which means that the negative attitudes are likely to influence also the estimation of testing tasks negatively.

The technical staff in both projects has very positive attitudes towards automated testing. This is claimed to reduce manual work and improve predictability. This is a fair assumption, and confirmed, for example, by Ramler and Wolfmeier (2006) and Hoffman (1999), but only if implemented properly. They emphasise that the project characteristics need to be accounted when planning testing, in order to implement effective and reasonably priced testing. This was supported by the comments from a senior manager in the Operational Control System (*"the extent of testing must consider the project at hand"*) and (*"testing must add value to the project"*), the project manager from the Developer Tool (*"degree and phase of testing must be carefully planned"*) and a senior manager in the Developer Tool (*"decisiveness and time boxing is important in user testing"*).

Changing attitudes, as well as other capabilities, seem to take time. The Large Multinational has invested resources in building capabilities significantly within the past two years, but the know-how or attitudes are still not on the same level as in implementation. This corresponds to results from other change situations (Piderit, 2000).

As a summary, similar projects as the Developer Tool and Operational Control System are recommended to follow the same estimation practices for testing related tasks as for implementation. Neither of the projects reported that the reason for postponing or omitting tasks was cutting cost. In this sense, it seems counter intuitive and productive not to follow the same rigour in testing as with implementation, especially when the short-cuts seem to cause even severe estimation errors.

Finally, it was noted in the interviews that the personnel was rarely properly educated to use the estimation tools. While the other of our case study companies offered support and education in the estimation to the project management, the development teams were not aware of these services. Furthermore, only a few mentioned university education as a source of cost estimation knowledge, even though cost estimation based on an expert's opinion is nowadays a part of the standard workflow in many companies. While effort estimation is a part of the curriculum guidelines suggested by

ACM[1], there might be a need for education organisations to re-evaluate the content of teaching.

5.2 Implications for theory

The current SCE literature provides only a few case studies providing in-depth knowledge of real-life situations (Jørgensen and Shepperd, 2007), and, to our best knowledge, this is among the first studies to report on experiences related to estimating software testing. This paper contributes to the body of knowledge by showing that organisations seem to deviate from their standard practices when estimating testing related tasks, causing estimation errors. The reason for deviations resides in negative attitudes towards testing. Cost savings are rejected as a reason for deviations, as well as poorly performed testing as a reason for overruns in testing.

Furthermore, our literature review shows that there is a lack of empirical and theoretical studies on addressing software testing effort estimation. While further work such as a systematic literature review is needed to confirm this observation, this hints that one of the reasons for software cost estimation related problems is the lack of proper focus on testing and its effort estimation. However, new estimation tools or methods for software testing are not silver bullets. A holistic view is needed for improving the accuracy of software estimates.

5.3 Limitations and further research

This study has certain limitations. First, the findings of this study are subject to constraints of the research methodology. This research studied two very similar software projects, which limits the validity of the findings to similar contexts. It is recommended that further research would be conducted in a different context to see if e.g. larger projects or continuous product development projects suffer from similar testing estimation related problems as reported in this paper. Second, a qualitative study is always, to some degree, subjective and might be influenced by the expectations and opinions of the researchers and interviewees. However, we did our best to treat the data objectively and tool countermeasures to reduce bias. For example, all transcripts and quotes, as well as the manuscript, were checked and approved by the interviewees before the publication.

Considering the big share of testing work in the overall software project and the problems reported in this study, further research of the topic is justified. First, our unstructured literature review did not reveal many studies in this topic. Thus, a more thorough literature review should focus to shed more light on possible research gaps

[1] Software Engineering 2014. Curriculum Guidelines for Undergraduate Degree in Software Engineering. https://www.acm.org/education/SE2014-20150223_draft.pdf

in software testing estimation. Second, there is a lack of studies addressing the methods and tools used in the industry to estimate software testing. Further work should be devoted to reveal the best practices already used in the industry.

6 Conclusions

This research provides evidence that software testing related tasks in software projects are estimated by using similar practices as for implementation related tasks, but deviations from the standard practice occur often. These deviations, such as estimating without a testing plan, estimating work that will be carried out by others or estimating without knowing the actual testers, have been a source for even severe overruns. The research rejects poorly performed testing as a source for overruns. Overruns themselves are a source of decreased team motivation and customer satisfaction, and additional work, among other things.

The results of this research suggest that the reason for process deviations resides in the negative attitudes towards testing. Widespread deviations from established practices among both project management and technical staff, together with the direct comments from the interviews, indicate that testing is not considered as important as implementation, and therefore deviations are allowed. The results reject cost savings as the reason for deviations.

The main implications from the results for software managers, experts, project managers and academia are the following:

- A deviating estimation process for software testing may lead to severe estimation errors.
- Software projects should use the same rigor in estimating testing as for estimating implementation. Deviations should not be allowed.
- Managers responsible for software and project processes must recognise the importance of testing and promote the importance of it to change the attitudes towards testing. This is necessary in order to establish the use of good practices in organisations.

Finally, the aforementioned serve also as a good starting point for further research.

Acknowledgements.
The authors gratefully acknowledge Tekes – the Finnish Funding Agency for Innovation, DIGILE Oy and Need for Speed research program for their support.

References

1. Ammann, P., Offutt, J.: Introduction to Software Testing. Cambridge University Press, UK (2008)
2. Benton, B.: Model-Based Time and Cost Estimation for Software Testing Environment. In: Sixth International Conference on Information Technology: New Generations, pp. 801-806. IEEE (2009)
3. Boehm, B., Abts, C., Chulani, S.: Software development cost estimation approaches — A survey. Annals of Software Engineering, 10(1-4), 177-205 (2000)
4. Boehm, B.: Software Engineering Economics. Prentice-Hall, Englewood Cliffs, New Jersey (1981)
5. Briand, L.C., Wieczorek, I.: Resource Estimation in Software Engineering. Encyclopedia of Software Engineering. John Wiley & Sons, Inc., New Jersey (2002)
6. Buenen, M., Teje, M.: World Quality Report 2014-15: Sixth Edition. Capgemini, Sogeti and HP (2014)
7. Calzolari, F., Tonella, P., Antoniol, G.: Dynamic model for maintenance and testing effort. In: Proceedings of the International Conference on Software Maintenance 1998, pp. 104-112. IEEE (1998)
8. Creswell, J.: Research Design: Qualitative and Quantitative and Mixed Methods Approaches. Second edition. SAGE Publications, Inc., Thousand Oaks, California (2003)
9. Engel, A. Last, M.: Modelling software testing costs and risks using fuzzy logic paradigm. Journal of Systems and Software, 80(6), 817-835 (2007)
10. Farr, L., Nanus, B.: Factors that Affect the Cost of Computer Programming. Volume I. Technical Documentary Report No. ESD-TDR-64-448. United States Air Force, Bedford, Massachusetts (1964)
11. Haberl, P., Spillner, A., Vosseberg, K., Winter, M.: Survey 2011: Software Testing in Practice. Auflage, dpunkt.verlag, (2012) http://www.istqb.org/documents/Survey_GTB.pdf
12. Hoffman, D.: Cost Benefits Analysis of Test Automation, Software Testing Analysis & Review Conference (STAR East). Orlando, FL (1999)
13. Jørgensen, M., Shepperd, M.: A Systematic Review of Software Development Cost Estimation Studies. IEEE Transaction on Software Engineering, 33(1), 33–53 (2007)
14. Kemayel, L., Mili, A., Ouederni, I.: Controllable factors for programmer productivity: A statistical study. Journal of Systems and Software 16(2), 151-163 (1991)
15. Laurenz Eveleens, J. and Verhoef, C.:. The rise and fall of the Chaos Report figures. IEEE Software, 27(1), 30–36 (2010)
16. Lederer, A.L., Prasad, J.: Causes of Inaccurate Software Development Cost Estimates. Journal of Systems and Software, 31(2), 125-134 (1995)
17. Lee, J., Kang, S., Lee, D.: Survey on software testing practices. IET Software, 6(3), 275-282 (2012)
18. Mieritz, L.: Surveys Shows Why Projects Fail. G00231952. Gartner, Inc. (2012)
19. Moløkken, K., Jørgensen, M.: A Review of Software Surveys on Software Effort Estimation. In: Proceedings of International Symposium on Empirical Software Engineering, pp. 220–230. IEEE (2003)
20. Nelson, E. A.: Management Handbook for the Estimation of Computer Programming Costs. Technical Documentary Report No. ESD-TR-67-66. United States Air Force, Bedford, Massachusetts (1966)
21. Ng, S.P., Murnane, T., Reed, K., Grant, D., Chen, T.Y.: A preliminary survey on software testing practices in Australia, In: Proceedings of the 2004 Australian Software Engineering Conference, pp.116-125. IEEE (2004)

22. Patton, M.: Qualitative Research and Evaluation Method, Third edition. SAGE Publications, Inc., Thousand Oaks, California (2001)
23. Piderit, S.K: Rethinking Resistance and Recognizing Ambivalence: A Multidimensional View of Attitudes toward an Organizational Change. Academy of Management Review, 25(4), 753-794 (2000)
24. Rahikkala, J., Leppänen V., Ruohonen, J., Holvitie, J.: Top management support in software cost estimation: A study of attitudes and practice in Finland. International Journal of Management Projects in Business, 8(3), 513-532 (2015)
25. Ramler, R., Wolfmaier, K.: Economic perspectives in test automation: balancing automated and manual testing with opportunity cost. Proceedings of the 2006 international workshop on Automation of software test, pp. 85-91. ACM (2006)
26. Runeson, P., Höst, M., Rainer, A., Regnell, B.: Case Study Research in Software Engineering: Guidelines and Examples. John Wiley & Sons, Inc., New Jersey (2012)
27. Standish Group: The Chaos Report. The Standish Group (1994)
28. Stelzer, D., Mellis, W.: Success factors of organizational change in software process improvement. Software Process: Improvement and Practice 4(4), 227-250 (1998)
29. The CHAOS Manifesto: Think Big, Act Small. The Standish Group International (2013)
30. Wiegers, K. E.: Software process improvement: Eight traps to avoid. CrossTalk, The Journal of Defense Software Engineering (1998)
31. Yin, R. K.: Case Study Research: Design and Methods. Third edition. SAGE Publications, Inc., Thousands Oaks, California (2003)

PIII

THE IMPACT OF A DELAYED SOFTWARE PROJECT ON PRODUCT LAUNCH COORDINATION: A CASE STUDY

By Jurka Rahikkala & et al

Originally published in *22nd ICE/IEEE International Technology Management Conference, At Trondheim, Norway*

The Impact of a Delayed Software Project on Product Launch Coordination: A Case Study

Jurka Rahikkala, Sami Hyrynsalmi
Department of IT
University of Turku, Finland
{juperah, sthyry}utu.fi

Marko Seppänen
Department of Pori
Tampere University of Technology, Finland
marko.seppanen@tut.fi

Ville Leppänen
Department of IT
University of Turku, Finland
villep@utu.fi

***Abstract*—An increasing number of today's products include software as their key component. This means that more and more product launches are depending on software projects, which are infamous for delays. While the impacts of delays are well studied in the scope of a software project and the company itself, the impacts on the management of launch activities are not very well understood. This study addresses the gap by an in-depth case study of one delayed software project. The results show that the delays may increase the cost of a product launch, as well as decrease the scope and quality of the launch activities. These impacts are influenced by key personnel's motivational factors, which in turn cause lost working time and postponing the work until it is too late to act as planned.**

***Index Terms*—software product management, product launch, project delay, change management**

I. Introduction

The global software industry has emerged to a significant business segment in a couple of decades [1], and there are no signs of slowing down in sight [2]. Software is applied in an increasingly wide spectrum of business, and it has been claimed that now every company is a software company[1]. Thus, most companies must deal with software projects and product launches, depending on them. This new situation is also a challenge for companies: while the timing of the market introduction is critical for the product's success [3], software projects tend to overrun their schedules [4].

This study focuses on the coordination between the software project and related product launch activities. A product launch refers to activities needed for bringing a new product into the market; a product launch strategy should define what to launch, where to launch, when to launch and why to launch [5]. Timing and good management of key aspects of the launch, such as marketing plans and overall launch direction, have been found as critical success factors for the launch success [6]. However, because of the nature of coordination, launching a product involving software is depending on the outcomes from the software project [7]. While the frequency and impacts of software project delays have been widely reported [4], [8] and the success factors of a product launch are known [6], there seems to be a gap in the pool of knowledge in how to successfully manage launch activities in case of a delay. Regardless of the frequency of delays, high product failure rates [9], [10] and reports about focusing on product development on the cost of market launch management [10], [11], we were, surprisingly, not able to locate a single study addressing the situation.

The research objective of this study is to address the following unanswered question:

RQ What is the impact of software project delays on managing the related market introduction in terms of cost, scope and quality of the launch activities?

This paper reports in-depth findings from one delayed software project and contributes to the field of product innovation management and software product management. The results demonstrate the impacts of delays on cost, scope and quality of product launch activities. Additionally, the significant role of motivation as a moderator of the impacts is considered. Improved understanding of the dependency between a software project and product launch activities may help top managers, product managers and researchers.

The remaining of the study is structured as follows. Section II presents the background and related work of market entry studies. Section III describes the case study subject and research design. It is followed by presentation of findings. Section V discusses the results and concludes the study.

II. Background

PIMS studies (e.g. [12]) were the first ones among the studies trying to identify profit impacts of marketing strategies. These studies developed an understanding of the causes and consequences of e.g. entry timing in order to explain successfulness at the market. A vast number of studies has been conducted in order to analyze the relationships between speed to market, quality, costs, and profitability (e.g. [13], [14]). Literature suggests that the consequences of being late to the market are significant, causing, for instance, lower profit margins, higher development and production costs, and lessening of the firm's market value [8]. Scholars have also argued speed keeps costs in control, is associated with high-quality products [15], and helps to ensure early entrant advantages, and overall profitability (e.g. [16]).

While much of the research has revolved around strategic issues, the importance of the execution of the strategies has

[1]Kirkpatrick, D. Now Every Company Is A Software Company. http://www.forbes.com/sites/techonomy/2011/11/30/now-every-company-is-a-software-company/#1505e9a21100.

also been recognized [6]. It is known that marketing and technological execution proficiency are significant predictors of new product success [17], and, on the other hand, controllable reasons for new product development (NPD) failure include poor execution of marketing and technical activities [18], [19]. Also cross-functional integration between the R&D and marketing has been found as an important success factor in NPD [20].

However, systematic planning and execution is not easy, when software projects are involved: software projects tend to be late [4]. Previous studies have found that the reasons for estimation errors are many [21] including intentional distortion of the estimates [22]. Estimation errors have been shown to cause decreased customer satisfaction, team motivation and additional work, among other things [23], [24]. The importance of estimates is well understood in the management [25], yet the overruns continue.

To summarize, market introductions and especially their timing are essential for product success. In addition to strategic considerations, also launch tactics are understood to influence NPD success significantly. Furthermore, software projects with unreliable schedules are a challenge for the successful execution and coordination of launch activities. Regardless of software being involved in a constantly increasing number of products, there seems to be a gap in the extant literature of what are the impacts of a software project delay on the launch activities.

The classic project management triangle of scope, costs, and schedule is commonly used for studying the goals of a project [26], and Dvir and Lechler have distinguished between two types of changes: plan changes and goal changes [27]. Plan changes refer to the environment and prevent the project from following the original plan. Goal changes refer to the project scope: changes in requirements or inability to meet them within the available budget and time [28]. For the purposes of this study, we employ these commonly used attributes, and focus on the changes in scope, quality, and costs to see what kinds of consequences project delays may have in a firm's internal launch activities.

III. Research process

A. Case company and project

The case company is a Finland-based medium-sized software producing company. It has offices in several countries and its main line of business includes selling software products and services. The case project aimed at launching an application tool for software developers. The product was completely new for the company and it was seen strategically important and generating new sales for the company.

The software project related to the launch started in mid 2014 with a prototype project. The first schedule for the commercial version of the product was set in the end of 2014. Initially, the project was estimated to be ready in three months. After two months, the schedule was extended for the first time, and the project received a late status. After 6 additional schedule extensions, the product was finally launched in October 2015. The launch project planning was started at the same time as the software project, but the implementation was put on hold before its full scale start, when the delay became evident.

The launch activities relied on videos. The marketing manager describes that video clips and YouTube were employed for the first time in large scale in a product launch. Other launch activities consisted of online advertising, product web pages, webinars, tutorials and documentation, among other things. In addition to the marketing activities, a direct sales campaign was conducted. According to the interviewees, there was plenty of manpower and money at disposal for the launch. They also describe that the company's capabilities for making a successful product launch are good. However, when launching new products for the new user groups, there is still room for improvement. Generally speaking, the product launch under the study followed the same process which the company had used several times, when launching other products. The interviewees characterized some of the previous product launches as highly successful.

B. Research approach

The selection of the case study company is based on good access to the company and the richness of the delayed project. Thus, we employ an exploratory qualitative research approach [29], more specifically a case study research strategy [30].

The case study subject, a project, aimed to create a new tool for software developers. The project was selected due to three main reasons: First, the project was delayed but delivered by the time of writing this report, which was a precondition for studying the impacts of a delay on a product launch. Second, the new product was highly expected in the company as it was strategically important. Thus, the company made an important investment into the product and it was followed carefully even by the top management, which improves the validity of the results. Third, the researchers were familiar with the company and the employees were expected to speak honestly, even about difficult topics.

The primary data gathering method for this study is semi-structured interviews [29]. Prior to the interviews, an interview instrument was created. The interviews were conducted by two researchers while one of them acted as the main interviewer. The interviews were recorded and carried out during one week in the beginning of January 2016. All interviews were transcripted and the results were sent to the interviewees for review. All participated in the study voluntarily. The company and the interviewees wished to remain anonymous in this study. The collected data was treated confidentially.

As a secondary data source, we collected different project related documents such as meeting minutes, and marketing and sales plans. In total, we interviewed seven persons, and studied 121 meeting minutes and four plan documents. The interviewees and their role descriptions, as well as their departments in the company, are shown in Table I. The

TABLE I
INTERVIEWEES' ROLES IN THE COMPANY

Role	Department	Length (min)
CEO, Product Director	CEO, Products	55
Key account manager	Sales	35
Product marketing manager	Marketing	55
Sales director	Sales	35
Sales analyst	Sales	25
Product owner (PO)	Product management	50
Marketing director	Marketing	50

analysis was conducted in a series of steps following the guidelines given by Robson [29].

IV. Findings and Results

The findings are grouped into categories, as presented in Fig. 1. When a delay occurs, it creates a need for delay management activities, which, according to the findings, influence marketing and sales tactics especially through motivational factors. The impact on sales and marketing tactics is observed through three findings categories, scope, quality and cost.

A. Delay management

1) Forums and other communication channels: Based on the interviews and studying the meeting minutes from different meetings of the case company, the delays were managed primarily in four different weekly meetings:

PRODUCTS WEEKLY: Chaired by the CEO for most of the duration of the project. Other participants were product owners, including the product owner (PO). The schedule was reviewed in every meeting.

MANAGEMENT WEEKLY: Chaired by the CEO, other participants being management team members, including marketing and sales director. The schedule for the beta release was reviewed regularly.

MARKETING WEEKLY: Chaired by the marketing director, other participants were the whole marketing team, including the product marketing manager responsible for the launch. The schedule was reviewed occasionally.

SALES WEEKLY: Chaired by the sales director, other participants being the whole sales team. The schedule was reviewed in every meeting.

The primary forum for the delay and schedule management was the products weekly meeting, which was chaired by the company CEO. The CEO tells that the information on the updated schedule was available for the whole company in the meeting minutes, and passed forward to other meetings through informal discussions by him and the PO. The document review confirmed that the meeting minutes from all of the previously mentioned meetings were publicly available for the whole company. The marketing director and product marketing manager describe having got information on weekly basis from the PO. The sales director got information in the

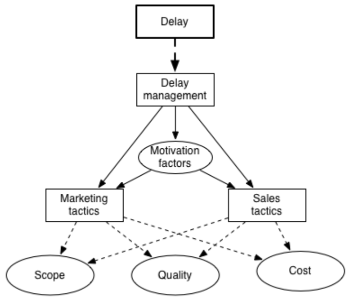

Fig. 1. The findings overview.

management team meetings and from the CEO, and passed the information forward to the sales team members.

The interviewed marketing manager, product marketing manager and sales analyst tell that they would have needed updated information about the product features for their launch related work, but this was not available. According to the minutes of the meetings, the features of the product were not discussed in any of the above mentioned meetings either. The product marketing manager describes the trustworthiness of any schedule and scope information having been low after a couple of updated and missed deadlines and scope changes. The PO concurs that when time passed, a lot of changes regarding the scope and schedule were made, and the status awareness probably suffered from that. As a result of the lack of trustworthy information, the PO describes that he received a lot of direct enquiries regarding the schedule, directly from the involved parties.

2) Schedule and features information availability: The schedule for the final release was updated seven times in the products weekly meeting during the project, as shown in Fig. 2. The beginning of the lines represents the date when the schedule was updated, while the right hand side end represents the planned release date for the final product. The thick gray lines indicate that the release date has been expressed as a range, and no exact date has been given for the release.

The schedule was updated once in January (R1), February (R2) and March (R3), the latest launch date being "in the first half of the year". Thereafter the schedule was updated next in early August (R4), followed by three consecutive updates (R5–R7). The interviewed sales director and key account manager describe that no schedule information was available during the sales campaign, which started in the beginning of August, even if asked from the PO. This confirms the gap in updating the schedule during the summer months. The product was finally released in the end of October.

The CEO and PO say that the scope was pretty stable during the whole project. However, the PO describes that there

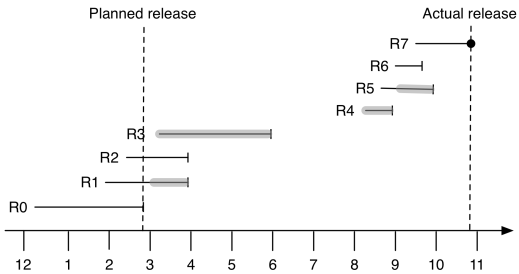

Fig. 2. Schedule updates for the final product release.

were attempts to reduce the scope because of the schedule pressure, but finally the original scope was restored. The product marketing manager also names a couple of items that were dropped from the scope early in the project. The PO and product marketing manager conclude that all this together had probably caused a perception of a changing scope. This was confirmed by the interviewees from the marketing and sales departments, who perceived that the scope of the product was changing.

3) Decision making: According to the CEO and other interviewees, the PO was accountable for the overall release of the product. The PO describes that he first re-estimated the schedule with the development team, and the updated schedules were accepted by the CEO as such. However, the PO adds that the targeted release dates were always known by the team at the time of re-estimation, and that the knowledge of the release dates had most likely affected the estimation work. The PO concludes that *"If you know that the targeted release date is after one month, you are not likely to present an extension of half a year in the schedule."* The CEO describes the product under the study having an important role in the company's strategy. He continues that the investors of the company were also following the project closely, and that there was a high pressure to release the product successfully, and realise the projected sales. The PO and the product marketing manager describe that because of the deep involvement of the CEO in the project, the whole team was well aware of the schedule and product related targets.

4) Summary of delay management practices: A covering meeting structure has been in place. The schedule has been reviewed regularly in the products weekly meeting, and the schedule has been available in the meeting minutes. However, the interviewees in sales and marketing describe the available information not being trustworthy, and that there was a lack of updated information. There were no schedule updates between April and September, even though the schedule was slipping simultaneously. These items seem to have created significant uncertainty and confusion, and triggered ad-hoc enquiries to the PO. As a response to the enquiries, the PO provided information, which differed from the targeted schedule. The scheduling decisions have been made by the PO, based on the estimates affected by business targets. The scope of the product was perceived to be changing, and although the feature information was requested and would have been needed, it was not provided.

B. Motivation factors

Both the marketing director and the sales director describe that their teams have experienced low motivation during the product launch and the sales campaign. The marketing director attributes the decreased motivation to the overly optimistic official schedules, not to the delay itself. Many marketing activities, like video recordings and messages of the key benefits, were depending on the product and its features. Without the final product, no recordings could be done, because the product was changing constantly. Also, the feature set of the product was perceived to be changing, making it impossible to know, which features and related benefits the final product would have. Thus, aiming at a moving target caused the production of the marketing artefacts to be postponed, since reproduction would be considered demotivating and unreasonable.

