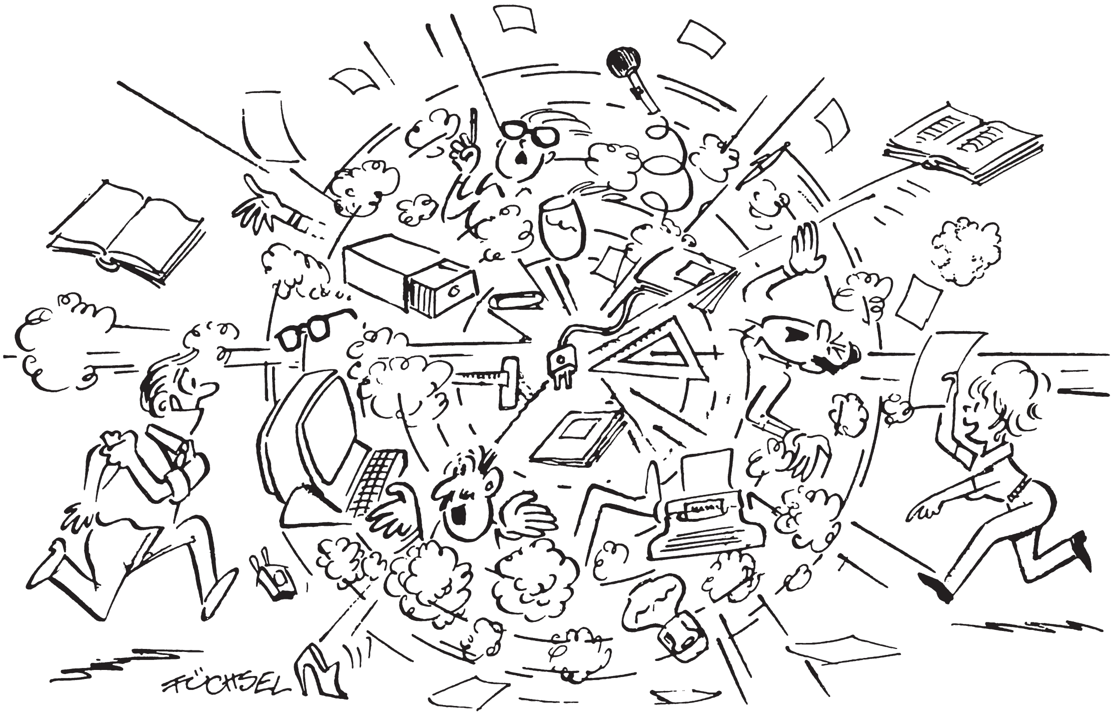
JYVÄSKYLÄ STUDIES IN COMPUTING

136

Paula Savolainen

Why Do Software Development Projects Fail?

Emphasising the Supplier's Perspective and the Project Start-Up



JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 136

Paula Savolainen

Why Do Software Development Projects Fail?

Emphasising the Supplier's Perspective and the Project Start-Up

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen Auditoriossa 3
marraskuun 18. päivänä 2011 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in the building Agora, Auditorium 3, on November 18, 2011 at 12 o'clock noon.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2011

Why Do Software Development Projects Fail?

Emphasising the Supplier's Perspective and the Project Start-Up

JYVÄSKYLÄ STUDIES IN COMPUTING 136

Paula Savolainen

Why Do Software Development Projects Fail?

Emphasising the Supplier's Perspective and the Project Start-Up

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2011

Editors
Seppo Puuronen
Department of Computer Science and Information Systems, University of Jyväskylä
Pekka Olsbo, Ville Korkiakangas
Publishing Unit, University Library of Jyväskylä

Cover picture by F. Füchel, printed with
permission by its owner, Morten Fangel

URN:ISBN:9789513944681
ISBN 978-951-39-4468-1 (PDF)

ISBN 978-951-39-4467-4 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2011

ABSTRACT

Savolainen, Paula
Why Do Software Development Projects Fail? Emphasising the Supplier's Perspective and the Project Start-Up
Jyväskylä: University of Jyväskylä, 2011, 81 p. (+ articles)
(Jyväskylä Studies in Computing,
ISSN 1456-5390; 136)
ISBN 978-951-39-4467-4 (nid.)
ISBN 978-951-39-4468-1 (PDF)
Finnish summary
Diss.