The sales director describes that they had no monetary targets for the sales campaign, because the product was not ready to be sold. Instead, they were demonstrating the product to potential customers. The sales director describes the motivation having been low, because the product quality was low and there was very little support for the sales from the products and marketing. The sales was conducting a campaign, from which they did not expect sales, but there was a risk of losing a credibility in front of the existing customers due to a low quality product. The interviewed key account manager describes the focus of the demonstrations having been in avoiding known bugs, and the sales analyst estimates the campaign having caused lost of sales in other product categories.

The PO and the product marketing manager had a monetary performance bonus bound to the release schedule. These bonus targets were not updated with the delays, and it became obvious in an early phase that the targets will not be reached. The PO tells having pretended that the bonus target does not exist, and that he could not let it affect the needed decision making: releasing something that is not ready does not contribute to the company's goals. He continues that many other important aspects, such as quality, were missing from the bonus scorecard. The product marketing manager comments that missing a target, which was perceived as unrealistic, in an early phase, was demotivating. The interviewed key account manager comments that even though the sales targets were not depending on the product, the time spent on the sales campaign not generating sales worked against achieving his goals.

C. Marketing tactics

1) Scope: According to the CEO, the scope of the launch was decreased significantly because of the delays. There were a lot of planned videos, tutorials, blog posts, webinars, documentation and other material, of which a significant amount was not ready by the product launch. This proposition gets support from the product marketing manager, who describes having postponed everything as much as possible to avoid doing everything again later. According to him, the last video edits were made in a hotel room the night before the launch event. The marketing manager continues that the videos were affected the most, more videos were planned.

2) Quality: The product marketing manager considered the impact of the delay having influenced the quality of the marketing activities the most. According to him, there were a lot of uncertainties related to the product's features and related benefits, which made creating marketing messages difficult. Long waiting times, uncertainty and noise in the communication caused the focus to be lost and the creativity was not at its best. Furthermore, the schedule was tight after timeboxing the development and deciding upon the release date in the trade show. Suddenly there was a hurry to do all the postponed work. Considering the previous and the reported reductions in the scope of the marketing activities, it seems likely that the level of finalization of the marketing artefacts was not as good as planned.

3) Cost: The marketing director describes that they lost roughly three persons' work effort for three months during the summer. The team was just waiting for the technical product release getting closer, so that they could start working with the related marketing artefacts. The marketing director speculates that it could have been possible to get something out of the waiting period, but it would have been difficult to motivate the team to do work, of which 80% would have needed to be reproduced later. The CEO describes that the prolonged launch required additional coordination work with e.g. an external video production company, as well as internally. The product marketing manager adds that the video material had required additional editing, causing a minor cost increase.

D. Sales tactics

1) Scope: The original target of the sales campaign was to sell licenses to the new product. Because of the delay, the target was updated to include the introduction the product to potential customers, collect feedback and get reference customers. According to the interviewed key account manager, the sales did not want to demonstrate the product to existing customers because of the low quality, and therefore the risk of losing credibility. This caused that not all customers were contacted as planned. The sales director adds that reaching customers in August was difficult because of vacations, and they did not get reference customers. The CEO describes that the collected feedback validated the value proposition, but the PO assesses that the feedback started to repeat the same problem reports quickly, and it was not very relevant. The sales director tells that the campaign was ended prematurely after one and a half months because of the aforementioned problems, instead of continuing for the planned three months.

2) Quality: According to the interviewees from the sales team, they did not have sales material, necessary training and sales support available during the campaign, which prevented them from giving product demonstrations to the potential customers by themselves. The PO confirms this. He was on vacation when the campaign started and after returning from the vacation, prioritized other things. As a result the sales team did not give product demonstrations, but invited potential buyers into demonstration events, where the PO demonstrated the product. According to the key account manager, the discussions were also more about informing the customers of the product instead of selling, because there had not been a marketing campaign prior to the sales campaign, making the potential customers aware of the product.

3) Cost: The CEO states that the sales campaign was a waste of time, because the product was not ready to be sold. The total lost working time in the sales was estimated to be 10 person months. The interviewed sales analyst comments that probably also some sales of other products were lost because of focusing on an immature product instead.

V. Discussion, Limitations and Conclusions

This study has captured in-depth experiences from a launch of a delayed software development tool product for the global market. As is typical for delay situations, many things did not go as planned, and the effects of the delays escalated in the product launch related activities. When a delay is a fact, preventing the difficulties from escalating and mitigating their impact on related activities becomes a priority.

A. Key findings and analysis

This study clearly shows that delays in a software project have a remarkable impact on the cost, scope and quality of a product launch. The most significant finding of this study is that the uncertainty of the software project schedule and product features makes planning and scheduling launch activities difficult, which may decrease the motivation of the involved personnel. The decreased motivation in turn leads to postponing the implementation of the launch activities, until there is not enough time for implementing them in the planned scope and quality. The decrease of motivation is mainly driven by the fear of needing to reproduce significant amounts of work because of the changes in product features and related customer benefits.

The schedule was updated by repeatedly extending the schedule by a short period of time. The schedule extensions seemed to be driven by the pressure to get the product out, rather than assessing the situation purely from the project realism's point of view. The estimation seemed to be affected by the business goals. Furthermore, there was a connection between the scope and schedule [26], also making the scope live with the schedule, although the eventual scope changes were minor. The launch team was facing a situation where there was no trustworthy schedule and scope available. This created uncertainty and mistrust, and decreased the motivation because any changes would mean significant reproduction of the launch related marketing artefacts.

Practice for updating the schedule correlates well with the previous findings. Excessive pressure occurs in 75% to 100% of large projects [31], which is often caused by the response of managers, when the schedule does not align with the business targets [23]. The technical staff is also known for being poor at defending their estimates [32]. Furthermore, the project seems to have experienced the anchoring phenomena [33], where the estimate is affected by an expressed starting point, i.e. the targeted release date. Sometimes managers may also want to launch products despite the fact that the products are compromised in terms of functionality and reliability [20]. Appreciation of work and interesting work have been found to be among the five most important employee motivation factors [34], and therefore it seems not surprising that the fear of losing the results of the work and needing to repeat the work is described as decreasing the motivation. The evidence suggests that the decreased motivation and related consequential effects might have been avoided by a more realistic rescheduling of the launch, and publishing a clear master schedule and scope to the software project, which would have been effectively rolled out in the organisation, making all involved parties aware of them. This would likely to have enabled a more effective coordination between the software and launch projects. The best project results are suggested to come from the most accurate estimates [35].

Personal scorecards were seen to have decreased the motivation instead of driving towards the goals. The scorecards were bound to the schedule of the product launch, while the involved personnel saw other topics, such as the quality of the work and the scope of the work, more important. Also, the performance indicators were not updated, although the project received a late project status eight months before the launch. This was perceived as unreasonable. Monitoring too many conflicting goals, like cost, budget and schedule, have been reported to be connected to demoralized staff, and staff ignoring all of the measures or shifting emphasis from one to another [36]. A reward becomes a demoralizing punishment, if it is missed out.

Postponing the work lead to a lack of time to implement the planned launch activities with the planned quality. The activities were planned early with the assumption that there is plenty of time to be used, but when the actual launch date was set, there was significantly less time to be used, although most of the work remained. This was reported to have lead to cutting out parts of the planned actions, as well as implementing them with a reduced scope or weaker quality. The additional spending originated mostly from lost working time resulting from waiting, reproducing changed items and additional coordination. All of these impacts are consequential, caused by changes in the launch schedule and scope. Coordination is managing dependencies between activities [7]. There was a clear producer / consumer dependency between the software project and the launch, as further described by Malone and Crowston [7]. The products from the software project were pre-requisites for the launch activities. In the light of the previous it seems clear that problems in the software project caused consequential effect in the related product launch.

Finally, the delays caused various other impacts on the product launch because of the previously described dynamics. For example, under pressure a sales campaign was started before the product was ready, and while the PO and customers were on summer vacation. Uncertainties also caused an extensive number of status enquiries for the PO, which prevented him from attending more important tasks, like training the sales and creating sales support material.

B. Managerial implications

We recommend that when a delay in a software project is unavoidable, the escalation of negative effects to product launch activities should be prevented or mitigated by implementing effective delay management actions. Especially important is to reduce the uncertainty by using the best effort to prepare a new, reliable master schedule and scope for the project, and making the involved parties effectively aware of them. This makes it possible to also update the respective launch plans. If done early enough, the marketing and sales activities may be able to continue their ordinary course of actions, until the product has reached the planned maturity to work as a basis for the launch activities. A clear master schedule is also likely to improve the work motivation.

Furthermore, we recommend carefully considering different scenarios before setting personal incentives bound to a launch date, especially in the marketing and sales activities, where the persons have limited influence over the schedule. Considering the track record of software projects overrunning their schedules, the scorecards bound to schedules are more likely to do harm than good.

C. Implications for theories

This paper contributes to the body of knowledge by showing that the delay impacts escalate easily from the software project to the launch activities. The impacts include increased cost, and decreased scope and quality of the launch. The reason for the negative impacts results from a missing reliable master schedule and scope, which decreases the motivation and makes a meaningful planning of launch activities difficult. The reason for the decreased motivation is an immature product and fear towards needing to reproduce work items because of the changes in the product. The decreased motivation itself causes persons involved in the launch to postpone all work as long as possible, which leaves not enough time to implement all of the planned launch activities, and not with the planned quality.

Furthermore, the impacts of a delayed software project are studied thoroughly in the scope of the project itself, and in the scope of the company. However, the impacts on the company's internal activities have not been in the focus. This paper clearly shows that the impacts can be significant also on the internal coordination, and cannot be ignored. A broader and holistic view is needed when assessing and managing the impacts of a runaway software project inside a company. Therefore, we would suggest the field of software product management to pay more attention on software project delay management, as well.

D. Validity discussion and further research

This study has certain limitations. First, the findings of this study are subject to constraints of the research methodology. This research studied only one software project, which limits the generalization of the findings to similar contexts. To improve transferability, we have given rich information about the case, thus allowing researchers to compare their cases against the one described in this study. Second, obviously mistakes were made when managing the delay in the case company, but nonetheless the results draw attention to the importance of the existence of a master schedule and scope. It is recommended that further research be conducted in different contexts to study the delay management especially in cases, where the delay has been managed successfully, and no decreased motivation or impacts on the launch activities are reported.

Third, to reduce inherent subjectivity of a case study, potential biases were attempted to reduce, e.g., by checking and approving all transacripts and the manuscript by the interviewees. The results of this study were also supported by an exemplary trail of meeting minutes. Finally, the confirmability of the results is partly supported by our literature review on non-software projects. However, further studies are needed to verify the results.

Considering the number of products containing software today, the high share of software projects overrunning their schedules and the importance of the product launch for the financial success of the product, further research of the topic is justified. Our unstructured literature review did not reveal many studies on the impact of a software project delay on a product launch, or on any other company internal activities for that matter. Thus, a more thorough literature review should focus to shed more light on possible research gaps in the impact of a delay on company's internal coordination.

E. Conclusions

This research provides evidence that the delays in a software project increase costs and decrease scope and quality of product launch activities. These impacts are mainly the results of an uncertain schedule and scope, which causes lost working time and reproduction of work within the launch project, and postponing the launch activities until there is too little time to implement all of the planned activities with the planned scope and quality. The research suggests that postponing the launch activities is caused by decreased motivation, driven by the fear of needing to repeat work phases because of changes in the immature product. The decreased motivation was amplified by personal reward systems, which were bound to the launch schedule. This goal was seen to conflict with other goals, which in turn caused ignorance of the schedule. Furthermore, the early discovered loss of reward was perceived as a punishment instead of an incentive.

The main implications are the following: 1) A delay in a software project may cause increased cost and decreased scope and quality of product launch activities. 2) A new, realistic master schedule and scope should be established immediately, when the delay becomes evident, and all involved parties should be effectively made aware of them. This may help to limit the escalation of negative impacts into the product launch. 3) A personal reward system bound to the launch schedule may lead to conflicting goals, decreased motivation and ignorance of the goals. 4) Top managers, software managers and sales and marketing managers must recognize the widespread impacts of a software project delay, and manage the situation holistically from the whole company's perspective, instead of individual departments' perspectives.

Finally, the study showed the importance of the delay management in software industry as well as noted lack of studies in this area regarding product launch processes. Further studies are with more companies and industries are needed to validate the results. Furthermore, the interaction of feature and requirement changes should be addressed in future studies.

REFERENCES

[1] M. Campbell-Kelly, "Development and structure of the international software industry, 1950-1990," *Business and Economic History*, vol. 24, no. 2, pp. 73–110, 1995.

[2] Gartner, "Gartner says worldwide software market grew 4.8 percent in 2013," Press release, March 2014.

[3] R. J. Calantone and C. A. Di Benedetto, "The role of lean launch execution and launch timing on new product performance," *Journal of the Academy of Marketing Science*, vol. 40, no. 4, pp. 526–538, 2012.

[4] K. Moløkken and M. Jørgensen, "A review of software surveys on software effort estimation," in *Proceedings of International Symposium on Empirical Software Engineering*. IEEE, 2003, pp. 220–230.

[5] E. J. Hultink, A. Griffin, S. Hart and H. S. J. Robben, "Industrial new product launch strategies and product development performance," *Journal of Product Innovation Management*, vol. 14, no. 4, pp. 243–257, 1997.

[6] C. A. Di Benedetto, "Identifying the key success factors in new product launch," *Journal of Product Innovation Management*, vol. 16, no. 6, pp. 530–544, 1999.

[7] T. W. Malone and K. Crowston, "The interdisciplinary study of coordination," *ACM Computing Surveys*, vol. 26, no. 1, pp. 87–119, 1994.

[8] K. B. Hendricks and V. R. Singhal, "Delays in new product introductions and the market value of the firm: The consequences of being late to the market," *Management Science*, vol. 43, no. 4, pp. 422–436, 1997.

[9] J. T. Gourville, "Eager sellers and stony buyers," *Harvard Business Review*, vol. 84, no. 6, pp. 98–106, 2006.

[10] E. Sivadas and F. R. Dwyer, "An examination of organizational factors influencing new product success in internal and alliance-based processes," *Journal of Marketing*, vol. 64, no. 1, pp. 31–49, 2000.

[11] L. C. Troy, T. Hirunyawipada, and A. K. Paswan, "Cross-functional integration and new product success: An empirical investigation of the findings," *Journal of Marketing*, vol. 72, no. 6, pp. 132–46, 2008.

[12] P. W. Farris and M. J. Moore, *The profit impact of marketing strategy project: retrospect and prospects*. Cambridge University Press, 2004.

[13] M. A. Stanko, F.-J. Molina-Castillo, and J.-L. Munuera-Aleman, "Speed to market for innovative products: blessing or curse," *Journal of Product Innovation Management*, vol. 29, no. 5, pp. 751–765, 2012.

[14] F. Langerak, A. Griffin, and E. J. Hultink, "Balancing development costs and sales to optimize the development time of product line additions," *Journal of Product Innovation Management*, vol. 27, no. 3, pp. 336–348, 2010.

[15] B. A. Lukas and A. Menon, "New product quality: Intended and unintended consequences of new product development speed," *Journal of Business Research*, vol. 57, no. 11, pp. 1258–1264, 2004.

[16] J. Chen, R. Reilly, and G. S. Lynn, "The impacts of speed to market on new product success: The moderating effects of uncertainty," *EEE Transactions on Engineering Management*, vol. 52, no. 2, p. 199–212, 2005.

[17] H. Nukhet, C. Droge, and R. J. Calantone, "Strategic fit to resources versus npd execution proficiencies: what are their roles in determining success," *Journal of the Academy of Marketing Science*, vol. 37, no. 3, pp. 266–282, 2009.

[18] K. Atuahene-Gima and A. Ko, "An empirical investigation of the effect of market orientation and entrepreneurship orientation alignment on product innovation," *Organization Science*, vol. 12, no. 1, pp. 54–74, 2001.

[19] R. G. Cooper and E. J. Kleinschmidt, "New products: what separates winners from losers," *Journal of Product Innovation Management*, vol. 4, no. 3, pp. 169–184, 1987.

[20] J. D. Sherman, W. E. Souder, and S. A. Jenssen, "Differential effects of the primary forms of cross functional integration on product development cycle time," *Journal of product innovation management*, vol. 17, no. 4, pp. 257–267, 2000.

[21] M. Van Genuchten, "Why is software late? an empirical study of reasons for delay in software development," *IEEE Transactions on Software Engineering*, vol. 17, no. 6, pp. 582–590, 1991.

[22] A. Magazinius, S. Börjesson, and R. Feldt, "Investigating intentional distortions in software cost estimation — an exploratory study," *Journal of Systems and Software*, vol. 85, no. 8, pp. 1770–1781, 2012.

[23] S. McConnell, *Software Estimation: Demystifying the Black Art*. Redmond, Washington, USA: Microsoft Press, 2006.

[24] J. Rahikkala, S. Hyrynsalmi, and V. Leppänen, "Accounting testing in software cost estimation: A case study of the current practice and impacts," in *Proceedings of 14th Symposium on Programming Languages and Software Tools*, Tampere, Finland: University of Tampere, 2015, pp. 64–75.

[25] J. Rahikkala, V. Leppänen, J. Ruohonen, and J. Holvitie, "Top management support in software cost estimation: A study of attitudes and practice in finland," *International Journal of Managing Projects in Business*, vol. 8, no. 3, pp. 513–532, 2015.

[26] PMI, *A Guide to the Project Management Body of Knowledge (PMBOK)*, 2004th ed. Project Management Institute, 2004.

[27] D. Dvir and T. Lechler, "Plans are nothing, changing plans is everything: the impact of changes on project success," *Research policy*, vol. 33, no. 1, pp. 1–15, 2004.

[28] W. Steffens, M. Martinsuo, and K. Artto, "Change decisions in product development projects," *International Journal of Project Management*, vol. 25, no. 7, pp. 702–713, 2007.

[29] C. Robson, *Real world research: A resource for social scientists and practitioner-researchers*, second edition ed. Oxford, UK: Blackwell Publishing, 2002.

[30] R. K. Yin, *Case Study Research: Design and Methods* Thousands Oaks, California: SAGE Publications, Inc., 2003.

[31] C. Jones, *Assessment and control of software risks*. Yourdon Press, 1994.

[32] P. W. Metzger, *Managing a Programming Project* Prentice Hall, 1981.

[33] G. Whyte and J. K. Sebenius, "The effect of multiple anchors on anchoring in individual and group judgment," *Organizational behavior and human decision processes*, vol. 69, no. 1, pp. 74–85, 1997.

[34] C. Wiley, "What motivates employees according to over 40 years of motivation surveys," *International Journal of Manpower*, vol. 18, no. 3, pp. 263–280, 1997.

[35] C. R. Symons, *Software sizing and estimating: Mk II FPA* . John Wiley & Sons, Inc, 1991.

[36] J. M. Lyneis, K. G. Cooper, and S. A. Els, "Strategic management of complex projects: a case study using system dynamics," *System Dynamics Review*, vol. 17, no. 3, pp. 237–260, 2001.

PIV

TOP MANAGEMENT SUPPORT FOR SOFTWARE COST ESTIMATION: A CASE STUDY OF THE CURRENT PRACTICE AND IMPACTS

By Jurka Rahikkala & et al

Originally published in *Product-Focused Software Process Improvement: 18th International Conference, PROFES 2017, Innsbruck, Austria, November 29–December 1, 2017, Proceedings. Springer, 2017.*

Top Management Support for Software Cost Estimation

A Case Study of the Current Practice and Impacts

Jurka Rahikkala[1], Sami Hyrynsalmi[2], Ville Leppänen[1], Tommi Mikkonen[3], and Johannes Holvitie[1]

[1] University of Turku, Turku, Finland
{juperah, ville.leppanen, jjholv}@utu.fi
[2] Tampere University of Technology, Tampere, Finland
sami.hyrynsalmi@tut.fi
[3] University of Helsinki, Helsinki, Finland
tommi.mikkonen@helsinki.fi

Abstract. *Context:* Despite decades of research in software cost estimation (SCE), the task remains difficult and software project overruns are common. Many researchers and practitioners agree that organisational issues and methodologies are equally important for successful SCE. Regardless of this recent development, SCE research is revolving heavily around methodologies. At the same time project management research has undergone a major shift towards managerial issues, and it found that top management support can be the most important success factor for projects.
Goal: This study sheds light on top management's role in SCE by identifying real-life practices for top management participation in SCE, as well as related organisational effects. Also, the impact of top management actions on project success is examined.
Method: The study takes a qualitative and explorative case study based approach. In total, 18 semi-structured interviews facilitated examination of three projects in three organisations.
Results: The results show that top management takes no, or very little, direct actions to participate in SCE. However, projects can conclude successfully regardless of the low extent of participation.
Conclusions: Top management actions may also induce bias in estimation, influencing project success negatively. This implies that senior managers must recognise the importance of seeking realism and avoid influencing the estimation.

Keywords: Senior management, Software cost estimation, Project management

1 Introduction

The global software spending is growing rapidly [12]. Especially R&D spending on software has increased by 65% between 2010 and 2015 [43], driven by innovations depending more and more on electronics and software [13]. While software has become increasingly important for companies, estimating the cost of software is difficult. The annual losses from software projects are measured in billions of euros [11, 36], and software project overruns are common [9, 16, 14].

Software cost estimation (SCE) and project management (PM) are both inseparable parts of a software project, and project management should always consider estimation [17]. Therefore the reasons for overruns may also reside in SCE, PM, or other project areas [6, 35, 38]. Considering the gravity of the problem and the known positive effect of using methodologies on project success [52], both SCE and PM professionals have developed a plethora of methodologies to aid in guiding the project to a planned conclusion. In the area of SCE, hundreds of estimation methodologies have been developed [34, 22], some of which have been proven to produce accurate results, when used properly [40, 42]. Yet, overruns are common [9, 16, 14].

Recent studies show that there are severe deficiencies in applying SCE methodologies in organisations [20, 30, 33, 45, 3], although the problems have been known for decades [15, 27]. The situation is significantly better in the area of PM, where 95% of the projects report using PM methodologies [50]. This difference in the extent of use of methodologies is surprising, because SCE research is methodology heavy, having 84% of the studies focusing on methodologies [22]. At the same time PM research has undergone a major shift towards topics like management and business, having only 16% of the recent articles focusing on methodologies [25]. Especially Top Management Support (TMS) has been an important topic for PM research, and it has been found to be even the most important success factor for projects [50]. The body of knowledge regarding top management support in PM is extensive, and contains clear advice for top management for how to support projects, including *refreshing project procedures and appropriate project management assignment* [53].

Considering the previous, the estimation related problems are not connected only to methodologies, but also to how these methodologies are applied in organisations. Although SCE research is still mainly focusing on methodologies, recently topics like estimation bias [20, 19, 18], organisational inhibitors and distortions [30, 33], and top management participation [45], have become focus of the research. This paper continues on this highly relevant path of examining other than technical factors in SCE.

The research objective of this paper is to address the role of top management in SCE, and to answer the following unanswered questions:

RQ1 What are the real life top management support practices for SCE and how do they appear in an organisation?
RQ2 How much effort top management invests in participating in SCE?
RQ3 Which persons or items are affected by top management actions?
RQ4 What is the impact of TMS for SCE on project success?

In the scope of our study, when a reference to top management is made, we refer to the highest up manager, who is aware of the estimate on the basis on their responsibilities related to the studied projects. This paper provides in-depth findings from three projects in three case companies. Based on the study of 18 interviews[4], the paper contributes to the scientific literature by reporting on the current practice of top management participation in software cost estimation, and the effects of this participation in organisations. Additionally, the impact of top management participation in SCE on project success is addressed. Understanding the role of top management in SCE may better justify project managers,

[4] Due to the non-disclosure agreements, the raw data cannot be disclosed.

other software professionals and researchers to pay more attention to top management's role in software cost estimation.

The remainder of the study is structured as follows. Section 2 presents the background and related work of software cost estimation and top management support for project management. Section 3 describes the case study subject and research design. It is followed by the presentation of findings in Section 4. Section 5 discusses the results and Section 6 concludes the study.

2 Background

The purpose of software cost estimation, or effort estimation, is to provide the management and project leadership a clear enough view of the project to make good decisions about how to control the project to hit its targets [34]. SCE has already been studied for over half of a century, c.f. [37], and hundreds of different estimation methods have been developed [5, 22]. Still, despite of the long and extensive work on the area of SCE, many software projects fail to meet estimates.

Software cost estimation research has heavily focused on estimation methodologies; leaving organisational issues with relatively little attention. According to Jørgensen's and Shepperd's [22] systematic literature review, organisational issues have been discussed only in 16% of the reviewed articles (Table 1). Furthermore, the interest towards organisational issues is decreasing. The recent study of SCE research trends shows also that the research focus has remained consistently on estimation methodologies and techniques between 1996 and 2016 [48].

The previous may be problematic, because the SCE challenge seems to reside elsewhere than in estimation methodologies. Researchers and practitioners largely agree on this point [30, 34, 22, 27], getting support from recent studies [30, 45, 44]. Also, major industrial software development frameworks, such as CMMI [1], ITIL v3 [2] and PRINCE2[5], continue along the same lines, emphasising the importance of estimation, without giving specific advice, which estimation techniques to use. Thus, while the estimation problems seem to reside on the application of the methodologies in an organisation, the research is still focusing on the methodologies themselves, leaving a gap between the actual problem and the means to fix it.

Much of the work performed in organisations is organised as projects which is understandable because the results of projects are critical for organisations [49, 7]. Considering the importance of PM, also PM has been intensively studied for over decades which has resulted into an extensive body of knowledge. However, whereas the SCE research is still focusing on methodologies as its primary line of research, the PM research has undergone a significant shift from methodologies towards other topics, such as leadership and business. According to Kolltveit et al. [25] (Table 2), PM research related to Task and Transaction perspectives, representing technical methodologies, has decreased from 68% to 18% over the time, measured in the number of published articles. This shift of focus seems natural, since organisational issues are reported to be even more important factors in project success than technical ones [29, 10, 52]. Also, top

[5] https://www.axelos.com/qualifications/prince2-qualifications

Table 1. Distribution of published SCE articles among research topics. [22]

Perspective	-1989	1990-1999	2000-2004	Total
Estimation method	73 %	59 %	58 %	61 %
Size measures	12 %	24 %	16 %	20 %
Organisational issues	22 %	15 %	14 %	16 %
Uncertainty assessment	5 %	6 %	13 %	8 %
Calibration of models	7 %	8 %	4 %	7 %
Production function	20 %	4 %	3 %	6 %
Measures of estimation performance	5 %	5 %	6 %	5 %
Data set properties	0 %	1 %	2 %	1 %
Other	0 %	2 %	1 %	1 %

Table 2. The distribution of published PM articles among different perspectives. [25]

Perspective	1983-1987	1988-1992	1993-1997	1998-2002	2003-2004	Total
Task	49%	34%	32%	23%	12%	29%
Leadership	8%	16%	25%	28%	33%	23%
System	23%	25%	18%	19%	15%	20%
Stakeholder	1%	3%	1%	5%	6%	3%
Transaction	19%	9%	6%	10%	6%	10%
Business	0%	13%	17%	15%	29%	15%

management's interest in PM is increasing along with the number of PM related articles published in top management and business journals [26].

Regardless of the methodology heavy mainstream of the SCE research, some of the recent research has also been attending to non-technical problems, such as human bias, organisational inhibitors and distortions, as well as top management participation. Jørgensen et al. have conducted a broad and widely cited work on human bias, originating from different sources. Their studies have covered e.g. the impact of the first impression [19], customer expectations [23], irrelevant or misleading information [18], and wording [21] on the estimate. Magazinius et al. have published their results regarding intentional distortions [30, 32, 31] and organisation inhibitors [33] in SCE. Additionally, among the studies of organisational factors, Rahikkala et al. [45, 44] have studies top management participation in SCE, and Ahonen et al. [3] have found problems in the reporting effort in projects.

To summarise, although both SCE and PM are inseparable parts of a software project [17], only PM research takes a holistic view, and examines the organisational context of the respective area to any great extent. SCE continues to focus on methodological problems. This is a noteworthy observation, because the problems for software project overruns reside both in SCE and PM [6, 35, 38]. Understanding the organisational context of SCE may better help to overcome many organisational problems related to SCE, and to eliminate related sources of estimation error. This paper continues examining the organisational context of SCE, and addresses specifically the top management's role, which has been found to be of critical importance in PM.

3 Research Process

3.1 Research approach

The study is based on three anonymous case companies and projects. For each company, we interviewed stakeholders involved in the projects (Table 3) and analysed 18 documents related to the project, including project plans, design documents, and minutes of meetings.

This study is based on a qualitative research approach [8]. We use a case study research strategy and interviews as the main tools of inquiry. The qualitative research approach was selected to allow us to get an in-depth understanding about the phenomenon under the study lens. The case study research strategy was used as the researchers have no control over the study subject [51]. As Patton [39] states, case studies are well capable of shedding light on phenomena occurring in the context of real-life. This study is of exploratory type, finding out what is happening, seeking new ideas, and generating hypotheses and ideas for new research [46]. The research uses a multiple case study design following a replication logic [51]. The unit of analysis is a single software cost estimate. The study is focused on the experiences gained during the preparation of the cost estimate. The conceptual framework of the study assisting in answering the research questions is presented in Fig. 1. Additionally, we have employed the list of 16 top management support practices suggested by Rahikkala et al. [45] for studying top management participation practices.

An interview protocol consisting of questions related to top management participation in SCE was created, following the guidelines by Runeson and Höst [47]. The one hour interviews were conducted as semi-structured [46] by two researchers, and the discussion was recorded. The recordings were transcripted and sent to the interviewees for review. All case subjects participated in the study voluntarily and anonymously, and the collected data was treated as confidential.

For the analysis of data, we used nVivo 10. All transcripted interviews, notes done during the interviews, in addition to the auxiliary materials, were imported into the software. The analysis was conducted in a series of steps [46]. First the texts were coded by the researchers, whereafter iteration followed, until conclusions were reached.

Table 3. Interviewees of the research.

Small Global	Large Multinational	Tech Giant
Product Owner (KI)	Project Manager (KI)	Program Manager (KI)
Senior Business Manager	Business Manager	Line Manager
	Testing Manager	Senior Manager
Senior Technology Manager	Requirements Engineer	Requirements Engineer
Project Manager	Software Developer	Head of Product Management
		Head of Programs

KI = Key informant for the study, interviewed twice

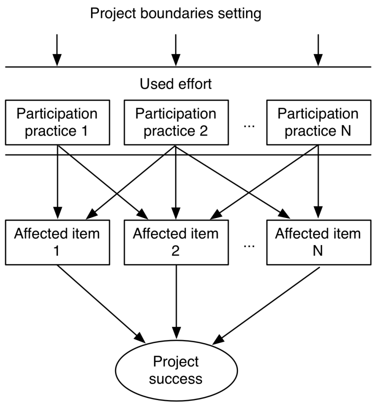

Fig. 1. The conceptual model of this research.

3.2 Case companies and projects

'*Small Global*' is a software producing firm of about 100 persons. The company's line of business consists of selling consultancy and support services in addition to software products to businesses. The company is global; it has customers and offices in several countries. The selected project, referred to as *Developer Tool* (DT), was about producing a visual design tool for developing applications. The end-result is a commercial product. The project followed a waterfall-style software engineering method, but the actual development work was divided into sprints. The estimation was done by using work breakdown structure (WBS) and expert estimation.

The DT project started with a prototype where technical challenges were studied. After the prototype project, a project aiming at the release of version 1.0 was planned. The product owner crafted a design document for the product, and based on that document, the project manager created a project plan with time and effort estimates. Initially, the project was estimated to take three months with a team of four people. The project completed nine months after the deadline with a team of approximately six persons.

'*Large Multinational*' produces software and consultancy for a wide area of business sectors. The company has tens of thousands of employees around the world. The selected project, referred to as *Operational Control System* (OCS), is a business intelligence reporting system for following certain control activities. The software was ordered by a long-term customer of the company.

Also this project followed a waterfall-like software development process. The estimation was done by the developers using expert estimation, whereafter the values were filled into a structured sheet. The project manager prepared the final estimates based on the results from expert estimation. The OCS project was planned according to certain preconditions: the customer had a fixed budget and schedule for development. The project lasted 10 months, and the size of the project was approximately 30 man-months. The project concluded successfully on time and budget.

'*Tech Giant*' is selling products with software to global business-to-business markets. The company has tens of thousands of employees around the world. We studied the *Network Management Product* (NSP) project of Tech Giant. The project produced a new release of a tool for managing the network. The project produced a new release of the system. The NSP has been in use for several years.

The project was part of a continuous development cycle involving just under 100 people. A new release of the system is developed every three months. The development methodology it used was based on Scrum with two week sprints. The development teams were distributed over several locations. The cost estimation was conducted in two phases: firstly, rough planning for the whole three month release in the product management function. Secondly, the backlog items were estimated in the scrum teams, the main responsible being the program manager. The backlog items were estimated using expert estimation. The project concluded successfully and delivered over 85% of the planned scope, which is the goal for all releases.

4 Findings and results

This section presents the findings identified during the analysis of the data as described in the research methodology section. The findings are grouped into the following five categories according to the conceptual framework (c.f. Fig. 1): 1) Project boundaries, 2) Participation practices, 3) Participation effort, 4) Practical impacts, and 5) Impact on project success. The Project boundaries were separated clearly from the participation practices because, from this study's point of view, they are related to creating prerequisites for the estimation and the project rather than directly to the estimation itself.

4.1 Project boundaries: scope, cost and schedule

Software cost estimation is fundamentally about estimating the size of the software for a given scope. The size is then converted into a schedule and budget, based on different factors, like the composition of the development team. However, there are usually boundaries for an acceptable scope, cost or schedule, originating from the business environment. Based on these boundaries, the decision makers, project management and estimators try to find an optimal balance between the previously mentioned three dimensions. This section summarises boundaries for the studied projects and estimation.

At Tech Giant, who operates in a three month release cycle, the schedule was fixed. Also the cost (resources) was fixed to a great extent, although there were some additional resources available for situations, where overruns seemed probable. Large Multinational reported that their customer also operated under a predefined system update cycle and budget framework, also fixing the schedule and cost. At Small Global, the Senior Business Manager and other team members reported that the schedule was fixed. The Senior Business Manager also reported that the planned scope was a minimum viable and nothing could not have been dropped out, making also the scope of the project fixed. Thus, for Tech Giant and Large Multinational, the only variable element was the scope, and for Small Global the resources. Additionally, the senior managers monitored the progress of the projects against the estimate regularly, and made adjusting decisions based on the situation, where deemed necessary.

4.2 Participation practices

First of all, top management did not exercise seven of the sixteen studied support practices at all, as shown in Table 4. Practices 1–16 are adapted from [45]. Additionally, the presence of three practices, 'TM ensures the involvement of the project manager during the estimation stage', 'TM ensures ongoing estimation skills training programmes' and 'TM recognizes that the estimates are inaccurate in the beginning of the project', was indirect, meaning that the presence of the practices could not be tracked back to any specific TM actions related to the studied projects. 'TM recognizes that the estimates are inaccurate in the beginning of the project' was not relevant for Tech Giant, as they are in a continuous three month release cycle, and the delivered scope must be constantly at least 85% of the planned scope. Large Multinational and Small Global had improved the accuracy with a specification phase, but this was a standard practice in both companies, like the involvement of the project manager during the estimation phase was for all three companies. Large Multinational and Tech Giant had arranged training for SCE earlier, but there were no ongoing training programs during the studied projects.

In all projects the senior managers reported that they had studied and approved the estimates. At Small Global, the Senior Business Manager studied the estimate in detail, as part of the project plan, while at Large Multinational and Tech Giant, the senior managers studied the estimates only on a summary level. Certain items in the estimates were also challenged by the senior managers in the OCS and NSP projects, which resulted in better estimates for the items in question. Considering the list of predefined 16 practices at hand, studying the estimates is close to 'TM ensures that the estimate relies on documented facts rather than guessing and intuition' and 'IT executive studies and approves the estimate'. However, as studying and approving the estimates does not fit precisely under either of the previous, we decided to report it as a new TM support practice for SCE, 'TM studies and approves the estimate'. 'TM is knowledgeable of estimation procedures' was present in the OCS project, where the Business Manager reported having been well aware of the estimation practices. This was, according to the Business Manager, coincidental rather than a result of planned actions. The presence of the four remaining support practices was strong in all case projects. The interviewees reported that the management considered the estimates having a high importance. However, none of the interviewees specified concrete examples of how the importance was demonstrated during the case projects, which means that the importance has most likely been established before these particular projects. At Large Multinational, the estimate was used for preparing an offer for a customer, who made the order decision based on it. At Tech Giant, a business plan, product roadmap and customer commitments were made based on the estimates. At Small Global, a GO/NOGO decision of the project was made based on the estimate. However, the Senior Business Manager at Small Global reported that the decision of making the product was practically made, and the estimate was used for reassuring that the scope was small or minimum viable, and that the delivery was possible in the targeted

Table 4. Exercised Top Management support practices.

Practice	Tech Giant	Large Multinational	Small Global
1. TM ensures existence of estimation procedures			
2. TM ensures that the estimator has adequate skills			
3. TM ensures improving estimation procedures			
4. TM ensures the involvement of the project manager during the estimation stage	+++	+++	+++
5. TM ensures good communication between the estimator and the organisation			
6. TM ensures that there are criteria for evaluating the meaningfulness of the estimate			
7. TM ensures ongoing estimation skills training programmes	+	+	
8. TM requires re-estimating during the project to get more accurate estimates			
9. TM ensures that the estimate relies on documented facts rather than guessing and intuition			
10. The IT executive studies and approves the estimate			
11. TM recognizes that estimates are critical to this organization's success	+++	+++	++
12. TM is knowledgeable of estimation procedures		+++	
13. TM understands the consequences of an erroneous estimate to the project success	+++	+++	+/-
14. TM can distinguish between estimates, targets and commitments	+++	+++	-
15. TM recognizes that the estimates are inaccurate in the beginning of the project	N/A	+	+
16. TM takes the output of an estimate as given without debate	+++	+++	+++
17. NEW: TM studies and approves the estimate	+++	+++	+++
18. NEW: TM ensures adequate resources for estimation	+++	+++	+++

(+) signs and (-) signs indicate evidence of assumed positive and negative presence, respectively. N/A signs for not available.

schedule. Thus, the estimate was connected to significant financial interests at Tech Giant and Large Multinational, and for making important planning decisions at Small Global.

When asked, all interviewees reported that realism and accuracy were always sought during the estimation. Furthermore, each interviewee also concluded that there was no push from the management to make the estimates smaller, and the management did not try to negotiate the estimate smaller. The Line Manager from Tech Giant says that estimates are accepted as facts, and the scope is reduced, if necessary. The Business Manager from Large Multinational says that the price can be negotiated with the customer, but not the estimate itself. However, although all interviewees at Small Global report that there was no push from the management, they also say that there was still a pressure to make the estimate smaller, conveyed by the Senior Business Manager in form of a strict deadline. The Project Manager, who was responsible for making the estimate, says that he experienced a high pressure and started to doubt his own estimates and eventually made them smaller.

As described earlier in this section, all of the projects had clear targets, or business goals, consisting of the scope, budget, and schedule. In the OCS and NSP projects the estimates were also accepted as facts which steered the planning. However, in the DT project, the Project Manager described that he made the estimate smaller because of the perceived pressure. The Senior Business Manager also told that the purpose of the estimate was to verify that the fixed scope was possible to be delivered within the target schedule, with higher resources, if necessary. The decision of executing the project was practically done. The previous signals that, in addition to creating estimates, the management seem to have expected the estimation to result into a plan, how to hit the targets, even though this seems not to have been consciously understood and intended.

In the NSP project, there was a continuous commitment to deliver at least 85% of the target scope, and at Large Multinational the normal practice was to use the estimate also as a commitment. At Small Global, the Project Manager says having been committed to the estimate in the beginning, but during the re-estimations in the later phases of the project he describes as having been afraid of giving estimates, because the estimates were taken literally by the management. Thus, estimates seem to have been implicitly taken as commitments by the management, although there was no explicit agreement on this.

In addition to the findings related to the 17 support practices reviewed earlier, resource provisioning for SCE emerged from the discussions. According to the interviewees' subjective perception, all projects had enough time and resources for preparing the estimates. At Small Global and Large Multinational, there was a separate specification phase prior to the actual implementation phase. The requirements engineer at Tech Giant reports that pre-studies are conducted, when necessary, to gain adequate understanding of the features. However, also this support practice was indirect of nature, and could not be attributed to any top management actions specific for the studied projects.

4.3 Participation effort

According to the evidence discovered during the interviews and review of the documents, top management's effort for participating in SCE was low in all case projects. In terms of time and effort, the most significant contribution was the follow-up of the progress against the estimate. This, however, is primarily connected to project management, and not to SCE. Additionally, the senior managers studied the estimate in all projects. However, as an investment of time and effort, this was relatively small. The effort related to all other participation practices could not be attributed to the studied project in particular. The practices had emerged in a longer period of time and become established routines, which do not need attention for each new project. The interviewees in all projects also confirmed that the top management did not participate directly in the estimation.

4.4 Affected items

As concluded earlier, top management sets boundaries for the project and estimation in form of budget, schedule and scope. This, however, is not influencing the estimation itself. Furthermore, the indirect support practices 'TM ensures the involvement of the project manager during the estimation stage', 'TM recognizes that the estimates are inaccurate in the beginning of the project', 'TM ensures ongoing estimation skills training programmes', 'TM ensures adequate resources for estimation' and 'TM is knowledgeable of estimation procedures' did not have any direct effects on estimation, which could have been attributed to the studied projects.

The awareness related practices, 'TM recognizes that estimates are critical to this organization's success', 'TM understands the consequences of an erroneous estimate to the project success', 'TM can distinguish between estimates, targets and commitments' and 'TM takes the output of an estimate as given without debate' did not have any tangible effects either in their positive occurrences. However, in the DT project the Project Manager reported that he had made the estimates smaller, because of the awareness of the target schedule. Furthermore, he reported that his willingness to give re-estimates during the project had decreased and he had started to give upper bound estimates, because the estimates were taken literally and interpreted as commitments. So, the awareness related support practices seem to have tangible effects on people or SCE related artefacts only, when the effects are harmful.

'TM studies and approves the estimate' was the only support practice that had direct positive impacts on estimation as a result of top management actions. After studying the estimates, managers challenged some parts of the estimate in the OCS and NSP projects. This lead to re-estimation, and improved the effort estimates for those particular functionalities.

4.5 Impact on project success

Cost estimation is an inseparable part of any software project [41], thus the cause of an overrun may reside in SCE, PM or other areas [6, 35, 38]. Not even the best project management can control a project if it has to meet unrealistic goals, while chaotic project control will usually overshoot set limits, making cost estimation meaningless. In this study our aim was to find evidence from the real-life experiences of how management's actions impact SCE, which further influences project success. Of the studied projects, two, OCS and NSP, delivered on time, scope and budget, and one project, DT, suffered from significant cost and schedule overruns.

In the two successful projects, top management's participation in SCE has been minimal, and we found very little evidence of their actions' impact on persons or artefacts during the estimation. On the other hand, top management seemed to have understood well that a realistic and unbiased estimate is critical for the success of a project and organisation. We found plenty of evidence of this understanding in both projects, although this understanding did not manifest into any concrete actions. For example, the software developer in the OCS project told that top management did not try to negotiate the estimate in any direction, customer agreements and offers are depending on the estimates. The requirements engineer in the NSP project said that top management was seeking realistic estimates — nobody wants to betray themselves, and everybody understands that without realistic estimates things will fail.

Top management's efforts for participating in SCE were equally low in the studied runaway project. But where the senior managers refrained themselves from any interference in SCE in the two successful projects, top management seemed to have influenced the estimation results by emphasising the importance of the targeted release date, and that the scope was small or minimum viable. The project manager reported having made the estimates smaller under this pressure. Additionally, implicitly interpreting estimates as commitments influenced the project manager's willingness to give estimates, and he reported having given upper bound estimates after noticing this. Although the reasons for the experienced project overruns may have been many, one of the reasons seem to have been top management induced pressure to make the estimate conform to the target delivery date. The Senior Business Manager of Small Global also attributes the overrun both to SCE and project execution.

5 Discussion

5.1 Implications for practice

Our study clearly shows that a project can conclude successfully with no, or with very little, direct top management participation in software cost estimation. On the other hand, this study presents evidence that top management's incautious interference may lead to undesired outcomes, and influence the project success negatively. The most important distinctive factor between a positive and negative top management participation seems to be to not create bias. Not creating bias manifests through understanding the negative impact of poor estimates on project and organisation success, and therefore avoiding influencing the estimation to any direction.

Previous studies have found plenty of evidence about the negative effects emerging from influencing estimation. Magazinovic and Pernstål [33] have found that management goals affect the results of estimation. Furthermore, Magazinius et al. [30] found that personal agenda, management pressure and attempt to avoid re-estimation may affect an estimate. The previous studies also show that cognitive bias may affect estimators: e.g. high or low expectations influence even experienced estimators [4], first impression may dictate a significant part of the estimation result [19], and even the wording may have a significant impact on the estimate [21]. The estimators may not even notice the influence of the expectations, or consider it to be very low [23]. The findings from the studied runaway project show, in accordance with the above mentioned studies, that it is indeed easy for top management to influence the estimation and project success in a negative way. Thus, in the light of our findings and previous studies, it seems advisable for top management to stay outside of estimation to minimise any biasing effect they may induce.

The most tangible top management participation practice in SCE was 'TM studies and approves the estimate'. Although the general recommendation seems to be staying outside of the estimation, we cannot reject the potential importance of this support practice. Studying the estimate may be a necessary action to ensure that the estimate is prepared professionally and with due care. Some other studies support the potential importance of studying the estimate: e.g. Rahikkala et al. [45] report that the extent of use for 'Top management ensures that the estimate relies on documented facts rather than guessing and intuition‘ correlates positively with project success, and Lederer and Prasad [28] recommend that computing management should study and approve the estimate.

The remaining three top management support practices that were present during the estimation, 'TM ensures the involvement of the project manager during the estimation stage', 'Top management ensures adequate resources for estimation' and 'Top management ensures ongoing estimation skills training programmes', are indirect of nature, and were not directly related to any of the studied projects. Additionally, none of these practices could be tracked back to any specific top management actions, implying that these practices were among the presumably many results of top management actions to create an overall framework for software development. Thus, because of the lack of direct top management participation, these practices cannot be considered as top management support practices for SCE, and do not seem to justify for top management's attention during SCE.

Finally, this study shows that top management invests very little time in SCE. In light of the previous findings this was expected, and even recommended, because the successful conclusion of a project did not need significant participation from top management. As is natural considering the low extent of top management participation, the footprint of their actions is also low. The results of top management actions tend to have a negative impact on project success, which was the case in the studied runaway project. The only exception for this was studying the estimate, which triggered re-estimation of certain items in the two successful projects, resulting in more accurate estimates.

5.2 Implications for theory

The current SCE literature sparsely contains studies addressing management aspects of software cost estimation [22], and, to our best knowledge, this is among the first studies to report on experiences related to top management participation practices in SCE. This paper contributes to the body of knowledge by showing that no, or very little, direct actions are required from senior management for a successful project delivery. On the contrary, the results indicate that top management must understand SCE's delicate nature prone to bias, and stay outside of the estimation to avoid any negative effects they may induce. This study also shows, from the perspective of top management that many known negative effects from biasing the estimation can also be caused by firms' top management.

Furthermore, our results show that the time top management invests in SCE is low, as well as the footprint that their actions leave on SCE related artefacts and actors. Considering the previous, the responsibility of improving SCE seems to move back towards project management and technical experts. However, as the literature has shown, methodologies are not a silver bullet, and a holistic view considering techniques, people and procedures is needed for producing more useful estimates.