This dissertation focus on finding answers to the question: why do software development projects fail? A post-mortem analysis of five cancelled software engineering projects was conducted. From this, the reasons for cancellation decisions and the causes behind those reasons were identified. In four cases there was a sub-contracting relationship between a customer and a supplier, and in these cases serious mistakes were made before the project start. As a result of this study, the customer-supplier framework was developed. This discerns differences between a customer and a supplier in sub-contracting situations and positions the phases which precede software development projects in the business context. A systematic literature review was also performed in order to find out how software development project success and failure is defined from the supplier's perspective. Three success criteria from the supplier's perspective were identified: customer satisfaction, short term business success for the supplier, and long-term success for the supplier. Furthermore, this research established that the academic literature does not distinguish between software development project management success and project success. Without this distinction it is easy to conclude that a software development project has failed. Following the systematic literature, the project start-up phase was researched in industry and it was found that software development projects may fail because supplier firms do not seem to actively aim at defining success criteria for each project during the start-up phase. Without defined success criteria different members may pursue their own goals instead of the project's goals. Therefore, it is difficult to make project management adjustments during project execution which ensure project success and evaluation of project success is not possible.

Keywords: software development project, project failure, project success, project management failure, project management success, project start-up, customer-supplier framework

Author's address

M.Sc.Econ. Paula Savolainen
School of Computing
University of Eastern Finland
paula.m.savolainen@uef.fi

Supervisors

Dr. Ita Richardson
Lero - the Irish Software Engineering Research Centre
Depart. of Computer Science & Information Systems
University of Limerick, Ireland

Professor Jarmo Ahonen
School of Computing
University of Eastern Finland

Professor Pasi Tyrväinen
Depart. of Computer Science and Information Systems
University of Jyväskylä, Finland

Reviewers

Professor Hannu Jaakkola
Tampere University of Technology

Dr. Rory O'Connor
Dublin City University

Opponent

Associate Professor Darja Šmite
University of Latvia

PREFACE

Working for a doctoral dissertation is a huge process lasting many years, and that was very true for me. However, working for my Ph.D was definitely something else for me than that which the public outlook recommends: young graduates without any other working experience are encouraged to conduct research for their Ph.D as fast as possible and to continue on their academic career paths.

Working as a researcher was not an option for me after graduation. I had to wait until Professor Anne Eerola tossed me her suggestion, and Dr. Seppo Lammi, head of the department at the University of Kuopio, offered me – a middle-aged woman – a one-year contract. That was a pre-start but the real start was an opportunity to visit Lero – the Irish Software Engineering Research Centre. After several fruitful months in Ireland everything has gone so fast, but I don't know yet whether I have found the first control point in a long distance or a sprint race.

This dissertation is the result of hard work; combining work, private life and research together. Therefore I especially want to thank my family and friends who have suffered my mental absence and desire to talk about how I have progressed. In addition to my dearest, I thank all my supervisors, but especially Ita: *Go raibh mile maith agat, Ita, as an gcabhair agus as an tacaiocht go léir a thug do dom.*

I express my gratitude to my reviewers, Professor Hannu Jaakkola and Dr. Rory O'Connor, for their valuable comments. Special thanks go to Morten Fangel, who kindly lent his own book to me. Finally, I want to thank my colleagues in Finland and abroad for supporting and encouraging me in my research.

The research reported in this dissertation has been supported in part by the Finnish Funding Agency for Technology and Innovation (Tekes) with grant 70011/08 (SAMeT) and 70054/10 (Kytkin), and was supported in part by Science Foundation Ireland grant 03/CE2/I303_1 to Lero – the Irish Software Engineering Research Centre. I thank the Saastamoinen Foundation (Saastamoisen säätiö) for the opportunity to go abroad; this Primrose path would be one route to be chosen.