5.3 Validity, limitations and further research

The qualitative case study methodology involves the researchers themselves as the instrument of the research, which poses a risk that the results are biased by the researchers' subjective opinions. As countermeasures to the validity threats, we have employed six strategies outlined by Robson [46]: prolonged involvement, triangulation, peer debriefing, member checking, negative case analysis and audit trail. Additionally, we have tried to maximise the richness of the data set by selecting different case companies and projects, improving the transferability of the results. However, as this study is explorative of nature and has not been widely examined prior to this study, generalisation of the results must be done with caution.

Overall, this study provides evidence that top management participation in SCE is low and that their participation is not needed for successful estimation. Although we believe that the results of this study can be transferred to similar settings, the situation can still vary from context to context. For example, we may have overlooked the role of some company properties, like size or maturity. Therefore, further studies in different project and company contexts are needed to see if the same phenomena are repeated, or new phenomena discovered. Quantitative studies would also provide certainty in how commonly the reported phenomena are repeated in organisations. The importance of top management studying and approving the estimate was also left unanswered in this study.

6 Conclusions

This study examined top management support for SCE by using a case study approach and interviewing 15 experts involved in three software projects in three organisations. Top management support practices for SCE were studied by employing a list of 16 predefined practices. The results show that 8 from the 16 studied practices were not present in any of the projects, and that 'Top management studies and approves the estimate' was the only tangible practice present (RQ1). This study also found evidence that the time and effort top management invested in SCE was low (RQ2), and the items or persons affected by their actions were only a few (RQ3). However, the results show further that some of the top management actions induced undesired bias on estimation, and affected project success negatively (RQ4).

The main implications from the results for managers, software experts, project managers and academia are the following:

1. No, or very little, direct top management participation in software cost estimation is required for the successful conclusion of a project.
2. 'Top management studies and approves the estimate' was the only concrete top management participation practice.
3. Top management actions may induce undesired bias on estimation, and affect project success negatively.
4. Senior managers must recognize the importance of seeking realism in estimation, and avoid inducing accidental bias in cost estimation.

Finally, the aforementioned also serve as a good starting point for further research.

Acknowledgment

The authors gratefully acknowledge Tekes – the Finnish Funding Agency for Innovation, DIMECC Oy and Need for Speed research program for their support.

References

1. CMMI for development. V1.3. Tech. Rep. CMU/SEI-2010-TR-0336, Carnegie Mellon (2010)
2. Adams, S.: ITIL V3 foundation handbook. Vol. 1. The Stationery Office (2009)
3. Ahonen, J.J., Savolainen, P., Merikoski, H., Nevalainen, J.: Reported project management effort, project size, and contract type. Journal of Systems and Software 109, 205–213 (2015)
4. Aranda, J., Easterbrook, S.: Anchoring and adjustment in software estimation. In: Proceedings of the 10th European Software Engineering Conference. pp. 346–355. ACM, (2005)
5. Briand, L.C., Wieczorek, I.: Resource estimation in software engineering. In: Encyclopedia of Software Engineering. John Wiley & Sons, Inc. (2002)
6. Cerpa, N., Verner, J.M.: Why did your project fail? Commun. ACM 52(12), 130–134 (2009)
7. Cleland, D.I.: The strategic context of projects. In: Project portfolio management – selecting and prioritizing projects for competitive advantage. CBP, PA, USA (1999)

8. Creswell, J.W.: Research Design: Qualitative and Quantitative and Mixed Methods Approaches. SAGE Publications, Inc., Thousand Oaks, California, (2003)
9. Dwivedi, Y.K., Wastell, D., Laumer, S., Henriksen, H.Z., Myers, M.D., Bunker, D., Elbanna, A., Ravishankar, M., Srivastava, S.C.: Research on information systems failures and successes: Status update and future directions. Information Systems Frontiers 17(1), 143–157 (2015)
10. Fortune, J., White, D.: Framing of project critical success factors by a systems model. International Journal of Project Management 24(1), 53–65 (2006)
11. Galorath, D.: Software project failure costs billions. Better Estimation & Planning Can Help. [online] http://www. galorath. com/wp/software-project-failure-costs-billions-betterestimation-planning-can-help.php (2012)
12. Gartner, I.: Gartner says global it spending to reach $3.5 trillion in 2017 (2016), http://www.gartner.com/newsroom/id/3482917
13. Grimm, K.: Software technology in an automotive company: major challenges. In: Proceedings of the 25th international conference on Software Engineering. pp. 498–503. IEEE (2003)
14. Halkjelsvik, T., Jørgensen, M.: From origami to software development: A review of studies on judgment-based predictions of performance time. Psychological bulletin 138(2), 238 (2012)
15. Hihn, J., Habib-agahi, H.: Cost estimation of software intensive projects: A survey of current practices. In: Proceedings of the 13th ICSE. pp. 276–287. IEEE Computer (1991)
16. Hughes, D.L., Dwivedi, Y.K., Simintiras, A.C., Rana, N.P.: Success and failure of IS/IT projects: A state of the art analysis and future directions. Springer (2015)
17. PMI: A Guide to the Project Management Body of Knowledge. PMBOK® Guide Series, Project Management Institute (2013)
18. Jørgensen, M., Grimstad, S.: The impact of irrelevant and misleading information on software development effort estimates: A randomized controlled field experiment. IEEE Transactions on Software Engineering 37(5), 695–707 (2011)
19. Jørgensen, M., Løhre, E.: First impressions in software development effort estimation: Easy to create and difficult to neutralize. In: Proceedings of EASE'12. pp. 216–222 (2012)
20. Jørgensen, M.: Communication of software cost estimates. In: Proceedings of EASE'14. p. 28:1–25:5. ACM (2014)
21. Jørgensen, M., Grimstad, S.: Avoiding irrelevant and misleading information when estimating development effort. IEEE Softw. 25(3), 78–83 (2008)
22. Jørgensen, M., Shepperd, M.: A systematic review of software development cost estimation studies. IEEE Transactions of Software Engineering 33(1), 33–53 (2007)
23. Jørgensen, M., Sjøberg, D.I.: The impact of customer expectation on software development effort estimates. International Journal of Project Management 22(4), 317–325 (2004)
24. Keil, M., Rai, A., Mann, J.E.C., Zhang, G.P.: Why software projects escalate: the importance of project management constructs. IEEE Transactions on Engineering Management 50(3), 251–261 (2003)
25. Kolltveit, B.J., Karlsen, J.T., K Grønhaug, K.: Perspectives on project management. International Journal of Project Management 25(1), 3–9 (2007)
26. Kwak, Y., Anbari, F.: Analyzing project management research: Perspectives from top management journals. International journal of project management 27(5), 435–446 (2009)
27. Lederer, A.L., Prasad, J.: Causes of inaccurate software development cost estimates. Journal of systems and software 31(2), 125–134 (1995)
28. Lederer, A.L., Prasad, J.: Software management and cost estimating error. Journal of Systems and Software 50(1), 33–42 (2000)
29. Luna-Reyes, L., Zhang, J., Gil-Garćia, J., Cresswell, A.: Software developments development as emergent socio-technical change: a practice approach. European Journal of Software developments 14(1), 93–105 (2005)
30. Magazinius, A., Börjesson, S., Feldt, R.: Investigating intentional distortions in software cost estimation — an exploratory study. Journal of Systems and Software 85(8), 1770–1781 (2012)

31. Magazinius, A., Feldt, R.: Exploring the human and organizational aspects of software cost estimation. In: Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering. pp. 92–95. ACM (2010)
32. Magazinius, A., Feldt, R.: Confirming distortional behaviors in software cost estimation practice. In: Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on. pp. 411–418. IEEE (2011)
33. Magazinovic, A., Pernstål, J.: Any other cost estimation inhibitors? In: Proceedings of the 2nd ACM-IEEE international symposium on Empirical software engineering and measurement. pp. 233–242. ACM (2008)
34. McConnell, S.: Software Estimation: Demystifying the Black Art. Microsoft Press, Redmond, USA (2006)
35. McLeod, L., MacDonell, S.G.: Factors that affect software systems development project outcomes: A survey of research. ACM Computing Surveys 43(4), 24 (2011)
36. McManus, J., Wood-Harper, T.: A study in project failure. BCS (2008)
37. Nanus, B., Farr, L.: Some cost contributors to large-scale programs. In: Proceedings of the Spring Joint Computer Conference 1964. pp. 239–248. AFIPS '64, ACM, New York (1964)
38. Nasir, M.H.N., Sahibuddin, S.: Critical success factors for software projects: A comparative study. Scientific research and essays 6(10), 2174–2186 (2011)
39. Patton, M.: Qualitative Research and Evaluation Method. SAGE Publications. (2001)
40. Pitterman, B.: Telcordia technologies: The journey to high maturity. IEEE Software 17(4), 89–96 (2000)
41. PMI: A Guide to the Project Management Body of Knowledge (PMBOK). Project Management Institute, Newtown Square, PA, USA, fifth edn. (2012)
42. Putnam, L., Myers, W.: Five core metrics: The Intelligence Behind Successful Software Management. Dorset House Publishing, New York, USA (2003)
43. PwC: Companies shifting more R&D spending away from physical products to software and services: 2016 global innovation 1000 study (2016), http://www.pwc.com/us/en/press-releases/2016/pwc-2016-global-innovation-1000-study-press-release.html
44. Rahikkala, J., Hyrynsalmi, S., Leppänen, V.: Accounting testing in software cost estimation: A case study of the current practice and impacts. In: Proceedings of 14th Symposium on Programming Languages and Software Tools. pp. 64–75. Tampere, Finland (2015)
45. Rahikkala, J., Leppänen, V., Ruohonen, J., Holvitie, J.: Top management support in software cost estimation: A study of attitudes and practice in finland. International Journal of Managing Projects in Business 8(3), 513–532 (2015)
46. Robson, C.: Real world research: A resource for social scientists and practitioner-researchers. Blackwell Publishing, Oxford, UK, second edn. (2002)
47. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering 14, 131–164 (2009)
48. Sehra, S.K., Brar, Y.S., Kaur, N., Sehra, S.S.: Research patterns and trends in software effort estimation. Information and Software Technology. Forthcoming. (2017)
49. Turner, R.: The handbook of project based management. McGraw-Hill, 2nd edn. (1999)
50. White, D., Fortune, J.: Current practice in project management–an empirical study. International journal of project management 20(1), 1–11 (2002)
51. Yin, R.K.: Case Study Research: Design and Methods. SAGE Publications, Inc., Thousands Oaks, California, third edn. (2003)
52. Zwikael, O.: Top management involvement in project management: A cross country study of the software industry. International Journal of Managing Projects in Business 1(4), 498–511 (2008)
53. Zwikael, O.: Top management involvement in project management: Exclusive support practices for different project scenarios. International Journal of Managing Projects in Business 1(3), 387–403 (2008)

PV

THE ROLE OF ORGANISATIONAL PHENOMENA INSOFTWARE COST ESTIMATION: A CASE STUDY OF SUPPORTING AND HINDERING FACTORS

By Jurka Rahikkala & et al

Originally published in *e-Informatica Software Engineering Journal*

The Role of Organisational Phenomena in Software Cost Estimation: A Case Study of Supporting and Hindering Factors

Jurka Rahikkala*, Sami Hyrynsalmi**, Ville Leppänen***, Ivan Porres****
*Vaadin Ltd
**Pervasive Computing, Tampere University of Technology
***Department of Future Technologies, University of Turku
****Department of Information Technologies, Åbo Akademi University
jurka.rahikkala@vaadin.com, sami.hyrynsalmi@tut.fi, ville.leppanen@utu.fi, ivan.porres@abo.fi

Abstract

Despite of many researchers and practitioners agreeing on that organisational issues are equally important as technical issues from the software cost estimation (SCE) success point of view, most of the research focus has been put on the development of methods, whereas organisational factors have received surprisingly little academic scrutiny. This study aims to identify organisational factors that either support or hinder meaningful SCE, identifying their impact on estimation success. Top management's role is specifically addressed. The study takes a qualitative and explorative case study based approach. In total, 18 semi-structured interviews aided the study of three projects in three organisations. Hence, the transferability of the results is limited. The results suggest that the role of the top management is important in creating prerequisites for meaningful estimation, but their day-to-day participation is not required for successful estimation. Top management may also induce undesired distortion in estimation. Estimation maturity and estimation success seem to have an interrelationship with software process maturity, but there seem to be no significant individual organisational factors, which alone would make estimation successful. Our results validate several distortions and biases reported in the previous studies, and show the SCE research focus has remained on methodologies and technical issues.

1. Introduction

Most software projects still suffer from budget and schedule overruns [19, 26, 29, 82]. Regardless of the high price of software projects that bring hundreds of billions of euros in losses annually [22, 57, 63], there are still severe deficiencies in the proper application of software cost estimation methodologies in organisations [2, 38, 49, 52, 55, 72].

Systematic overruns have continued for decades, although researchers and practitioners have developed hundreds of estimation methodologies [39, 55]. However, the reason for the overruns may not reside only in the estimation methodologies as they are shown to be able to produce accurate results when used properly [69, 70]. Thus, the problems that result in estimation errors may occur because estimation methodologies are used ineffectively by organisations [39, 49, 72]. Consequently, organisational inhibitors [52], top management focus [71] and the sources of distortions [2, 49] have become the focus of recent studies.

While most SCE does not use a proper methodology, the situation is considerably better in the area of project management (PM) as, according to Fortune and White [85], only 5% of projects do not use any PM tools. Considering the fact that cost estimation is an inseparable part of all projects [30], and that the cause of overruns in software projects may reside in software cost estimation (SCE), project management (PM) or other areas [11, 56, 61], the difference in the extent of the use of methodologies between software project management and management of other types of projects is surprising. Especially, because commonly used industrial project management and process improvement frameworks, such as CMMI [81], PMBOK [30] and IPMA ICB [32], promote the importance of estimation and the use of methodologies. The use of proper methodologies is proven to have a positive effect on the outcome of both SCE and PM [15, 85, 91], nevertheless only PM professionals utilise these valuable tools and methods to any great extent.

As scientific literature or industrial advice does not provide a clear explanation for the gap in the extent of the use of methodologies between SCE and PM, one assumption is that the difference arises from organisational priorities and does not seem to be related to the availability of proven cost estimation methodologies. Project management is widely linked to the execution of the corporate strategy [17, 48, 79], but SCE seems to have very little visibility among top management. Also, while project management research has paid close attention to non-technical factors, like top management support, communication, skills and learning [42, 85], SCE research has mostly focused on developing and improving estimation techniques [39]. This is an important observation, indicating that the explanation for the difference in the extent of use of SCE and PM methodologies could reside within the research areas omitted from the study of SCE.

The goal of this study is to identify organisational factors that either support or hinder meaningful SCE, and to establish their impact on estimation success. The study takes a holistic view with specific attention to top management participation. A qualitative, exploratory case study approach is employed, using interviews as the primary data collection method. In total, three projects were studied and 18 semi-structured interviews conducted.

Some research papers addressing SCE from the organisational rather than technical viewpoint have been published recently [e.g. 18, 49, 52, 71]. This paper continues on this highly relevant path but diverges from previous studies by studying the impact of organisational factors related to software process or project process on the effectiveness of the use of estimation methodologies. Improving the understanding of the real-world dynamics related to the effective use of estimation methodologies may provide practitioners with valuable tools for improving SCE in organisations. Especially, the gap between the advice provided by the industrial project management frameworks and the low extent of use of methodologies could be narrowed. This study may also provide further evidence that organisational issues are equally important as technical ones for effective SCE, and generate new theories about the reasons for why the extent of use of methodologies is low regardless of the experienced importance of SCE and industrial advise. This would justify further study on the organisational dimension of SCE.

The remaining part of the paper is structured as follows: Section 2 presents related work focusing on four areas: Software cost estimation, Project management, Top management involvement and Software cost estimation in industrial frameworks. Section 3 presents the research questions. Section 4 introduces the case companies and projects, and Section 5 elaborates on the research design. Section 6 presents the results of the case study and is followed by a discussion of the key findings in Section 7. Section 8 concludes the study.

2. Related work

In the following subsections, we will review top management's relationship to SCE and PM and summarise the focus areas of earlier research on these subjects.

2.1. Software cost estimation

Software cost estimation is an activity that aims to produce a prediction of the effort required to build a software component. As most costs in software development projects are personnel costs, 'cost' and 'effort' are often used interchangeably. The literature that studies and develops methods to estimate costs in software projects began in the 1960s [60, 62]. However, despite five decades of research and hundreds of studies [10, 39], software projects still exceed their budgets and timetables.

Jørgensen and Shepperd [39] conducted the most recent systematic literature review of SCE. In total, they selected 304 journal articles for their study and identified eight active research topics in SCE:

Estimation methods: the key issues include formal estimation models, expert estimation processes, decomposition based estimation processes and combinations of those three.

Production function: the key issues are the linear versus nonlinear relationship between effort and size, and the relationship between effort and schedule compression.

Calibration of models: the key issue is the calibration of estimation models, e.g. studies on local versus multi-organisational data and the calibration of the COCOMO model for certain types of projects.

Size measures: the key issues include validity and improvements in the size measures that are important in estimation models, e.g. the inter-rater validity of function point counting.

Organisational issues: the key issues are estimation processes in a wide organisational context, e.g. estimation practice, the reasons for cost overruns, the impact of estimates on project work, and estimation in the general context of project management.

Effort uncertainty assessment: the key issue is the uncertainty of effort or size estimates, e.g. methods providing minimum-maximum intervals for effort.

Measures of estimation performance: the key issues include the evaluation and selection of estimation methods, e.g. how to measure estimation accuracy or how to compare estimation methods.

Data set properties: the key issue is how to analyse data sets for the purpose of estimation methods, e.g. data sets with missing data.

Other: unclassified topics.

The distribution of the topics is presented in Table 1.

As shown in Table 1, all other categories except 'Organisational issues' and 'Other' focus on estimation methodologies or other formal methods for improving the estimation of size, effort or schedule. Only 16% of the articles discussed issues other than non-technical issues, i.e. organisational issues. Thus, SCE research strongly focuses on formal and technical issues and has relatively little focus on non-technical topics. Furthermore, the share of the articles focusing on organisational issues seems to be decreasing, having been only 14% during the period from 2000 to 2004. The recent study of SCE research trends shows also that the research focus has remained consistently on estimation methodologies and techniques between 1996 and 2016, the emerged research areas being 'size metrics', 'estimation by analogy', 'tools for estimation', 'soft computing techniques' and 'expert judgement' in five topic solution [77].

Table 1. Distribution of research topics in software cost estimation. A single study can belong to multiple categories. Adapted from [39].

Perspective	-1989	1990-1999	2000-2004	Total
Estimation method	73 %	59 %	58 %	61 %
Size measures	12 %	24 %	16 %	20 %
Organisational issues	22 %	15 %	14 %	16 %
Uncertainty assessment	5 %	6 %	13 %	8 %
Calibration of models	7 %	8 %	4 %	7 %
Production function	20 %	4 %	3 %	6 %
Measures of estimation performance	5 %	5 %	6 %	5 %
Data set properties	0 %	1 %	2 %	1 %
Other	0 %	2 %	1 %	1 %

Estimation methodologies produce good results when applied properly [69, 70]. Regardless of this, overruns still continue. While an obvious research topic should be the effective application of estimation methodologies, 84% of the articles still focus on improving methodologies. Hihn and Habib-agahi noticed already in 1991 that only 17% of the estimators used proper estimation methodologies [28]. This, however, seems not to have affected the research focus either. Also according to our experiences, the basic problem of SCE is that the estimation methodologies are not applied properly; researchers and practitioners largely agree on this point [39, 55]. Furthermore, Jørgensen and Shepperd's [39] review reports that only eight articles out of 304 were in-depth case studies and only three evaluated the background to the estimation processes. This, together with the technical focus of the research, confirms that concentrating on real-world issues that prevent the effective use of SCE methods is justified as a systematic improvement in SCE success that can only be realised through the successful application of estimation methods in real-world situations.

2.2. Project management

The share of work organised as projects is very high in organisations, and the results of such projects are critical for the success of an organisation [12, 84]. Due to the significance of PM, the topic has been broadly studied and the body of knowledge on it is extensive. Several different categorisations of PM research areas exist and the following six perspectives have been presented by Kolltveit, Karlsen and Grønhaug [42]:

The task perspective: key issues include the scope of project management for a task, project targets, project results and planning and control.

The leadership perspective: key issues are leadership, communication, uncertainty and learning

The system perspective: key issues are systems, elements of systems, boundaries and dynamics.

The stakeholder perspective: key issues include stakeholders, communication, negotiation, relationships, influence and dependence.

The transaction cost perspective: key issues are transactions, transaction costs, production costs, and governance structure.

The business by project perspective: key issues include business, project results, project success, strategy, profit and benefits.

In their article, Kolltveit et al. [42] identified 562 articles published in *International Journal of Project Management* and classified them into the six above mentioned categories (see Table 2).

Once again, when dividing the areas or aspects into technical and non-technical, the task and transaction cost perspectives can be seen as technical. The other four can be seen as non-technical,

Table 2. The distribution of research perspectives in project management. A single study can belong to multiple categories. Adapted from [42].

Perspective	1983-1987	1988-1992	1993-1997	1998-2002	2003-2004	Total
Task	49%	34%	32%	23%	12%	29%
Leadership	8%	16%	25%	28%	33%	23%
System	23%	25%	18%	19%	15%	20%
Stakeholder	1%	3%	1%	5%	6%	3%
Transaction	19%	9%	6%	10%	6%	10%
Business	0%	13%	17%	15%	29%	15%

or at least having most of their key issues beyond the purely technical focus. As Table 2 shows, the focus of the project management research has been shifting from the task perspective towards the leadership and business perspectives. This can be seen from the table as with the above classification into technical and non-technical aspects, the share of technical perspectives decreased from 68% to 18% between the first and the last period, respectively. This shift of focus seems reasonable since organisational issues are reported to be even more important factors in project success than technical ones [21, 47, 64, 91]. Top management support (TMS) has even been suggested as the most important factor affecting project success [89], which corresponds well with the largest share of the leadership perspective related papers.

In comparison to SCE research, PM research has undergone a major shift from task oriented or technical topics towards people oriented or non-technical ones, whereas the SCE research focus remains on task oriented subjects. Thus, it is also reasonable to assume that the focus of PM research has placed more focus on how methods are applied by people and therefore increased the awareness, effectiveness and extent of use of the methods. The mere existence of a method seldom leads to its success.

2.3. Top management focus

Top management support has been found to be one of the most important critical success factors for project success in several studies [21, 76, 89] and few would doubt the need for TMS [54]. Also, top management's interest in PM is increasing along with the number of PM related articles published in top management and business journals [43]. However, top managers are generally more interested in non-technical issues of a strategic nature [16, 83].

The practices through which TMS is demonstrated for a project have been extensively studied. Garrity [23] recommends top management review plans and monitor results. Beath [6] has found that top managers are able to make organisational changes, while Morton [59] notes top managers—as project champions—have the skills to mobilise public opinion, resolve conflicts between stakeholders and win the hearts and minds of project teams. Zwikael [91] has identified a list of 10 critical top management support processes that influence a project's success, including appropriate PM assignment, project manager involvement during the initiation stage and the use of standard PM software.

TMS has not been studied widely in the scope of SCE. However, Rahikkala et al. [71] found that top management pays attention to SCE and recognises that good estimates are critical for an organisation's success, as well as for understanding the consequences of an erroneous estimate. In general, there is very little information about TMS in SCE. This suggests that the actual top management focus on SCE is low. Regardless of the reported attention, the limited use of SCE methodologies supports this assumption.

2.4. Software cost estimation in industrial frameworks

Many of the commonly used project management frameworks, standards and other related guidelines address cost estimation. Project Management Institute's PMBOK Guide [30], as well as its Software Extension [31], give detailed guidance on preparing software cost estimates. Another popular framework, International Project Management Association's Competence Baseline [32], includes cost estimation as an important step. Furthermore, PRINCE2[1] and ITIL v3 [1] frameworks emphasize estimation and cost management, as well as the CMMI process improvement program [13] and the ISO 21500:2012 standard for project management [33]. Even the U.S. Government Accountability Office (GAO) has published a 12 step guide for cost estimation[2]. Finally, cost estimation is also covered by agile methodologies [14].

3. Research questions

The literature review shows that SCE research has been centred on methodology for decades without a significant change. In contrast, PM research is very broad and covers topics like methodologies, leadership and business. The focus of research has also shifted from methodologies towards other areas, currently having a relatively even distribution on a broad range of topics. In particular, TMS has been studied in the scope of PM but not SCE. Hence, though SCE and PM belong to software project delivery, the research focus is different. In the industrial context, the importance of SCE is widely recognized, and practically all major industrial bodies of knowledge provide guidance for cost estimation.

The above, together with the argument that proper cost estimation is often omitted [28, 52], suggests that the accountability of the use of meaningful estimation methodologies is unclear in organisations. There are no reports that SCE would be commonly omitted completely, rather that it is not conducted in a meaningful way. The previously reviewed project management and process improvement frameworks define clearly that project management is responsible for that the estimation is done, but not specifically that they would be responsible for how it is done. This seems to leave a gap in the software process, which may be one reason for malpractices and overruns. This motivates our first initial objective:

RQ1 What are the real-world factors concerning the organisational context of SCE (organisational factors) that either support or hinder the creation of a meaningful software cost estimate?

In our study, the organisational context refers widely to properties and mechanisms of an organisation, such as top management commitment, leadership, organisational structure, communication, monitoring, recognition and education [68]. Effectively, the definition of the organisational context used in this study does not exclude any properties or mechanisms of an organisation, and we seek to identify the aspects affecting SCE that human subjects can or are willing to tell us about the topic [50]. Additionally, although the organisational context is our primary focus, we also consider biases emerging from human behaviour, as human subjects are centric for the organisational context.

It has been found that technical issues are of little interest to senior managers [16, 83]. One reason for the existence of the previously described gap may be that SCE is perhaps perceived as too technical and too specific to software development to interest project managers. On the other hand, although software developers traditionally focus on technical topics and have little interest in or power over non-technical issues, they may not perceive SCE as a technical issue, and

[1] https://www.axelos.com/qualifications/prince2-qualifications
[2] http://energy.gov/sites/prod/files/GAO%2012-Step%20Estimating%20Process.pdf

Table 3. Case study companies and projects.

Company	Software Vendor	Service Provider	Tech Giant
Number of employees	Approx. 150	Several thousands	Several thousands
Business area	Software and services	Software and services	Software and services
Project	Tool	Operational Control System	Network Management System
Initial / actual size of the project	12 / 44 person-months	20 / 20 person-months	Approx. 200 / 200 person-months
Initial / actual duration of the project	3 / 11 months	10 / 10 months	3 / 3 months
Project type	Internal product development	External product development, i.e. tailored software	Continuous internal product development
Estimation methodology	WBS and expert estimation	WBS and expert estimation, historical data, peer review	WBS and expert estimation, historical data
Estimation responsible	Project Manager	Project Manager	Program Manager
Development methodology	Scrumbut: Waterfall (design) + Scrum (sprints)	Waterfall-like method	Scrum
Result	Challenged	Successful	Successful

consider it to belong under project management's domain. Technical experts may also be protective of their domain in order to prevent loss of power to outsiders [78], while the suspicious and negative attitudes of senior managers towards IT and technical personnel [88] may hinder cooperation further. Therefore, the second initial objective of this study is to answer the second research question:

RQ2 What is the impact of top management in either supporting or hindering software cost estimation practices?

Finally, this paper draws attention to the difference between the extent of use of SCE and PM methodologies, as well as to the different focus areas of research on SCE and PM. Additionally, the gap between the extensive amount of industrial advice on cost estimation and the low extent of use of SCE methodologies is addressed. An enhanced understanding of the reasons behind these differences may help organisations improve their SCE success, positively affecting project success.

4. Case contexts

The topics covered in this paper have not been widely addressed prior to this study and our goal was to collect widely different perspectives related to the organisational phenomena affecting SCE, and especially top management's role. Thus, we selected the cases so that they would generate rich information about the phenomena being studied. We focused on large and small companies, selecting higher and lower maturity organisations and exemplary and challenged projects. The case companies and projects are different in their industrial domains, size, as well as in their processes. The final decision of including a particular project in the study was made based on a discussion with a company representative, confirming that the project was likely to add new perspectives in the study. Table 3 depicts the characteristics of the case study companies and the projects. The companies wished to remain anonymous.

4.1. Case 1 — Software Vendor's Tool project

Software Vendor is a software producing company of about one hundred and fifty people. Its main line of business consists of selling consultancy and support services as well as software products to businesses. The company is global and has offices in several countries. We studied Software Vendor's Tool project, which aimed to produce an application development tool.

While the overall project was strictly planned beforehand, the actual development work was divided into sprints. The development work started with a prototype version in which technical challenges were studied. The Product Owner and Project Manager were named to the project already in the prototype phase. The Product Owner was responsible for creating a design document for the product, whereas the Project Manager, based on the design document, was responsible for crafting a timetable and cost estimates. Initially, the project was designed to take three months with a team of four people. Based on the estimate and design document, top management approved and started the project.

The Tool project overran its schedule and budget by over 200%. However, the project delivered the planned scope and the Senior Business Manager reports that the outcome of the project met his expectations and he attributes the overruns to estimation error and project performance related issues.

4.2. Case 2 — Service Provider's Operational Control System project

Service Provider is a large software producing company with thousands of employees, providing tailor-made and package software and consultancy services for businesses in various sectors. The company has premises in several countries. We studied an Operational Control System project by Service Provider that aims to produce custom software for a long-term customer. The Operational Control System is used for reporting and analysing process control data.

The project followed a Waterfall-like software development process. The first stage of the project was requirement elicitation and analysis. After the specification was approved, the project was estimated. The estimation was made by the developers and testers, led by the project manager, who had the overall responsibility of the cost estimate. The estimate was a result of expert estimates, placed into a software tool specifically tailored for the application area.

The project was planned according to certain restrictions: the budget and the timetable was fixed. The development started when the customer and the vendor had agreed upon the scope. There was a small number of unknown features that needed further elaboration. The development work continued straightforwardly from design through implementation and testing to delivery. The duration and effort of the project was 10 months and 600 man-days, respectively. Regardless of a significant rescoping during the project, it concluded under budget and on schedule with good customer satisfaction.

4.3. Case 3 — Tech Giant's Network Management System project

Tech Giant is a large company selling products with software to global business-to-business markets. The company has tens of thousands of employees around the world. We studied the Network Management System project of Tech Giant. The project produced a new release of a tool for managing the network. The Network Management System has been in use for several years.

The project was a part of a continuous development cycle involving just under 100 people. A new release of the system is developed every three months. The development methodology it used was based on Scrum with two week sprints. The development teams were distributed over several locations. The cost estimation was conducted in two phases: firstly, rough planning for the whole three month release in the product management function. Secondly, the backlog items were estimated in the Scrum teams, the main responsible being the program manager. The estimate for the whole release was based on historical data about certain parts and the estimates for those parts were prepared by requirement engineers. The backlog items were estimated by using an expert estimation. The project concluded successfully and delivered over 85% of the planned scope, which is the goal for all releases.

5. Case study design

The question of how the organisational phenomena (RQ1) and specifically the actions of top management (RQ2) affect SCE are investigated through three case studies. Since this study deals with contemporary phenomena in a real-world context—over which the researcher has little or no control—case studies were chosen as a suitable research approach [87]. This study is exploratory, discovering what is happening, seeking new ideas and generating hypotheses and research areas [74]. The research uses a multiple case study design and replication logic [87]. The richness of the information is maximised by using both exemplary and average organisations as cases [66]. The unit of analysis is a single software cost estimate. The study focuses on the experiences gained during the preparation of the cost estimate and the related software process.

To facilitate the identification of organisational phenomena, we have decided to utilise the concept of maturity. Software process maturity is the extent to which a specific process is explicitly defined, managed, measured, controlled and effective [67]. Paulk et al. [67] argue that maturity implies the potential for growth in capability and indicates both the richness of an organisation's process and the consistency with which it is applied in projects. Furthermore, mature organisations provide training for processes and the processes are monitored and improved. In general, the concept of maturity measures organisational capability, culture and consistency in a holistic way, thus it can be expected to usefully facilitate the discovery of organisational phenomena. Thus the maturity of SCE and software processes are assessed for this study.

5.1. Instrumentation of SCE maturity

To assess the maturity level of SCE in an organisation, we have developed a definition of an ideal SCE procedure, covering its most important aspects as identified in [55]:

1. The use of an estimation methodology: A clearly defined, established estimation methodology is used to produce the estimate, instead of making presumptions.
2. Proper communication of the estimate: The assumptions, accuracy and intended use of an estimate are communicated as part of the estimate, instead of being presented as a figure lacking further explanation.
3. Planned re-estimation: An estimate is improved systematically when information about the assumptions behind an estimate is increased and updated after the initial estimate.
4. The use of a documented estimation procedure: A documented procedure for producing and communicating an estimate is followed, instead of an ad-hoc procedure.

If the above-mentioned areas of SCE are properly covered, the estimation process should avoid many of the worst pitfalls and the outcome will have a fair chance of being useful for project control. As demonstrated by Lederer and Prasad [44], using guessing or intuition as an estimation methodology is connected to budget and schedule overruns. Also, the accuracy of an estimate increases as a project progresses [8, 46], which encourages the re-estimation and good communication of an estimate. In addition, one poorly estimated aspect can become an *anchor* and may contaminate a whole project's estimate [4, 40]. Furthermore, a documented estimation procedure protects organisations from poor estimation practices and promotes good practices [55]. Standardised procedures have also been found to improve the results in PM [30, 58], specifically in software development [34, 69]. Thus, if an estimate is the result of a rigorous procedure covering the above mentioned aspects, it is more likely to be useful.

5.2. Instrumentation of process maturity

In order to ensure that the relevant phenomena are discovered, we will also extend the scope of our investigation outside the actual SCE and assess the maturity of the software processes in the studied organisations by using the Capability Maturity Model (CMM) [67]. The CMM establishes a set of publicly available criteria describing the characteristics of mature organisations. CMM presents the process maturity of an organisation in a scale from 1 (low maturity) to 5 (high maturity). For the CMM assessment we use the general characterisations of maturity levels presented by Paulk et al. [67, pp. 9–14] as well as key software process area goals [67, pp. 59–64]. Together, the CMM characteristics and goals cover a wide range of process areas, so it is probable that reviewing these items will facilitate the discovery of organisational factors affecting SCE, helping us to answer RQ1 and RQ2. While CMM is rather old, it still describes well the relevant properties and mechanisms of an organisation, making it a relevant tool for discovering phenomena in the organisational context.

Higher maturity organisations have been found to perform better in software development [24, 25]. The maturity assessment is also related to process areas rather than to techniques, to what rather than to how, making it agnostic to any specific development methodology. Therefore, the software development and estimation maturities are relevant to the discussion of organisational phenomena. The CMM is also specifically intended to be used for software process assessment and software capability evaluations [67].

The CMM evaluation for the case study companies was made by the researchers during the interviews and documentation review. We would like to point out that we followed good auditing practices and the main author had over five years of experience of auditing and holds an ISO 9001:2008 Lead Auditor certificate. Therefore, we believe that the CMM requirements conformance evaluations conducted as part of the research are valid and we gained a good overall understanding of an organisation's CMM level, even though our focus was still primarily on SCE. We focused on SCE related topics and CMM acted only as a facilitating instrument.

5.3. Subject selection

The subject sampling strategy was to interview the management and representatives of other roles related to the case projects. In total 15 people were interviewed in 18 interviews (key informants were interviewed twice), as presented in Table 4. All participants attended interviews voluntarily and anonymously and the collected data is treated confidentially.

5.4. Data collection procedures

The data for this study was collected within seven weeks. The primary data collection methods were semi-structured interviews [74] and a review of documentation. In total 15 people were interviewed and 18 documents reviewed. The documents included typical project documentation, such as cost estimates, project plans, meeting minutes and status reports, to gain a better understanding of

Table 4. Interviewees and their role in the projects

Software Vendor	Service Provider	Tech Giant
Product Owner (key informant)	Project Manager (key informant)	Program Manager (key informant)
Senior Business Manager	Business Manager	Line Manager
Senior Technology Manager	Testing Manager	Senior Manager
Project Manager	Requirements Engineer	Requirements Engineer
	Software Developer	Head of Product Management
		Head of Programs

the procedures and SCE methods used. The case studies were completed one at a time to allow the reflection and refinement of the research and interview questions [41]. All the interviews (but not key informant interviews) related to a single case study were conducted on the same day, with the exception of one interview for the last case study. Each interview lasted approximately one hour. Each interview day was preceded by a key informant interview day during which background information about the case was collected from a person in a central role in the case study area. The key informant interviews addressed the following topics:

1. Project background, size, status and success
2. Project team members and their roles
3. Estimation methodology and success
4. Software development methodology
5. Software process maturity, capabilities and track record

The semi-structured interviews were based on a predefined list of questions. Any interesting facts and observations that were mentioned led to additional questions being asked on that subject. The interview instrument was developed by three researchers and adapted slightly for the individual case studies. All the interviews were conducted by two researchers, who interviewed one subject at a time. The interview instrument is provided in Appendix A, and it consists of the following main areas:

1. Introduction
2. Personal, team and project background
3. Current state of SCE in the organisation
4. Experiences of the organisational phenomena affecting SCE
5. Ending (uncovered topics)

5.5. Data analysis procedures

The primary steps for deriving conclusions from the experiences of the study subjects included 1) semi-structured interviews, which were sound recorded, 2) collection of documentation, 3) transcription of the interviews, 4) the coding of transcripts and documents, 5) grouping the coded pieces of text, and 6) making conclusions. The NVivo 10 application was used for aiding the process, and special care was taken to maintain a clear chain of evidence. The overall process of analysis was conducted as outlined by [75].

During the coding phase, each interview transcript and collected document was reviewed statement by statement, and statements containing information about organisational factors (RQ1) or top management participation (RQ2) were assigned a code representing the findings category. After that, readily coded main categories were reviewed statement by statement to identify subcategories. Subcategories were also identified from the original transcripts. After a couple of iterations, subcategories emerged from these two approaches. The performed analysis was of the inductive type, meaning that the patterns and categories of the analysis come from the data, instead of being pre-defined. Themes that were often raised in the interviews were identified and coded. The application used for coding (NVivo 10) maintained the evidence trail from the coded pieces of text back to the documents, transcripts and interviewees automatically. The coding of the texts was primarily conducted by one of the researchers. Another researcher conducted a shorter coding of the data, with fewer iterations, independently, to validate the results of the coding. Any differences were discussed and resolved, and the categorisation was refined. The final categorisation formed a structure for reporting the findings of the study.

After the coding of the data, the coded statements were grouped together to form initial hypothesis, or candidates, for conclusions. The process progressed iteratively, and was, once again, conducted primarily by one of the researchers, while another researcher conducted an independent analysis with fewer iterations to validate and refine the results. After a certain number of iterations, and until the end of analysis, the analysis of the statements was conducted by two researchers together. The other two researchers reviewed and validated the results. During the process of forming a hypothesis, interviewees were asked clarifying or additional questions, where deemed necessary, to resolve any unclarities and to provide additional confidence for the hypothesis. The traceability was secured by marking all statements used for forming the hypothesis with identification codes, enabling back tracing to the coded statements.

In addition to the interview data and documentation, the researchers' memos written during the interviews were used as information sources and as part of the data analysis. The collected project documentation provided mostly background data for the case projects, and to some extent, information regarding top management's participation in different phases of the projects. From the organisational context point of view, the documentation provided some information about the software process and related decision making. The role of the collected documentation was mostly to provide background information and to support statements made by the interviewees.

5.6. Validity procedures

The qualitative case study methodology involves the researchers themselves as the instrument of the research, which poses a risk that the results are biased by the researchers' subjective opinions. More generally speaking, Robson [74] has identified three types of threats to validity: reactivity, researcher bias and respondent bias. Reactivity means that the presence of the researcher may influence the study, and particularly the behaviour of the study objects. Researcher bias refers to the preconceptions of the researcher, which may influence how questions are asked and answers are interpreted. Finally, respondent bias originates from the respondents' attitudes towards the research, which may lead, for example, to withholding information or giving answers the respondents think the researcher is looking for.

Because of the researcher related threat to validity, a discussion of the effects of the involvement of particular researchers is appropriate [74]. The main author of this article has been involved in professional software development since 1996, including companies from start-ups to international giant corporations. Additionally, he has been conducting academic research within the area of SCE since 2012, holds an ISO 9001:2008 Lead Auditor certificate, and has over seven years of experience of quality management system audits. The other authors are from academia, having their main focus in software process, software development methodologies and software economy. Together they have published hundreds of research papers, and used different methodologies extensively in their research, including qualitative case studies.

The reactivity, researcher bias and respondent bias threats to the validity of the study were addressed through six strategies provided by [74]: prolonged involvement, triangulation, peer debriefing, member checking, negative case analysis and audit trail. The summary of the taken countermeasures to negate the validity threats are summarised below:

Prolonged involvement While the study observations were completed during a short period of time, all the researchers had followed the case study companies for at least two years and were intimately aware of recent developments in the software development methodologies being used. All case organisations had participated in a national research programme, Need4Speed (www.n4s.fi), enabling the confidential sharing of information between the organisations and the researchers.

Data source triangulation Multiple data sources were used, including interviews with persons in different roles, project documentation and informal observations.

Observer triangulation Interviews were conducted by two researchers together. This also reduced the strain caused by conducting up to six interviews during one day. Additionally, the interviewees had a short break before each interview, and a longer break in the middle of the day. Important analysis steps were conducted by two researchers independently, and emerging issues were discussed and refined.

Methodological triangulation The data analysis included qualitative interviews and the analysis of project documentation.

Theory triangulation Several perspectives were considered for interpreting the results, including the perspectives of the subjects, researchers and other peer group members.

Peer debriefing Peers, including practitioners and researchers, reviewed the research in different research phases. One research paper based on the conducted research has already been published [71]. The results of this research have been reviewed by the Need4Speed research programme steering group.

Member checking Interviewees reviewed both transcripts and analysis, providing feedback and commentary.

Negative case analysis Elements that seemed to contradict the conclusions of the analysis were identified and alternative explanations discussed.

Audit trail Strict scrutiny was practiced to maintain a clear audit trail from data collection to the final conclusions. All interviews, transcripts, codings and other analysis are archived.

Considering that our study is based on three projects, exploratory of nature, and that the study topic has not been widely explored prior to this study, *generalizability* of results is low. However, our study consists of three case companies and 15 interviewees with different roles, and it provides in-depth findings and detailed information of the study itself. Thus, *transferability* of the study should be fair, although case studies are always coloured by their specific context.

6. Results

In the following sections we will present the findings related to organisational phenomena (RQ1) and top management actions (RQ2) affecting SCE. The findings are divided into four main categories (the role of management, communication, process maturity and attitudes) that were found in the analysis and classification of the results by the authors. Additionally, the main categories are divided into subsections as appropriate. The main observations related to the second research question are located in Section 6.1 whereas the sections 6.2–6.4 contribute the first research question.

6.1. Management role

Findings related to the management's role are presented in the following sections. Table 5 summarises the findings.

6.1.1. Estimate visibility and purpose

In Case 1, the Tool project, Senior Business Manager studied the project plan containing the estimate considering the strategic importance of the project to the company. In Case 2, the Operational Control System project, the business manager responsible for the important customer relationship reviewed the estimate. Practically, the visibility of the estimate correlated with the ownership of the project and the daily involvement of the managers with the project domain. There was no visibility of the estimate beyond the review as the project was no longer part of the manager's daily responsibilities. In Case 3, the Network Management System project, the most senior manager aware of the estimate was the manager of the whole product family. There are roughly 1,000 experts involved in the system development, so the estimate was visible to relatively senior managers.

In Case 2, the estimate was used for preparing an offer for a customer and planning the project, while in Case 1 and Case 3, the managers reported that they needed the estimate to ensure that the resources, scope and schedule were in balance with each other. In Case 1, the Senior Business Manager reported that the estimate was needed to ensure the project scope was the minimum viable and that the project would deliver the results as soon as possible.

6.1.2. Participation in estimation

None of the managers studied the estimate in detail. In Case 1, the Senior Business Manager reviewed the estimate only as part of the project plan. In Case 2 and Case 3, the managers reviewed the estimates on a summary level. None of the managers participated in the estimation work, and the managers in Case 1 and Case 3 were not aware of the estimation practices. In Case 2, the manager was aware of the practices because cooperation with the customer was said to be very intense; the customer wanted to discuss processes related to daily cooperation. While the managers were not involved in estimation on a practical level, the managers in cases 2 and 3 stated that they challenged the estimate when necessary. Also, in these two cases, the Project Manager and Product Owner, respectively, scrutinized the estimate. An awareness of such scrutinizing allowed the managers to have greater trust in the estimate. That is, there was no need for them to personally study the estimate in detail.

6.1.3. Resource provisioning

In Case 1 and Case 2, the Tool and Operational Control System projects, the estimators reported that they had enough time to prepare the estimates. In Case 3, the Network Management System project, the estimators wished to have more time. However, although the estimation work was very time consuming and complex, when considering the previous good results, the time reserved for estimation seems to have been reasonable. The perceived lack of time was connected to the complexity and size of the estimation domain. Also, an estimator in Case 3 wondered whether additional time would actually improve the estimates. In Case 1 and Case 3, building prototypes was also used as a method for acquiring additional information to use for estimation, which supported the idea that management provided adequate resources for the estimation work.

Table 5. Summary of management role findings.

	Case 1	Case 2	Case 3
Company	Software Vendor	Service Provider	Tech Giant
Project	Tool	Operational Control System	Network Management System
Estimate purpose	Ensuring the resources, scope and schedule balance, ensuring the minimum viable scope and fast delivery	Preparing an offer for a customer	Ensuring the resources, scope and schedule balance
Participation in estimation	The project plan containing the estimate studied on a summary level, management not aware of the estimation practices	The estimate reviewed on a summary level, management aware of the estimation practices, the project manager scrutinized the estimate	The estimate reviewed on a summary level, management not aware of the estimation practices, the product owner scrutinized the estimate
Resource provisioning	Estimators had enough time for preparing the estimate, prototypes used for supporting estimation	Estimators had enough time for preparing the estimate	Estimators wished to have more time, prototypes used for supporting estimation
Demonstrated importance	Estimates considered as important, confirmed by interviewees	Estimates considered as important, confirmed by interviewees, importance linked to customer promises	Estimates considered as important, confirmed by interviewees, importance linked to customer promises
Goal setting	Goals perceived as realistic, realism pursued, no support for realism from historical data, clear expectations of the scope and schedule, pressure to fit the estimate to expectations	Goals perceived as realistic, realism pursued, hundreds of annually delivered projects supported realism	Goals perceived as realistic, realism pursued, four annual releases for the same product supported realism
Other	No shared project vision		

6.1.4. Demonstrated importance

In all cases the projects had strong support from management, and the managers emphasized the importance of the estimates. In Case 2 and Case 3, the estimate was strongly linked to keeping the promises given to customers. All the interviewees concurred that management considered the estimates to be of high importance.

6.1.5. Goal setting

All interviewees reported that the project goals seemed realistic and achievable at the beginning of the project, and that everybody pursued realistic estimates. In Case 2, Service Provider delivers hundreds of projects yearly, while in Case 3, Network Management System has four releases per year, thus its management is likely to have a realistic picture of its organisational performance. This probably also supports the setting of realistic and achievable goals for releases and projects. In Case 1, the Tool project was using a new development methodology for the first time, meaning relevant historical data about the process performance was lacking and goal setting was unsupported.

In Case 1, Senior Business Manager had expressed the strategic importance of the project, which he had initiated personally, prior to the estimation. Also a roadmap vision, which presented a release date, had been communicated for the product. Furthermore, the scope of the project was considered to be the minimum viable, meaning that the scope could not be reduced. As a result, the estimator was facing a situation in which both the scope and schedule were effectively set, which is always a challenging situation from project planning point of view. The estimator describes having perceived pressure to fit the estimate to these expectations and having started to doubt the estimates when they did not match initial expectations. Case 1, the Tool project, thus seems to have experienced the anchoring phenomena [4, 40], i.e. the estimate is affected by an expressed starting point. However, Senior Business Manager of Case 1 points out that flexibility in resources and schedule was emphasised prior to estimation.

6.1.6. Provided direction

The interviewees in Case 1 report that there were different expectations for its outcome: Senior Business Manager expected a strong commercial product, while others were building a pre-version, which would contain the full scope of features but not on the quality level expected of a commercial product. The expectation of the rest of the team was that the quality issue would have to be addressed in the next version of the product. This difference in the expectations was probably a significant source of estimation error. Actions for error detection and customer feedback collection add to the amount of work required, as do fixing bugs and improving functionalities based on customer feedback.

6.2. Communication

The role of the written documents, as required by the processes, was significant in Case 1 and Case 2, which followed Waterfall-like development methods. The projects had significantly invested in preparing the documents on which the estimates were heavily reliant. Interviewees from both projects reported that the documents were detailed and of high quality. Also the Network Management System team in Case 3 used documentation as part of its estimation but—as is typical of agile development—it did not have an official role. The documents were prepared on demand when necessary, including pre-studies, memos, presentations and user stories. In addition to the documents, Software Vendor in Case 1 had developed a prototype to get more information on the application area. Prototypes are artefacts, which are likely to support successful estimation because they contain significant amounts of relevant information on the estimated application area and answer many questions relevant to estimation [5]. Tech Giant in Case 3 also reports that it occasionally uses prototypes, while the Business Manager from Service Provider adds that prototypes would be useful but are not utilised at the moment.

While the interviewees recognised the importance of the written documents, all the interviewees in Case 2 and Case 3 emphasised that the process of preparing an estimate is more important than the result itself. The Requirements Engineer and the Project Manager in Case 3 describe the importance of mutual understanding, and all reported that truly understanding each other's needs is crucial. The Requirements Engineer pointed out that estimates become ever more reliable through discussions and said that he is satisfied when all the questions are answered. The Requirements Engineer also highlighted the fact that working together provides confidence in each other. Group estimation sessions were used regularly in both Case 2 and Case 3. The Senior Manager at Case 3 concluded that a good estimate is based on good skills in preparing the specifications and having a broad knowledge about the application area and software development — the majority of the

Network Management System project team members in Case 3 had worked on the product for five or more years. Communication seems to be central to estimation in Case 3 because issues like multiple locations and time zones hindering estimation were mentioned. Agile grooming was also mentioned as an important forum for estimation and related communication.

In Case 2, the Project Manager and Testing Manager reported that good cooperation and fact based communication with customers supported estimation. They also emphasised the role of feedback. The interviewees at Case 2 described team members as competent in their area of expertise, stating that estimates were prepared together to a large extent. The Testing Manager added that the atmosphere was open in general. Peer estimation was used on both the programming and PM level. The Project Manager stated that being able to receive consultation or a peer review from another project manager is more important than using information systems to support estimation. The Business Manager added that the project's estimation succeeded because they understood the customer's needs. The Software Developer expanded on that by saying the estimation succeeded because all the details relevant to the case were found. The Testing Manager described an estimation as meaningful if the right experts were consulted and involved in discussions.

In Case 1, the communication relied more on the documentation. The project manager who prepared the estimate described it as being stored on a shared folder, although no feedback was received. The estimate was based on a design document, which was prepared by the Product Owner. The Project Manager revealed that there had been some discussions with the Product Owner to scope down certain features but the Product Owner and the Senior Technical Manager reported that the estimate had not been challenged at any phase. However, they both stated that they had been sceptical about the estimate but could not point out exactly where the problems resided, and therefore did not raise their reservations. In general, the interviewees reported very few occasions when the estimate would have been discussed. The communication relied mostly on documents prepared by individuals. However, the Senior Technical Manager and Product Owner reported that the atmosphere was open and there was no pressure not to discuss a topic.

6.3. Process maturity

6.3.1. Estimation maturity

All of the case study companies had a documented software process describing how estimation was related to the whole and which documents were required, but only Service Provider in Case 2 had a written procedure for the estimation itself. However, Tech Giant in Case 3 had established estimation procedures, although not documented. Service Provider (Case 2) and Tech Giant (Case 3) had used the same practices for several years, whereas this was the first time for Software Vendor (Case 1) using the estimation procedure in question. The interviewees at Tech Giant and Service Provider reported that they had a history of making successful estimates, while the interviewees at Software Vendor stated that they tend to underestimate and have a poor track record in estimation.

The progress of the project was monitored from the estimation point of view in all case projects. In Case 1, the estimate was presented as a single point estimate. In Case 2, the estimate was presented as a range, consisting of an optimistic, pessimistic and nominal scenario. In Case 3, the target was to deliver at least 85% of the nominal estimate, which can also be seen as a range. The actual project team was more or less known in all projects at the time of estimation. The interviewees in Cases 2 and 3 report that the general estimation capabilities are good, emphasising the importance of professional competence in estimation. The interviewees in Case 1 reported that their estimation capabilities and experience are low. There has also been training related to estimation practices in

Table 6. Summary of SCE capability findings.

	Case 1	Case 2	Case 3
Company	Software Vendor	Service Provider	Tech Giant
Project	Tool	Operational Control System	Network Management System
Use of an estimation methodology	(–) No defined standard practice	(+) Work break-down, historical data, software tool	(+) Agile grooming, work break-down, historical data
Proper communication	(+) Assumptions presented (–) Single point	(+) Assumptions presented, range	(+) Assumptions presented, range
Re-estimation and follow-up	(+) Regular follow-up	(+) Regular follow-up	(+) Regular follow-up
Documented estimation procedure	(–) No documented or established procedure	(+) Documented procedure adjusted for the application area, improved continuously	(+) Established, but (–) Not documented
Other	(–) Short experience, low competence, poor track record	(+) Long experience, high competence, good track record	(+) Long experience, high competence, good track record

Case 2 and Case 3. In Case 2, at Service Provider, there was a named person who was responsible for developing estimation practices, which was not the case at the other two companies.

Applying the CMM scale from 1 (low maturity) to 5 (high maturity) and related behavioural characteristics [67, pp. 9–14] to SCE maturity, Service Provider (Case 2) was assessed as being on the highest level, level 5. Their estimation procedures produce reliable results, which are adjusted to specific application areas and technologies and there is systematic work to improve estimation practices. According to our assessment, Tech Giant (Case 3) is on level 4, meaning that while there is room for improvement, the standard processes are defined and established and produce reliable results. Finally, Software Vendor (Case 1) is on level 2, meaning that the processes are defined and may support the production of consistent results. However, in practice, the process discipline was low and the defined practices cannot be applied in real-world situations consistently and successfully.

Table 6 summarises the findings on the SCE procedures used in our case projects; categorised according to the SCE capability criteria defined in Section 4.2. The SCE maturity, when set against the criteria in Table 5, seems to correlate well with the CMM maturity levels and the related behavioural characteristics: Service Provider and Tech Giant have practices in place for repeating processes and gaining predictable results. This issue will be discussed more in Section 6.1. There was no standard practices that support the development of consistency at Software Vendor.

6.3.2. Software process maturity

In Case 1, the process used for Tool was relatively new, implemented in the first half of 2014, and was followed by an organisational change in the second half of 2014. The company was adopting Scrum methodology and abandoning the process used in the case project. The Senior Technical Manager of the company said that the primary focus has always been on programming at the cost of other things, such as leadership and PM. The interviewees also referred to similar overruns in projects resembling Tool.

In Case 2, the project manager reported that they deliver hundreds of projects yearly using the same delivery process as used in the case project. The processes are stable and under constant development. According to the Project Manager and Business Manager, the results have been generally good, which was also true of the case project. There has also been training related to the different aspects of the software project delivery model.

Also, Tech Giant in Case 3 has used the current Scrum based process for approximately seven years. According to the Line Manager, the process was under constant development, which was supported by comments from other interviewees. However, the two representatives from product management report that there is still much room for improvement, especially regarding the basing of estimates on current data instead of historical data and the managing of dependencies. Regardless of the pointers for improvement, the product management representative, and other interviewees, described the overall software development performance as good.

To recapitulate, according to our assessment of the overall software process maturity, Software Vendor (Case 1), Service Provider (Case 2) and Tech Giant (Case 3) are on the CMM levels 2, 5 and 5, respectively. A summary of the assessment findings is presented in Appendix B.

6.3.3. Attitudes

All the interviewees in this study recognised the importance of estimation. The reasons for the experienced importance varied. In Case 3, the Senior Manager argued that estimation facilitates the planning process before the actual work, connecting work to reality. In Case 1, the Project Manager stated that estimation is important from the planning perspective and the Testing Manager in Case 2 concurred. Nevertheless, estimation was experienced as being of high importance. In all case projects, the project manager had the overall responsibility for preparing the estimate. All of the project managers reported that their commitment to the estimate was high.

In Case 1, the general attitudes towards estimation were negative. For example the Senior Technical Manager, Project Manager and Product Owner argued that estimates were not trusted because they are likely to fail. The Senior Technical Manager stated that people are indifferent about the estimates because the usual reaction to overruns is just to continue the project. The Project Manager reported that he did not like giving an estimate and was afraid that the estimate would be interpreted as a commitment. During the re-estimation of the functionalities, the Project Manager described having given upper-bound estimates due to the high level of uncertainty, which also led to the implementation team's reluctance to estimate.

In Case 2, the Customer Manager describes the general attitude towards estimation as good and all the other interviewees agreed, reporting that estimation was a meaningful and motivating task. However, the Testing Manager and Software Developer report that when they are asked for quick and rough estimates, the work does not feel meaningful. They felt that some experts in their company, at Service Provider, take estimation too lightly, not necessarily recognising it as demanding and important work, although the importance of an estimate is understood by all. The Project Manager commented that estimates are sometimes given reluctantly because they are then interpreted as commitments. The Requirements Engineer reported that estimation was not necessarily a pleasant task due to its difficulty. However, the interviewees agreed that estimation generally works well.

In Case 3, the Requirements Engineer and Project Manager stated that estimation is not a pleasant task, though the discussions are seen as meaningful and relevant. Like the two interviewees in the Operational Control System project, the Requirement Engineer in the Network Management System project said making quick, rough estimates was not motivating. The Line Manager noted that estimators may be afraid that the estimates may not be as desired or that inaccurate estimates will lead to re-planning and corrective actions in the later phases of a project. Estimating was seen as an onerous responsibility. The Senior Manager commented that the development organisation should improve their estimation practices in order to improve the accuracy.

7. Discussion

The following Section 7.1 presents the key findings of this study. The remainder of this section will present the academical (Section 7.2) and practical implications (Section 7.3) of this study, addressing the study's limitations and giving pointers for future research (Section 7.4).

7.1. Key findings

In this study, we have focused on gaining insight into top management's role in SCE and discovering organisational phenomena that either support or hinder successful SCE. There were two main research questions: (RQ1) What are the real-world organisational factors that either support or hinder the creation of a meaningful software cost estimate? (RQ2) What is the impact of top management in either supporting or hindering software cost estimation practices?

The primary findings of the study are summarised in Table 7. We demonstrate that communication, attitudes and process maturity seem to support and hinder the creation of meaningful SCE (RQ1). Furthermore, top management's support and realism were found to support the results of SCE, although anchoring and the lack of a shared project vision were found to hinder SCE (RQ2). Finally, many of the factors affecting SCE, such as communication, providing resources and shared vision, have been found to affect project execution as well. This overlap is natural, since both SCE and project execution are inseparable parts of a software project. Our study, however, focuses on SCE influences, and presents evidence on factors affecting SCE specifically.

7.2. Implications for theory

It has been argued that only a very few papers examine the organisational context of SCE and how its methodologies are applied in real-world situations [39]. According to Jørgensen and Shepperd [39], the basic problems experienced by software companies in relation to SCE are not technical. Hence, this paper has specifically focused on the organisational context related to SCE and in increasing our understanding of the prerequisites for meaningful SCE. This paper also demonstrates that SCE research remains focused on technical issues, while the focus of PM research has undergone a major shift from a technical to a managerial focus.

The primary finding of this study is that there seems to be a connection between the software process maturity, estimation maturity and estimation success. The maturity as a construct consists of several factors. Our study did not identify individual significant organisational factors, which alone would make estimation successful. The connection between the maturity and estimation success suggests that successful estimation is a sum of several factors, such as communication, competence, experience and attitudes.

The more specific results from this study show that commonly used estimation techniques, WBS and expert estimation, can produce good results, if the overall project management and software practices are established and produce consistent results. This paper also suggests that communication is an important factor in the scope of SCE. Furthermore, the findings suggest that SCE should not set any specific requirements for top management, other than that they should carry out their basic responsibilities effectively and avoid the harmful anchoring of estimates.

The finding of this study also correlate well with the previous studies in the area of organisational context and human factors. From the organisational context point of view, Magazinovic and Pernstål [52] researched causes for estimation error, also validating results of Lederer and Prasad's [45] earlier study. They found that management goals affect the results of estimation. This seemed to happen also in Case 1 of our study. Also, in the same study, they found that unclear requirements are

Table 7. Summary of findings from the case projects by category

	Case 1	Case 2	Case 3
Company	Software Vendor	Service Provider	Tech Giant
Project	Tool	Operational Control System	Network Management System
Outcome	Challenged	Success	Success
Management role	(+) Strong support, realism pursued, enough resources (–) Anchoring, no shared project vision	(+) Strong support, realism pursued, enough resources	(+) Strong support, realism pursued, enough resources
Communication	(+) Detailed plans and specifications, prototype (–) Estimate prepared by one person, lack of discussions and cooperation	(+) Detailed plans and specifications, mutual understanding and insight pursued, cooperation intensive process, expertise and competence emphasised, shared project vision	(+) Aide memoir documentation, mutual understanding and insight pursued, cooperation intensive process, expertise and competence emphasised, shared project vision
Process maturity	(+) Documented software process, regular follow-up (–) No documented estimation procedure, non-established processes, no continuous improvement, no training arranged, low estimation experience and competence, no historical data used	(+) Documented software process, documented estimation procedure, established processes, continuous improvement, training, historical success, high estimation experience and competence, estimate as a range, regular follow-up	(+) Documented software process, established processes, continuous improvement, training, historical success, high estimation experience and competence, estimate as a range, regular follow-up (–) No documented estimation procedure
Attitudes	(+) Importance recognised (+) Project manager commitment high (–) Generally not pleasant, generally negative attitudes, indifference to failure, reluctance	(+) Importance recognised, estimation regarded as meaningful and motivating, general opinion that estimation works well (+) Project manager commitment high (–) Quick, rough estimates not motivating, sometimes unpleasant because of difficulty, some people do not recognise its seriousness, estimates interpreted as commitments	(+) Importance recognised, discussions regarded as meaningful and motivating, general opinion that estimation works well (+) Project manager commitment high (–) Generally not pleasant, quick, rough estimates not motivating, estimates interpreted as commitments, fear of failure, some reluctance

a source for estimation error, and that organisations do not have guidelines for conducting cost estimation. Our Case 1 suffered from unclear requirements, and Case 1 and Case 3 did not have guidelines for estimation. Furthermore, Magazinius, Börjesson and Feldt [49] found that personal agenda, management pressure and attempt to avoid re-estimation may affect the estimate. This seemed to be the case also in the Tool project of our study. Promotion of the project [51] may also explain parts of the tight target setting for the Tool project.

Cognitive bias is another non-technical topic related to SCE, which has gained attention recently. While our primary focus was in the organisational context, we discovered the presence of anchoring [40] in Case 1. There also seemed to be, at least to some extent, attitudinal tendency in all cases to find hindrances for estimation outside the respondent's direct influence, corresponding with [36].

Based on the results presented above, this paper supports the assumption that the estimation challenges experienced in companies are not only technical, but are also related to the organisational context, specifically to the project management and software process maturity. Also, easy to use estimation techniques may not be used by chance but because of the fact that these methods require less organisational capabilities for their successful application. These findings, along with similar findings, should justify SCE researchers shifting their research focus from technical topics to managerial and processual ones.

7.3. Implications for practice

This study addressed the top management's role in software cost estimation. In the following, we will discuss the practical advice found in this research. These are categorized into four groups: top management's role, the importance of communication, organization's process maturity and general attitudes towards SCE.

7.3.1. Top management role

This study suggests that by supporting SCE through the basic TMS practices found in this study, demonstrating SCE's importance, reviewing plans, providing resources and ensuring a shared vision and commitment, top management can create an environment for successful SCE. Earlier studies support this conclusion. For example Boonstra [9] has found that the provision of resources, the establishment of a clear and well defined project framework, communication with the project team, being knowledgeable about a project and using power to resolve conflicts are important behavioural categories for top management. Zwikael [91] has reported similar findings, and concludes that, e.g. an organisational structure that is supportive of a project, communication between the project manager and the organisation and appropriate project manager assignment have a positive impact on project success. However, the previously defined behaviour is likely to be enough only in an environment where management has already created the necessary capabilities and gained the required experience for successful software work.

On the other hand, the results indicate that if there is a lack of a shared vision or a lack of commitment, the negative impacts on SCE can be significant. This finding receives support from earlier studies. White and Fortune [85] report that 'Clear goals/objectives' was the most frequently mentioned success factor for projects. Fortune and White [21] report that 'Clear realistic goals' was the second most cited factor for success. However, clearly expressed expectations may also become harmful anchors and distort SCE, as found in this and other studies [4, 40].

In summary, successful SCE seems not to require any specific actions from top management, if the general maturity of a work environment is good. Thus, it is enough if management performs its role effectively by providing typical TMS behaviour. However, top management should avoid situations in which their expectations could become anchors that negatively affect SCE.

7.3.2. Communication

The results provide evidence that communication related issues are important factors in successful SCE, when work breakdown structure (WBS) and expert estimation are in use. In both of the successful projects, Cases 2 and 3, the interviewees reported that mutual understanding and understanding the requirements were sought by management. Furthermore, there were many opportunities and forums for discussions on the issues. Hence, cooperation was described as good and the expertise as sufficient for reaching an adequate level of understanding.

There are plenty of similar findings from other areas related to the importance of communication. In the scope of project cost management, [18] has found that early interaction with key stakeholders and the establishment of clear lines of communication for sharing professional and project based knowledge are crucial during the inception phases of projects. Furthermore, the significant role of communication in managing the coordination process has been addressed by Malone and Crowston [53]. Communication has been found to be a common success factor when discussing change in software projects and teams [80] and the best way to build trust in development teams [27]. Communication has also been found to make software development more efficient in companies [65] and is one of the cornerstones of agile development [7]. In the scope of SCE, Jørgensen [36] noted, in a case study, that poor communication skills or team dynamics might have had an impact on the SCE's result in one team.

On a practical level, these findings suggest that project managers, software professionals and other project team members should focus on achieving an understanding of requirements through discussion, instead of focusing on compliance, techniques and documentation.

7.3.3. Process maturity

All of the case projects used easy to implement [37] estimation methodologies, such as WBS, expert estimation and group estimation. The methodologies seem to produce useful results in a mature environment. Established processes and at least moderate maturity seem to be the key to successful application of estimation methodologies. This conclusion also receives support from earlier research. The success of expert estimation has been shown by Jørgensen [35] and studies on the impact of CMM levels on estimation results show that companies who have levels from three to five produce significantly more accurate results than companies on the lower maturity levels [20, 55, p. 10]. However, although the estimation accuracy and CMM level seem to correlate with each other, we would like to point out that there seems to be no significant correlation between the project management maturity (PMM) of an organisation and the project success [86]. The correlation between the CMM level and estimation accuracy observed in this study occurs within the studied area of maturity, SCE being part of the software process maturity.

Maturity as a construct consists of several factors, like experience, skills and processes. While we report several maturity related findings connected to successful estimation, like training, experience and continuous improvement, we believe that none of the individual factors is likely to lead to success on its own. However, a lack of one of those factors may have significant negative impacts. Thus, based on our findings, we decided to focus on maturity as a whole, instead of individual factors.

Software process maturity (or project management maturity), estimation maturity and attitudes seem to have a clear interrelationship. If software process maturity is good, estimation maturity seems to be good, furthermore attitudes become more positive. This is not surprising, because SCE is part of a software project and managed under the relevant software project or software process management. The CMM model does not include attitudes in its attributes, although, for example, [3] suggest attitudes are an important factor in project management maturity, in addition to knowledge and action. However, the true relationship between these three is beyond the scope of our study.

Considering the previous and the findings presented in Table 7, it seems intuitive that the overall maturity correlates with the estimation success. This is supported by Flowe and Thordahl [20] and findings from Boeing, presented by McConnell [55, p.10]. Furthermore, each of the elements of maturity is likely to contribute to estimation success also individually. For example Jørgensen [35] has provided evidence that training opportunities, good estimator competence and use of an estimation checklist improve estimation success. In other words, the more there are elements of high maturity present, the higher is the probability of estimation success, and vice versa, low presence of high maturity elements increases uncertainties in estimation.

Our advice for organisations would be to include a simple maturity self assessment in the software cost estimation process, for example based on a publicly available criteria like CMM or CMMI. If the maturity is assessed to be low, a thorough uncertainty analysis is appropriate. Even knowledge of a high level of uncertainty may help managers in their decision making, even though the uncertainties could not be mitigated. Also, we understand that self assessments are perhaps not typical for low maturity organisations. However, use of a simple maturity assessment is far easier than accounting the whole industrial and scientific body of knowledge as individual items. In the beginning, the awareness of the high level of uncertainty could help to make better decisions, and in the longer term act as a list of development pointers towards higher maturity.

For the practitioners in higher maturity organisations, we suggest addressing specific estimation challenges, like estimating change requests or estimating testing. For example, those two areas seem to be sources of errors [72] and serve to decrease motivation, even in exemplary organisations. Also the relationship between the estimate, target and commitment is not always clear, which has been reported as resulting in a reluctance to make estimates; the importance of making a distinction between these three aspects is addressed by McConnell [55].

7.3.4. Attitudes

In cases 2 and 3, project managers had the overall responsibility of preparing the estimate, while the actual estimation was done by software developers. In both projects the estimation was seen as an important and relevant task, and the project managers reported that they were committed to the estimates.

However, in both projects the developers' attitudes towards estimation were negative. Estimation was not considered as a pleasant task and reluctancy and low motivation were reported, especially originating from lack of trust and quickly emerging needs requiring flexibility. Negative attitudes, low motivation and reluctance have been found to decrease the quality of work [73]. Although estimates and outcomes have correlated well in these two projects, it is likely that the risk of estimation error increases when negative attitudes are present, especially in low maturity organisations. Trust and flexibility as values have been found to have a positive effect on project outcome [90]. A trivial advice is to support a positive atmosphere around estimation. However, further research is needed to provide better and more specific advice on this topic.

7.4. Limitations and future work

Although we have taken a number of countermeasures to validity threats (see Section 5.6), and improved the transferability of the results by collecting a rich set of data, this research has certain limitations. This research considered the organisational phenomena on a general level, without taking the project or organisation specific characteristics, like development methodology or company size, into account in the study design.

The findings provide evidence that, on a general level, organisational issues, like the role of management, process maturity and communication, are important factors in SCE. However, although we believe that the results are transferable to similar project settings, the organisational challenges may vary between different contexts. For example, we may have overlooked some organisational properties or mechanisms, like the size of the company, which causes variation between projects. In addition, there are different reasons for the cost estimates: one company was using them to set the price to the customer while the others were seeking balancing content and timing of their products with the estimates.

Therefore, we encourage further studies in different project and company contexts to see if the same phenomena are repeated, or if there are other context specific phenomena not discovered in this study. Quantitative studies would also provide insight in how commonly the reported phenomena repeat in organisations.

This study also provides evidence that there is an interrelationship between the estimation maturity and project management maturity. This is an important observation, and should be confirmed with a quantitative study that considers a large number of projects as well as studied qualitatively to understand the phenomenon. For example, it might just be that companies with a low CMM level do not recognize that there are situations when it is inappropriate to estimate at all (e.g., new development and estimation methods, new product with no client). This is a lack of risk management procedures, not just an estimation problem.

The findings of this paper are based on three projects, and do not provide a generalizable level of confidence for their relationship. The SCE maturity and software process maturity were also assessed only to the extent necessary for the purposes of this study. We suggest that further studies establish a more precise model for assessing SCE maturity and conduct the actual maturity assessment with maturity as the sole focus of the study.

As an exploratory study, the purpose was also to generate new theories and pointers for further research. One interesting observation revealed by this study was that the attitudes towards estimation were negative among the developers participating in estimation, whereas the attitudes of the project managers were positive and the level of commitment to the estimation high. Negative attitudes may be a source of estimation errors, and increase the probability of overruns. This should be studied further, since negative attitudes hinder any work.

From the construct point of view, the aim was to discover organisational factors affecting SCE. We covered many relevant aspects related to the organisational context in which the estimation took place. Thus, we studied what we planned to study and felt that we developed a clear picture of each of the studied projects.

Generally speaking this study has found that management and process related topics are equally important from the SCE point of view as estimation technique related topics. This suggests that SCE research would benefit from approaching those topics from a PM or software process point of view, and that elements from these areas should be synthesised into SCE research. Lastly, as demonstrated in the introduction of this paper, e.g. PM research is more advanced than SCE research on management and other organisation related issues.

8. Conclusions

Many researchers and practitioners argue that organisational issues are equally important from the software estimation success point of view as technical issues. Some of the often cited works related to this important topic have been Lederer and Prasad [45], Jørgensen and Shepperd [39] and Magazinovic and Pernstål [52]. Regardless of this knowledge of the importance of organisational issues in SCE, the focus of the SCE research has remained heavily on estimation methodologies and other technical issues.

The findings of this paper have potential to contribute to the current body of knowledge on organisational issues related to SCE, and specifically on top management's role, in several ways, regardless of the limited transferability of the results. By using exploratory case study approach and interviewing 15 practitioners involved in software development in three organisations, we have found that the role of top management is important in creating prerequisites for meaningful estimation, but their day-to-day participation is not required for successful estimation. Top management may also induce undesired distortion in estimation. We have also found that estimation maturity and estimation success seem to have an interrelationship with software process maturity, but there seem to be no significant individual organisational factors, which alone would make estimation successful. Additionally, our study validated many of the distortions and biases reported in the earlier studies, and showed that the SCE research focus has remained on estimation methodologies.

Low maturity organisations may be able to reduce overruns through a better understanding of their increased risk level and the existence of good estimation practices. We suggest therefore that future studies and software process improvement activities should pay more attention to low maturity organisations and their specific needs.

Acknowledgements

The authors gratefully acknowledge the support of Tekes — the Finnish Funding Agency for Innovation, DIMECC Oy and the Need for Speed (http://www.n4s.fi) research programme. The funders had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

[1] S. Adams. *ITIL V3 foundation handbook. Vol. 1.* The Stationery Office, Belfast, Ireland, 2009.
[2] J. J. Ahonen, P. Savolainen, H. Merikoski, and J. Nevalainen. Reported project management effort, project size, and contract type. *Journal of Systems and Software*, 109:205–213, 2015.
[3] E. S. Andersen and S. A. Jessen. Project maturity in organisations. *International journal of project management*, 21(6):457–461, 2003.
[4] J. Aranda and S. Easterbrook. Anchoring and adjustment in software estimation. In *Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ESEC/FSE-13, pages 346–355, New York, NY, USA, 2005. ACM.
[5] J. Arnowitz, M. Arent, and N. Berger. *Effective prototyping for software makers.* Elsevier, 2010.
[6] C. M. Beath. Supporting the information technology champion. *MIS quarterly*, pages 355–372, 1991.
[7] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, et al. Manifesto for agile software development. http://www.agilemanifesto.org, 2001.
[8] B. W. Boehm, R. Madachy, B. Steece, et al. *Software cost estimation with Cocomo II.* Prentice Hall PTR, 2000.
[9] A. Boonstra. How do top managers support strategic information system projects and why do they sometimes withhold this support? *International Journal of Project Management*, 31(4):498–512, 2013.
[10] L. C. Briand and I. Wieczorek. Resource estimation in software engineering. In *Encyclopedia of Software Engineering.* John Wiley & Sons, Inc., 2002.
[11] N. Cerpa and J. M. Verner. Why did your project fail? *Communications of the ACM*, 52(12):130–134, 2009.

[12] D. I. Cleland. The strategic context of projects. In L. Dye and J. Pennypacker, editors, *Project portfolio management – selecting and prioritizing projects for competitive advantage*. Center for Business Practices, West Chester, PA, USA, 1999.

[13] CMMI Product Team. CMMI for development. version 1.3. Technical Report CMU/SEI-2010-TR-0336, Software Engineering Institute, Carnegie Mellon University, 2010.

[14] M. Cohn. *Agile Estimating and Planning*. Robert C. Martin Series. Pearson Education, Inc., Upper Saddle River, NJ, October 2005.

[15] T. Cooke-Davies. The 'real' success factors on projects. *International journal of project management*, 20(3):185–190, 2002.

[16] L. Crawford. Senior management perceptions of project management competence. *International journal of project management*, 23(1):7–16, 2005.

[17] P. Dietrich and P. Lehtonen. Successful management of strategic intentions through multiple projects–reflections from empirical study. *International Journal of Project Management*, 23(5):386–391, 2005.

[18] H. K. Doloi. Understanding stakeholders' perspective of cost estimation in project management. *International journal of project management*, 29(5):622–636, 2011.

[19] Y. K. Dwivedi, D. Wastell, S. Laumer, H. Z. Henriksen, M. D. Myers, D. Bunker, A. Elbanna, M. Ravishankar, and S. C. Srivastava. Research on information systems failures and successes: Status update and future directions. *Information Systems Frontiers*, 17(1):143–157, 2015.

[20] R. M. Flowe and J. B. Thordahl. A correlational study of the sei's capability maturity model and software development performance in dod contracts. Technical report, DTIC Document, 1994.

[21] J. Fortune and D. White. Framing of project critical success factors by a systems model. *International Journal of Project Management*, 24(1):53–65, 2006.

[22] D. Galorath. Software project failure costs billions. *Better Estimation & Planning Can Help.[online] Available at: http://www. galorath. com/wp/software-project-failure-costs-billions-betterestimation-planning-can-help. php [Accessed 16 September 2014]*, 2012.

[23] J. T. Garrity. Top management and computer profits. *Harvard Business Review*, 41(4):6–12, 1963.

[24] D. R. Goldenson and D. L. Gibson. Demonstrating the impact and benefits of CMMI (trademark): An update and preliminary results. Technical report, DTIC Document, 2003.

[25] D. R. Goldenson and J. D. Herbsleb. After the appraisal: A systematic survey of process improvement, its benefits, and factors that influence success. Technical report, DTIC Document, 1995.

[26] T. Halkjelsvik and M. Jørgensen. From origami to software development: A review of studies on judgment-based predictions of performance time. *Psychological bulletin*, 138(2):238, 2012.

[27] K. Henttonen and K. Blomqvist. Managing distance in a global virtual team: the evolution of trust through technology-mediated relational communication. *Strategic Change*, 14(2):107–119, 2005.

[28] J. Hihn and H. Habib-agahi. Cost estimation of software intensive projects: A survey of current practices. In *Proceedings of the 13th international conference on Software engineering*, pages 276–287. IEEE Computer Society Press, 1991.

[29] D. L. Hughes, Y. K. Dwivedi, A. C. Simintiras, and N. P. Rana. *Success and failure of IS/IT projects: A state of the art analysis and future directions*. Springer, 2015.

[30] P. M. Institute. *A Guide to the Project Management Body of Knowledge: PMBOK Guide*. PMBOK® Guide Series. Project Management Institute, 2013.

[31] P. M. Institute. *Software Extension to the PMBOK Guide, Fifth ed. Project Management Institute (PMI).* USA: IEEE Computer Society, 2013.

[32] IPMA, ICB. Ipma competence baseline version 3. *Nijkerk, The Netherlands*, 2006.

[33] ISO. *21500: 2012: Guidance on Project Management.* ISO, Geneva, Switzerland, 2012.

[34] J. J. Jiang, G. Klein, H.-G. Hwang, J. Huang, and S.-Y. Hung. An exploration of the relationship between software development process maturity and project performance. *Information & Management*, 41(3):279–288, 2004.

[35] M. Jørgensen. A review of studies on expert estimation of software development effort. *Journal of Systems and Software*, 70(1):37–60, 2004.

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

[37] M. Jørgensen. Practical guidelines for expert-judgment-based software effort estimation. *IEEE software*, 22(3):57–63, 2005.

[38] M. Jørgensen. Communication of software cost estimates. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, page 28. ACM, 2014.

[39] M. Jørgensen and M. Shepperd. A systematic review of software development cost estimation studies. *IEEE Transactions of Software Engineering*, 33(1):33–53, January 2007.

[40] M. Jørgensen and D. I. Sjøberg. The impact of customer expectation on software development effort estimates. *International Journal of Project Management*, 22(4):317–325, 2004.

[41] H. K. Klein and M. D. Myers. A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS quarterly*, pages 67–93, 1999.

[42] B. J. Kolltveit, J. T. Karlsen, and K. K Grønhaug. Perspectives on project management. *International Journal of Project Management*, 25(1):3–9, January 2007.

[43] Y. Kwak and F. Anbari. Analyzing project management research: Perspectives from top management journals. *International journal of project management*, 27(5):435–446, 2009.

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

[45] A. L. Lederer and J. Prasad. Causes of inaccurate software development cost estimates. *Journal of systems and software*, 31(2):125–134, 1995.

[46] T. Little. Schedule estimation and uncertainty surrounding the cone of uncertainty. *IEEE software*, 23(3):48, 2006.

[47] L. Luna-Reyes, J. Zhang, J. Gil-Garćia, and A. Cresswell. Software developments development as emergent socio-technical change: a practice approach. *European Journal of Software developments*, 14(1):93–105, 2005.

[48] M. Lycett, A. Rassau, and J. Danson. Programme management: a critical review. *International Journal of Project Management*, 22(4):289–299, 2004.

[49] A. Magazinius, S. Börjesson, and R. Feldt. Investigating intentional distortions in software cost estimation — an exploratory study. *Journal of Systems and Software*, 85(8):1770–1781, August 2012.

[50] A. Magazinius and R. Feldt. Exploring the human and organizational aspects of software cost estimation. In *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*, pages 92–95. ACM, 2010.

[51] A. Magazinius and R. Feldt. Confirming distortional behaviors in software cost estimation practice. In *Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on*, pages 411–418. IEEE, 2011.

[52] A. Magazinovic and J. Pernstål. Any other cost estimation inhibitors? In *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measure-*

ment, pages 233–242. ACM, 2008.

[53] T. W. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, 1994.

[54] M. L. Markus. Implementation politics: Top management support and user involvement. Technical Report 75, Cambridge, Mass.: Center for Information Systems Research, Alfred P. Sloan School of Management, 1981.

[55] S. McConnell. *Software Estimation: Demystifying the Black Art*. Microsoft Press, Redmond, Washington, USA, 2006.

[56] L. McLeod and S. G. MacDonell. Factors that affect software systems development project outcomes: A survey of research. *ACM Computing Surveys (CSUR)*, 43(4):24, 2011.

[57] J. McManus and T. Wood-Harper. A study in project failure. bcs, 2008.

[58] D. Milosevic and P. Patanakul. Standardized project management may increase development projects success. *International Journal of Project Management*, 23(3):181–192, 2005.

[59] G. Morton. Become a project champion. *International Journal of Project Management*, 1(4):197–203, 1983.

[60] B. Nanus and L. Farr. Some cost contributors to large-scale programs. In *Proceedings of the April 21-23, 1964, Spring Joint Computer Conference*, AFIPS '64 (Spring), pages 239–248, New York, NY, USA, 1964. ACM.

[61] M. H. N. Nasir and S. Sahibuddin. Critical success factors for software projects: A comparative study. *Scientific research and essays*, 6(10):2174–2186, 2011.

[62] E. A. Nelson. Management handbook for the estimation of computer programming costs. Technical Documentary Report No. ESD-TR-67-66, United States Air Force, Bedford, Massachusetts, 2002.

[63] R. R. Nelson. It project management: infamous failures, classic mistakes, and best practices. *MIS Quarterly Executive*, 6(2):67–78, 2007.

[64] T. Okoro. Diverse talent: Enhancing gender participation in project management. *Procedia — Social and Behavioral Sciences*, 226:170–175, 2016. Proceedings of the 29th IPMA World Congress WC2015 (28-30 September – 1 October, Panama).

[65] M. Paasivaara and C. Lassenius. Collaboration practices in global inter-organizational software development projects. *Software Process: Improvement and Practice*, 8(4):183–199, 2003.

[66] M. Patton. *Qualitative Research and Evaluation Method*. SAGE Publications, Inc., Thousand Oaks, California, USA, third edition edition, 2001.

[67] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber. Capability maturity model for software, version 1.1. *Software Engineering Institute-Carnegie Mellon University. Pittsburgh, Pennsylvania*, 1993.

[68] M. Petrini and M. Pozzebon. Managing sustainability with the support of business intelligence: Integrating socio-environmental indicators and organisational context. *The Journal of Strategic Information Systems*, 18(4):178–191, 2009.

[69] B. Pitterman. Telcordia technologies: The journey to high maturity. *IEEE Software*, 17(4):89–96, July 2000.

[70] L. Putnam and W. Myers. *Five core metrics: The Intelligence Behind Successful Software Management*. Dorset House Publishing, New York, USA, 2003.

[71] J. Rahikkala, S. Hyrynsalmi, and V. Leppänen. Accounting testing in software cost estimation: A case study of the current practice and impacts. In J. Nummenmaa, O. Sievi-Korte, and E. Mäkinen, editors, *Proceedings of 14th Symposium on Programming Languages and Software Tools*, pages 64–75, Tampere, Finland, 2015. University of Tampere.

[72] J. Rahikkala, V. Leppänen, J. Ruohonen, and J. Holvitie. Top management support in software cost estimation: A study of attitudes and practice in finland. *International Journal of Managing*

Projects in Business, 8(3):513–532, 2015.

[73] A. Raja, W. Mohsin, N. Ehsan, E. Mirza, and M. Saud. Impact of emotional intelligence and work attitude on quality of service in the call centre industry of pakistan. In *Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference on*, pages 402–407. IEEE, 2010.

[74] C. Robson. *Real world research: A resource for social scientists and practitioner-researchers*. Blackwell Publishing, Oxford, UK, second edition edition, 2002.

[75] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14:131–164, April 2009.

[76] R. Schmidt, K. Lyytinen, and P. C. Mark Keil. Identifying software project risks: An international delphi study. *Journal of management information systems*, 17(4):5–36, 2001.

[77] S. K. Sehra, Y. S. Brar, N. Kaur, and S. S. Sehra. Research patterns and trends in software effort estimation. *Information and Software Technology*, 91:1–21, 2017.

[78] J. Smyrk. Why most it projects are really it without the project. In *Third world project management conference, Gold Coast, Australia*, 2002.

[79] S. Srivannaboon and D. Z. Milosevic. A two-way influence between business strategy and project management. *International journal of project management*, 24(6):493–505, 2006.

[80] D. Stelzer and W. Mellis. Success factors of organizational change in software process improvement. *Software Process: Improvement and Practice*, 4(4):227–250, 1998.

[81] C. P. Team. CMMI for development, version 1.3 (cmu/sei-2010-tr-033). software engineering institute, 2010.

[82] The Standish Group International. The CHAOS manifesto: Think big and act small, 2013.

[83] J. Thomas, C. Delisle, K. Jugdev, and P. Buckle. Selling project management to senior executives: the case for avoiding crisis sales? *Project Management Journal*, 33(2):19–28, June 2002.

[84] R. Turner. *The handbook of project based management*. McGraw-Hill, London, UK, 2nd edition, 1999.

[85] D. White and J. Fortune. Current practice in project management – an empirical study. *International journal of project management*, 20(1):1–11, 2002.

[86] H. J. Yazici. The role of project management maturity and organizational culture in perceived performance. *Project Management Journal*, 40(3):14–33, 2009.

[87] R. K. Yin. *Case Study Research: Design and Methods*. SAGE Publications, Inc., Thousands Oaks, California, third edition edition, 2003.

[88] R. Young and E. Jordan. Lifting the game: Board views on e-commerce risk. *IFIP TG8*, 6:102–113, 2002.

[89] R. Young and E. Jordan. Top management support: Mantra or necessity? *International journal of project management*, 26(7):713–725, 2008.

[90] Y. J. Zidane, B. A. Hussein, J. Ø. Gudmundsson, and A. Ekambaram. Categorization of organizational factors and their impact on project performance. *Procedia-Social and Behavioral Sciences*, 226:162–169, 2016.

[91] O. Zwikael. Top management involvement in project management: A cross country study of the software industry. *International Journal of Managing Projects in Business*, 1(4):498–511, 2008.

A. Interview instrument

1. Introduction (approximately 5 minutes)
   – A brief introduction to the study.
   – An introduction of the benefits of participation
   – Anonymity and confidentiality.
2. Personal, team and project background (approximately 5 minutes)
   – Interviewee's personal history and job position in the company.
   – Background of the estimated project and the development methodology that was used.
3. Current state of SCE in the organisation (approximately 25 minutes)
   – Describe the procedure for creating the estimate.
   – Describe the method for creating the estimate of the effort required.
   – Describe the responsibilities related to maintaining and improving the software and estimation practices.
   – Describe the outcome of the estimation.
   – Describe the approach to re-estimation during the project.
4. Experiences of organisational phenomena affecting the four SCE aspects (approximately 20 minutes)
   – Describe the management, project manager and personal expectations of the estimate.
   – Describe the overall SCE skills and motivation in your organisation during the estimation.
   – Describe the demonstrated importance and attitudes regarding the estimate.
   – Describe the ways in which top management and other stakeholders were involved in SCE.
   – Did the project have clear goals and realistic expectations?
   – Was there pressure to make the estimate smaller or other pressures?
   – Was the estimate allowed to change over time?
   – Was there enough time allocated for preparing the estimate?
   – Did all stakeholders seek realistic and accurate estimates?
   – What was the level of commitment of different stakeholders to the estimate?
   – What were the primary issues hindering and supporting successful estimation?
5. Ending (approximately 5 minutes)
   – Any other relevant observations that we have not covered?

B. Software process CMM level assessment summary

The following tables 8, 9, 10 and 11 presents our CMM assessments for levels 2, 3, 4 and 5, respectively, to the case study companies.

Table 8. The Key Process Areas for Level 2: Repeatable

Process area	Goal	Software Vendor	Service Provider	Tech Giant
Requirements Management	System requirements allocated to software are controlled to establish a baseline for software engineering and management use.	Yes.	Yes.	Yes.
Requirements Management	Software plans, products, and activities are kept consistent with the system requirements allocated to software.	Yes.	Yes.	Yes.
Software Project Planning	Software estimates are documented for use in planning and tracking the software project.	Yes.	Yes.	Yes.
Software Project Planning	Software project activities and commitments are planned and documented.	Yes.	Yes.	Yes.
Software Project Planning	Affected groups and individuals agree to their commitments related to the software project.	N/A	N/A	N/A
Software Project Tracking and Oversight	Actual results and performances are tracked against the software plans.	Yes.	Yes.	Yes.
Software Project Tracking and Oversight	Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans.	Yes.	Yes.	Yes.
Software Project Tracking and Oversight	Changes to software commitments are agreed to by the affected groups and individuals.	N/A	N/A	N/A
Software Subcontract Management	The prime contractor selects qualified software subcontractors.	N/A	N/A	N/A
Software Subcontract Management	The prime contractor and the software subcontractor agree to their commitments to each other.	N/A	N/A	N/A
Software Subcontract Management	The prime contractor and the software subcontractor maintain ongoing communications.	N/A	N/A	N/A
Software Subcontract Management	The prime contractor tracks the software subcontractor's actual results and performance against its commitments.	N/A	N/A	N/A
Software Quality Assurance	Software quality assurance activities are planned.	Yes.	Yes.	Yes.
Software Quality Assurance	Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively.	Yes.	Yes.	Yes.
Software Quality Assurance	Affected groups and individuals are informed of software quality assurance activities and results.	Yes.	Yes.	Yes.
Software Quality Assurance	Noncompliance issues that cannot be resolved within the software project are addressed by senior management.	N/A	N/A	N/A
Software Network Management Management	Software configuration management activities are planned.	N/A	N/A	N/A
Software Network Management Management	Selected software work products are identified, controlled, and available.	N/A	N/A	N/A
Software Network Management Management	Changes to identified software work products are controlled.	N/A	N/A	N/A
Software Network Management Management	Affected groups and individuals are informed of the status and content of software baselines.	N/A	N/A	N/A

Notes: Yes. - assessment provides evidence of fulfilling the goal; No. - assessment provides evidence of not fulfilling the goal; N/A - fulfillment of the goal was not assessed.

Table 9. The Key Process Areas for Level 3: Defined

Process area	Goal	Software Vendor	Service Provider	Tech Giant
Organization Process Focus	Software process development and improvement activities are coordinated across the organization.	No.	Yes.	Yes.
Organization Process Focus	The strengths and weaknesses of the software processes used are identified relative to a process standard.	N/A	N/A	N/A
Organization Process Focus	Organization-level process development and improvement activities are planned.	No.	Yes.	Yes.
Organization Process Definition	A standard software process for the organization is developed and maintained.	Yes.	Yes.	Yes.
Organization Process Definition	Information related to the use of the organization's standard software process by the software projects is collected, reviewed, and made available.	N/A	N/A	N/A
Training Program	Training activities are planned.	No.	Yes.	Yes.
Training Program	Training for developing the skills and knowledge needed to perform software management and technical roles is provided.	No.	Yes.	Yes.
Training Program	Individuals in the software engineering group and software-related groups receive the training necessary to perform their roles.	No.	Yes.	Yes.
Integrated Software Management	The project's defined software process is a tailored version of the organization's standard software process.	N/A	N/A	N/A
Integrated Software Management	The project is planned and managed according to the project's defined software process.	Yes.	Yes.	Yes.
Software Product Engineering	The software engineering tasks are defined, integrated, and consistently performed to produce the software.	Yes.	Yes.	Yes.
Software Product Engineering	Software work products are kept consistent with each other.	N/A	N/A	N/A
Intergroup Coordination	The customer's requirements are agreed to by all affected groups.	No.	Yes.	Yes.
Intergroup Coordination	The commitments between the engineering groups are agreed to by the affected groups.	N/A	N/A	N/A
Intergroup Coordination	The engineering groups identify, track, and resolve intergroup issues.	N/A	N/A	N/A
Peer Reviews	Peer review activities are planned.	No.	Yes.	Yes.
Peer Reviews	Defects in the software work products are identified and removed.	Yes.	Yes.	Yes.

Notes: Yes. - assessment provides evidence of fulfilling the goal; No. - assessment provides evidence of not fulfilling the goal; N/A - fulfillment of the goal was not assessed.

Table 10. The Key Process Areas for Level 4: Managed

Process area	Goal	Software Vendor	Service Provider	Tech Giant
Quantitative Process Management	The quantitative process management activities are planned.	No.	Yes.	Yes.
Quantitative Process Management	The process performance of the project's defined software process is controlled quantitatively.	N/A	N/A	N/A
Quantitative Process Management	The process capability of the organization's standard software process is known in quantitative terms.	No.	Yes.	Yes.
Software Quality Management	The project's software quality management activities are planned.	Yes.	Yes.	Yes.
Software Quality Management	Measurable goals for software product quality and their priorities are defined.	N/A	N/A	N/A
Software Quality Management	Actual progress toward achieving the quality goals for the software products is quantified and managed.	N/A	N/A	N/A

Notes: Yes. - assessment provides evidence of fulfilling the goal; No. - assessment provides evidence of not fulfilling the goal; N/A - fulfillment of the goal was not assessed.

Table 11. The Key Process Areas for Level 5: Optimizing

Process area	Goal	Software Vendor	Service Provider	Tech Giant
Defect Prevention	Defect prevention activities are planned.	Yes.	Yes.	Yes.
Defect Prevention	Common causes of defects are sought out and identified.	N/A	N/A	Yes.
Defect Prevention	Common causes of defects are prioritized and systematically eliminated.	N/A	N/A	N/A
Technology Change Management	Incorporation of technology changes are planned.	N/A	N/A	N/A
Technology Change Management	New technologies are evaluated to determine their effect on quality and productivity.	N/A	N/A	N/A
Technology Change Management	Appropriate new technologies are transferred into normal practice across the organization.	N/A	N/A	N/A
Process Change Management	Continuous process improvement is planned.	No.	Yes.	Yes.
Process Change Management	Participation in the organization's software process improvement activities is organization wide.	No.	Yes.	Yes.
Process Change Management	The organization's standard software process and the project's' defined software processes are improved continuously.	No.	Yes.	Yes.

Notes: Yes. - assessment provides evidence of fulfilling the goal; No. - assessment provides evidence of not fulfilling the goal; N/A - fulfillment of the goal was not assessed.

Turku Centre for Computer Science
TUCS Dissertations

1. **Marjo Lipponen**, On Primitive Solutions of the Post Correspondence Problem
2. **Timo Käkölä**, Dual Information Systems in Hyperknowledge Organizations
3. **Ville Leppänen**, Studies on the Realization of PRAM
4. **Cunsheng Ding**, Cryptographic Counter Generators
5. **Sami Viitanen**, Some New Global Optimization Algorithms
6. **Tapio Salakoski**, Representative Classification of Protein Structures
7. **Thomas Långbacka**, An Interactive Environment Supporting the Development of Formally Correct Programs
8. **Thomas Finne**, A Decision Support System for Improving Information Security
9. **Valeria Mihalache**, Cooperation, Communication, Control. Investigations on Grammar Systems.
10. **Marina Waldén**, Formal Reasoning About Distributed Algorithms
11. **Tero Laihonen**, Estimates on the Covering Radius When the Dual Distance is Known
12. **Lucian Ilie**, Decision Problems on Orders of Words
13. **Jukkapekka Hekanaho**, An Evolutionary Approach to Concept Learning
14. **Jouni Järvinen**, Knowledge Representation and Rough Sets
15. **Tomi Pasanen**, In-Place Algorithms for Sorting Problems
16. **Mika Johnsson**, Operational and Tactical Level Optimization in Printed Circuit Board Assembly
17. **Mats Aspnäs**, Multiprocessor Architecture and Programming: The Hathi-2 System
18. **Anna Mikhajlova**, Ensuring Correctness of Object and Component Systems
19. **Vesa Torvinen**, Construction and Evaluation of the Labour Game Method
20. **Jorma Boberg**, Cluster Analysis. A Mathematical Approach with Applications to Protein Structures
21. **Leonid Mikhajlov**, Software Reuse Mechanisms and Techniques: Safety Versus Flexibility
22. **Timo Kaukoranta**, Iterative and Hierarchical Methods for Codebook Generation in Vector Quantization
23. **Gábor Magyar**, On Solution Approaches for Some Industrially Motivated Combinatorial Optimization Problems
24. **Linas Laibinis**, Mechanised Formal Reasoning About Modular Programs
25. **Shuhua Liu**, Improving Executive Support in Strategic Scanning with Software Agent Systems
26. **Jaakko Järvi**, New Techniques in Generic Programming – C++ is more Intentional than Intended
27. **Jan-Christian Lehtinen**, Reproducing Kernel Splines in the Analysis of Medical Data
28. **Martin Büchi**, Safe Language Mechanisms for Modularization and Concurrency
29. **Elena Troubitsyna**, Stepwise Development of Dependable Systems
30. **Janne Näppi**, Computer-Assisted Diagnosis of Breast Calcifications
31. **Jianming Liang**, Dynamic Chest Images Analysis
32. **Tiberiu Seceleanu**, Systematic Design of Synchronous Digital Circuits
33. **Tero Aittokallio**, Characterization and Modelling of the Cardiorespiratory System in Sleep-Disordered Breathing
34. **Ivan Porres**, Modeling and Analyzing Software Behavior in UML
35. **Mauno Rönkkö**, Stepwise Development of Hybrid Systems
36. **Jouni Smed**, Production Planning in Printed Circuit Board Assembly
37. **Vesa Halava**, The Post Correspondence Problem for Market Morphisms
38. **Ion Petre**, Commutation Problems on Sets of Words and Formal Power Series
39. **Vladimir Kvassov**, Information Technology and the Productivity of Managerial Work
40. **Frank Tétard**, Managers, Fragmentation of Working Time, and Information Systems

41. **Jan Manuch**, Defect Theorems and Infinite Words
42. **Kalle Ranto**, Z_4-Goethals Codes, Decoding and Designs
43. **Arto Lepistö**, On Relations Between Local and Global Periodicity
44. **Mika Hirvensalo**, Studies on Boolean Functions Related to Quantum Computing
45. **Pentti Virtanen**, Measuring and Improving Component-Based Software Development
46. **Adekunle Okunoye**, Knowledge Management and Global Diversity – A Framework to Support Organisations in Developing Countries
47. **Antonina Kloptchenko**, Text Mining Based on the Prototype Matching Method
48. **Juha Kivijärvi**, Optimization Methods for Clustering
49. **Rimvydas Rukšėnas**, Formal Development of Concurrent Components
50. **Dirk Nowotka**, Periodicity and Unbordered Factors of Words
51. **Attila Gyenesei**, Discovering Frequent Fuzzy Patterns in Relations of Quantitative Attributes
52. **Petteri Kaitovaara**, Packaging of IT Services – Conceptual and Empirical Studies
53. **Petri Rosendahl**, Niho Type Cross-Correlation Functions and Related Equations
54. **Péter Majlender**, A Normative Approach to Possibility Theory and Soft Decision Support
55. **Seppo Virtanen**, A Framework for Rapid Design and Evaluation of Protocol Processors
56. **Tomas Eklund**, The Self-Organizing Map in Financial Benchmarking
57. **Mikael Collan**, Giga-Investments: Modelling the Valuation of Very Large Industrial Real Investments
58. **Dag Björklund**, A Kernel Language for Unified Code Synthesis
59. **Shengnan Han**, Understanding User Adoption of Mobile Technology: Focusing on Physicians in Finland
60. **Irina Georgescu**, Rational Choice and Revealed Preference: A Fuzzy Approach
61. **Ping Yan**, Limit Cycles for Generalized Liénard-Type and Lotka-Volterra Systems
62. **Joonas Lehtinen**, Coding of Wavelet-Transformed Images
63. **Tommi Meskanen,** On the NTRU Cryptosystem
64. **Saeed Salehi**, Varieties of Tree Languages
65. **Jukka Arvo**, Efficient Algorithms for Hardware-Accelerated Shadow Computation
66. **Mika Hirvikorpi**, On the Tactical Level Production Planning in Flexible Manufacturing Systems
67. **Adrian Costea**, Computational Intelligence Methods for Quantitative Data Mining
68. **Cristina Seceleanu**, A Methodology for Constructing Correct Reactive Systems
69. **Luigia Petre**, Modeling with Action Systems
70. **Lu Yan**, Systematic Design of Ubiquitous Systems
71. **Mehran Gomari**, On the Generalization Ability of Bayesian Neural Networks
72. **Ville Harkke**, Knowledge Freedom for Medical Professionals – An Evaluation Study of a Mobile Information System for Physicians in Finland
73. **Marius Cosmin Codrea**, Pattern Analysis of Chlorophyll Fluorescence Signals
74. **Aiying Rong**, Cogeneration Planning Under the Deregulated Power Market and Emissions Trading Scheme
75. **Chihab BenMoussa**, Supporting the Sales Force through Mobile Information and Communication Technologies: Focusing on the Pharmaceutical Sales Force
76. **Jussi Salmi**, Improving Data Analysis in Proteomics
77. **Orieta Celiku**, Mechanized Reasoning for Dually-Nondeterministic and Probabilistic Programs
78. **Kaj-Mikael Björk**, Supply Chain Efficiency with Some Forest Industry Improvements
79. **Viorel Preoteasa**, Program Variables – The Core of Mechanical Reasoning about Imperative Programs
80. **Jonne Poikonen**, Absolute Value Extraction and Order Statistic Filtering for a Mixed-Mode Array Image Processor
81. **Luka Milovanov**, Agile Software Development in an Academic Environment
82. **Francisco Augusto Alcaraz Garcia**, Real Options, Default Risk and Soft Applications
83. **Kai K. Kimppa**, Problems with the Justification of Intellectual Property Rights in Relation to Software and Other Digitally Distributable Media
84. **Dragoş Truşcan**, Model Driven Development of Programmable Architectures
85. **Eugen Czeizler**, The Inverse Neighborhood Problem and Applications of Welch Sets in Automata Theory

86. **Sanna Ranto**, Identifying and Locating-Dominating Codes in Binary Hamming Spaces
87. **Tuomas Hakkarainen**, On the Computation of the Class Numbers of Real Abelian Fields
88. **Elena Czeizler**, Intricacies of Word Equations
89. **Marcus Alanen**, A Metamodeling Framework for Software Engineering
90. **Filip Ginter**, Towards Information Extraction in the Biomedical Domain: Methods and Resources
91. **Jarkko Paavola**, Signature Ensembles and Receiver Structures for Oversaturated Synchronous DS-CDMA Systems
92. **Arho Virkki**, The Human Respiratory System: Modelling, Analysis and Control
93. **Olli Luoma**, Efficient Methods for Storing and Querying XML Data with Relational Databases
94. **Dubravka Ilić**, Formal Reasoning about Dependability in Model-Driven Development
95. **Kim Solin**, Abstract Algebra of Program Refinement
96. **Tomi Westerlund**, Time Aware Modelling and Analysis of Systems-on-Chip
97. **Kalle Saari**, On the Frequency and Periodicity of Infinite Words
98. **Tomi Kärki**, Similarity Relations on Words: Relational Codes and Periods
99. **Markus M. Mäkelä**, Essays on Software Product Development: A Strategic Management Viewpoint
100. **Roope Vehkalahti**, Class Field Theoretic Methods in the Design of Lattice Signal Constellations
101. **Anne-Maria Ernvall-Hytönen**, On Short Exponential Sums Involving Fourier Coefficients of Holomorphic Cusp Forms
102. **Chang Li**, Parallelism and Complexity in Gene Assembly
103. **Tapio Pahikkala**, New Kernel Functions and Learning Methods for Text and Data Mining
104. **Denis Shestakov**, Search Interfaces on the Web: Querying and Characterizing
105. **Sampo Pyysalo**, A Dependency Parsing Approach to Biomedical Text Mining
106. **Anna Sell**, Mobile Digital Calendars in Knowledge Work
107. **Dorina Marghescu**, Evaluating Multidimensional Visualization Techniques in Data Mining Tasks
108. **Tero Säntti**, A Co-Processor Approach for Efficient Java Execution in Embedded Systems
109. **Kari Salonen**, Setup Optimization in High-Mix Surface Mount PCB Assembly
110. **Pontus Boström**, Formal Design and Verification of Systems Using Domain-Specific Languages
111. **Camilla J. Hollanti**, Order-Theoretic Mehtods for Space-Time Coding: Symmetric and Asymmetric Designs
112. **Heidi Himmanen**, On Transmission System Design for Wireless Broadcasting
113. **Sébastien Lafond**, Simulation of Embedded Systems for Energy Consumption Estimation
114. **Evgeni Tsivtsivadze**, Learning Preferences with Kernel-Based Methods
115. **Petri Salmela**, On Commutation and Conjugacy of Rational Languages and the Fixed Point Method
116. **Siamak Taati**, Conservation Laws in Cellular Automata
117. **Vladimir Rogojin**, Gene Assembly in Stichotrichous Ciliates: Elementary Operations, Parallelism and Computation
118. **Alexey Dudkov**, Chip and Signature Interleaving in DS CDMA Systems
119. **Janne Savela**, Role of Selected Spectral Attributes in the Perception of Synthetic Vowels
120. **Kristian Nybom**, Low-Density Parity-Check Codes for Wireless Datacast Networks
121. **Johanna Tuominen**, Formal Power Analysis of Systems-on-Chip
122. **Teijo Lehtonen**, On Fault Tolerance Methods for Networks-on-Chip
123. **Eeva Suvitie**, On Inner Products Involving Holomorphic Cusp Forms and Maass Forms
124. **Linda Mannila**, Teaching Mathematics and Programming – New Approaches with Empirical Evaluation
125. **Hanna Suominen**, Machine Learning and Clinical Text: Supporting Health Information Flow
126. **Tuomo Saarni**, Segmental Durations of Speech
127. **Johannes Eriksson**, Tool-Supported Invariant-Based Programming

128. **Tero Jokela**, Design and Analysis of Forward Error Control Coding and Signaling for Guaranteeing QoS in Wireless Broadcast Systems
129. **Ville Lukkarila**, On Undecidable Dynamical Properties of Reversible One-Dimensional Cellular Automata
130. **Qaisar Ahmad Malik**, Combining Model-Based Testing and Stepwise Formal Development
131. **Mikko-Jussi Laakso**, Promoting Programming Learning: Engagement, Automatic Assessment with Immediate Feedback in Visualizations
132. **Riikka Vuokko**, A Practice Perspective on Organizational Implementation of Information Technology
133. **Jeanette Heidenberg**, Towards Increased Productivity and Quality in Software Development Using Agile, Lean and Collaborative Approaches
134. **Yong Liu**, Solving the Puzzle of Mobile Learning Adoption
135. **Stina Ojala**, Towards an Integrative Information Society: Studies on Individuality in Speech and Sign
136. **Matteo Brunelli**, Some Advances in Mathematical Models for Preference Relations
137. **Ville Junnila**, On Identifying and Locating-Dominating Codes
138. **Andrzej Mizera**, Methods for Construction and Analysis of Computational Models in Systems Biology. Applications to the Modelling of the Heat Shock Response and the Self-Assembly of Intermediate Filaments.
139. **Csaba Ráduly-Baka**, Algorithmic Solutions for Combinatorial Problems in Resource Management of Manufacturing Environments
140. **Jari Kyngäs**, Solving Challenging Real-World Scheduling Problems
141. **Arho Suominen**, Notes on Emerging Technologies
142. **József Mezei**, A Quantitative View on Fuzzy Numbers
143. **Marta Olszewska**, On the Impact of Rigorous Approaches on the Quality of Development
144. **Antti Airola**, Kernel-Based Ranking: Methods for Learning and Performace Estimation
145. **Aleksi Saarela**, Word Equations and Related Topics: Independence, Decidability and Characterizations
146. **Lasse Bergroth**, Kahden merkkijonon pisimmän yhteisen alijonon ongelma ja sen ratkaiseminen
147. **Thomas Canhao Xu**, Hardware/Software Co-Design for Multicore Architectures
148. **Tuomas Mäkilä**, Software Development Process Modeling – Developers Perspective to Contemporary Modeling Techniques
149. **Shahrokh Nikou**, Opening the Black-Box of IT Artifacts: Looking into Mobile Service Characteristics and Individual Perception
150. **Alessandro Buoni**, Fraud Detection in the Banking Sector: A Multi-Agent Approach
151. **Mats Neovius**, Trustworthy Context Dependency in Ubiquitous Systems
152. **Fredrik Degerlund**, Scheduling of Guarded Command Based Models
153. **Amir-Mohammad Rahmani-Sane**, Exploration and Design of Power-Efficient Networked Many-Core Systems
154. **Ville Rantala**, On Dynamic Monitoring Methods for Networks-on-Chip
155. **Mikko Pelto**, On Identifying and Locating-Dominating Codes in the Infinite King Grid
156. **Anton Tarasyuk**, Formal Development and Quantitative Verification of Dependable Systems
157. **Muhammad Mohsin Saleemi**, Towards Combining Interactive Mobile TV and Smart Spaces: Architectures, Tools and Application Development
158. **Tommi J. M. Lehtinen**, Numbers and Languages
159. **Peter Sarlin**, Mapping Financial Stability
160. **Alexander Wei Yin**, On Energy Efficient Computing Platforms
161. **Mikołaj Olszewski**, Scaling Up Stepwise Feature Introduction to Construction of Large Software Systems
162. **Maryam Kamali**, Reusable Formal Architectures for Networked Systems
163. **Zhiyuan Yao**, Visual Customer Segmentation and Behavior Analysis – A SOM-Based Approach
164. **Timo Jolivet**, Combinatorics of Pisot Substitutions
165. **Rajeev Kumar Kanth**, Analysis and Life Cycle Assessment of Printed Antennas for Sustainable Wireless Systems
166. **Khalid Latif**, Design Space Exploration for MPSoC Architectures

167. **Bo Yang**, Towards Optimal Application Mapping for Energy-Efficient Many-Core Platforms
168. **Ali Hanzala Khan**, Consistency of UML Based Designs Using Ontology Reasoners
169. **Sonja Leskinen**, m-Equine: IS Support for the Horse Industry
170. **Fareed Ahmed Jokhio**, Video Transcoding in a Distributed Cloud Computing Environment
171. **Moazzam Fareed Niazi**, A Model-Based Development and Verification Framework for Distributed System-on-Chip Architecture
172. **Mari Huova**, Combinatorics on Words: New Aspects on Avoidability, Defect Effect, Equations and Palindromes
173. **Ville Timonen**, Scalable Algorithms for Height Field Illumination
174. **Henri Korvela**, Virtual Communities – A Virtual Treasure Trove for End-User Developers
175. **Kameswar Rao Vaddina**, Thermal-Aware Networked Many-Core Systems
176. **Janne Lahtiranta**, New and Emerging Challenges of the ICT-Mediated Health and Well-Being Services
177. **Irum Rauf**, Design and Validation of Stateful Composite RESTful Web Services
178. **Jari Björne**, Biomedical Event Extraction with Machine Learning
179. **Katri Haverinen**, Natural Language Processing Resources for Finnish: Corpus Development in the General and Clinical Domains
180. **Ville Salo**, Subshifts with Simple Cellular Automata
181. **Johan Ersfolk**, Scheduling Dynamic Dataflow Graphs
182. **Hongyan Liu**, On Advancing Business Intelligence in the Electricity Retail Market
183. **Adnan Ashraf**, Cost-Efficient Virtual Machine Management: Provisioning, Admission Control, and Consolidation
184. **Muhammad Nazrul Islam**, Design and Evaluation of Web Interface Signs to Improve Web Usability: A Semiotic Framework
185. **Johannes Tuikkala**, Algorithmic Techniques in Gene Expression Processing: From Imputation to Visualization
186. **Natalia Díaz Rodríguez**, Semantic and Fuzzy Modelling for Human Behaviour Recognition in Smart Spaces. A Case Study on Ambient Assisted Living
187. **Mikko Pänkäälä**, Potential and Challenges of Analog Reconfigurable Computation in Modern and Future CMOS
188. **Sami Hyrynsalmi**, Letters from the War of Ecosystems – An Analysis of Independent Software Vendors in Mobile Application Marketplaces
189. **Seppo Pulkkinen**, Efficient Optimization Algorithms for Nonlinear Data Analysis
190. **Sami Pyöttiälä**, Optimization and Measuring Techniques for Collect-and-Place Machines in Printed Circuit Board Industry
191. **Syed Mohammad Asad Hassan Jafri**, Virtual Runtime Application Partitions for Resource Management in Massively Parallel Architectures
192. **Toni Ernvall**, On Distributed Storage Codes
193. **Yuliya Prokhorova**, Rigorous Development of Safety-Critical Systems
194. **Olli Lahdenoja**, Local Binary Patterns in Focal-Plane Processing – Analysis and Applications
195. **Annika H. Holmbom**, Visual Analytics for Behavioral and Niche Market Segmentation
196. **Sergey Ostroumov**, Agent-Based Management System for Many-Core Platforms: Rigorous Design and Efficient Implementation
197. **Espen Suenson**, How Computer Programmers Work – Understanding Software Development in Practise
198. **Tuomas Poikela**, Readout Architectures for Hybrid Pixel Detector Readout Chips
199. **Bogdan Iancu**, Quantitative Refinement of Reaction-Based Biomodels
200. **Ilkka Törmä**, Structural and Computational Existence Results for Multidimensional Subshifts
201. **Sebastian Okser**, Scalable Feature Selection Applications for Genome-Wide Association Studies of Complex Diseases
202. **Fredrik Abbors**, Model-Based Testing of Software Systems: Functionality and Performance
203. **Inna Pereverzeva**, Formal Development of Resilient Distributed Systems
204. **Mikhail Barash**, Defining Contexts in Context-Free Grammars
205. **Sepinoud Azimi**, Computational Models for and from Biology: Simple Gene Assembly and Reaction Systems
206. **Petter Sandvik**, Formal Modelling for Digital Media Distribution

207. **Jongyun Moon**, Hydrogen Sensor Application of Anodic Titanium Oxide Nanostructures
208. **Simon Holmbacka**, Energy Aware Software for Many-Core Systems
209. **Charalampos Zinoviadis**, Hierarchy and Expansiveness in Two-Dimensional Subshifts of Finite Type
210. **Mika Murtojärvi**, Efficient Algorithms for Coastal Geographic Problems
211. **Sami Mäkelä**, Cohesion Metrics for Improving Software Quality
212. **Eyal Eshet**, Examining Human-Centered Design Practice in the Mobile Apps Era
213. **Jetro Vesti**, Rich Words and Balanced Words
214. **Jarkko Peltomäki**, Privileged Words and Sturmian Words
215. **Fahimeh Farahnakian**, Energy and Performance Management of Virtual Machines: Provisioning, Placement and Consolidation
216. **Diana-Elena Gratie**, Refinement of Biomodels Using Petri Nets
217. **Harri Merisaari**, Algorithmic Analysis Techniques for Molecular Imaging
218. **Stefan Grönroos**, Efficient and Low-Cost Software Defined Radio on Commodity Hardware
219. **Noora Nieminen**, Garbling Schemes and Applications
220. **Ville Taajamaa**, O-CDIO: Engineering Education Framework with Embedded Design Thinking Methods
221. **Johannes Holvitie**, Technical Debt in Software Development – Examining Premises and Overcoming Implementation for Efficient Management
222. **Tewodros Deneke**, Proactive Management of Video Transcoding Services
223. **Kashif Javed**, Model-Driven Development and Verification of Fault Tolerant Systems
224. **Pekka Naula**, Sparse Predictive Modeling – A Cost-Effective Perspective
225. **Antti Hakkala**, On Security and Privacy for Networked Information Society – Observations and Solutions for Security Engineering and Trust Building in Advanced Societal Processes
226. **Anne-Maarit Majanoja**, Selective Outsourcing in Global IT Services – Operational Level Challenges and Opportunities
227. **Samuel Rönnqvist**, Knowledge-Lean Text Mining
228. **Mohammad-Hashem Hahgbayan**, Energy-Efficient and Reliable Computing in Dark Silicon Era
229. **Charmi Panchal**, Qualitative Methods for Modeling Biochemical Systems and Datasets: The Logicome and the Reaction Systems Approaches
230. **Erkki Kaila**, Utilizing Educational Technology in Computer Science and Programming Courses: Theory and Practice
231. **Fredrik Robertsén**, The Lattice Boltzmann Method, a Petaflop and Beyond
232. **Jonne Pohjankukka**, Machine Learning Approaches for Natural Resource Data
233. **Paavo Nevalainen**, Geometric Data Understanding: Deriving Case-Specific Features
234. **Michal Szabados**, An Algebraic Approach to Nivat's Conjecture
235. **Tuan Nguyen Gia**, Design for Energy-Efficient and Reliable Fog-Assisted Healthcare IoT Systems
236. **Anil Kanduri**, Adaptive Knobs for Resource Efficient Computing
237. **Veronika Suni**, Computational Methods and Tools for Protein Phosphorylation Analysis
238. **Behailu Negash**, Interoperating Networked Embedded Systems to Compose the Web of Things
239. **Kalle Rindell**, Development of Secure Software: Rationale, Standards and Practices
240. **Jurka Rahikkala**, On Top Management Support for Software Cost Estimation

Turku Centre *for* Computer Science

http://www.tucs.fi
tucs@abo.fi

University of Turku
Faculty of Science and Engineering
- Department of Future Technologies
- Department of Mathematics and Statistics

Turku School of Economics
- Institute of Information Systems Science

Åbo Akademi University
Faculty of Science and Engineering
- Computer Engineering
- Computer Science

Faculty of Social Sciences, Business and Economics
- Information Systems

ISBN 978-952-12-3835-2
ISSN 1239-1883

Jurka Rahikkala

On Top Management Support for Software Cost Estimation

Jurka Rahikkala

On Top Management Support for Software Cost Estimation