Father, I'm sure you would have been proud of me.

Siilinjärvi 10.10.2011
Paula Savolainen

FIGURES

TABLE

CONTENTS

1 INTRODUCTION

Software development has its roots in the 1960s and since then software has become an essential part of modern society. With the help of software it is possible to write a report, build a bridge, control the space shuttle, build social networks in the Internet, and it may play a crucial role in overall economy. However, despite the successful application of software to almost all possible areas, software development projects have a reputation that they often fail; they are e.g. late, over-budget, or not able to satisfy customers' needs (Cerpa and Verner, 2009; Glass, 2006).

Although software development projects have a reputation that they fail, it is difficult to find trustworthy figures of the economic impact of software development project failures, and even the percentage of cancelled projects is not clear (Glass, 2005; 2006). For example Charette (2005), after several decades as an IT consultant, estimates that project failures have a likely cost the U.S. economy of at least US$25 billion and maybe US$75 billion during the last five years. He defines project failure as the total abandonment of a project before or shortly after delivery, and therefore his figures do not include projects which exceeded their budgets, were delivered late, or costs of rework of bug-ridden systems. Assuming that Charette's estimates are correct, the number of cancellations is daunting, and their economic impact is significant.

The inaccuracy of the rate of the economic impact of software development project failure is said to be due to the situation that most of the software failures, especially in private-sector organizations, remain unreported and are written off as part of the cost of doing business (Ewusi-Mensah, 2003, p. 4). The situation is very understandable because neither the organizations nor the individuals involved want the details of those projects to appear in any media, and as a consequence, research on these projects is difficult.

However, there have been studies on unsuccessful software development projects. The studied cases have been either too massive to be hidden or have been public in some legal sense. Because of the publicity these cases have been thoroughly studied in order to find reasons for failure, failure factors, or risks related to these cases (e.g. Charette, 2005; Chua, 2009; Ewusi-Mensah, 2003;

Glass, 1998; Verner, Sampson and Cerpa, 2008). In addition to studies on publicly available cases, there are other studies which have tried to find signs or patterns which could foretell or ensure software development project success or improve efficiency of numerous projects (e.g. Bannerman, 2008; Barki, Rivard and Talbot, 2001; Barry, Mukhopadhyay and Slaughter, 2002; Boehm, 1991; de Bakker, Boonstra and Wortmann, 2010; Dvir, Raz and Shenhar, 2003; Schmidt, Lyytinen, Keil and Cule, 2001; Verner and Evanco, 2005).

Those studies on failed software projects which have identified various factors that lead to failure have usually done this in a rather generalized fashion, and the reason for the failure of a specific project is often ignored. It is rather the effect of the failure that is described in detail, e.g., the failure of project X resulted in a loss of $Y dollars and as a consequence the company went bankrupt (Verner et al., 2008).

Therefore, although software development projects have been carried out for over 50 years, it seems that we have not yet learned enough to ensure that our software development projects are successful (Cerpa and Verner, 2009), and there is still a need for research on software development failure. In addition to the learning point of view, the need for research on software development failure is intensified by the human factors; a failed software development project means despair and embarrassment for the project team, and may ruin careers. Moreover, software development is difficult and expensive, and therefore software development project failure means loss of economic resources.

Hence, in this dissertation, the focus is on software development project failure. Instead of research on large and massive projects, the research for this dissertation concentrates on private-sector everyday projects that are not high-profile enough to be reported in the literature or press. The aim of the research is to extend understanding of software development project failure and have answers to question why software development projects fail.

This dissertation consists of this introduction (Chapter 1), 8 further chapters, and five publications. A literature review and the basic concepts are introduced in Chapter 2.

The research process is presented in Chapters 3 and 4. The overall research problem, research questions, and their relationships are presented in Chapter 3. This chapter emphasizes the decisions made during the research process. It also details the original publications co-authored by the researcher presenting this dissertation and how each one contributes to the dissertation. Chapter 4 discusses the research methodologies which were used during the research process.

Chapter 5 presents results from the post-mortem analysis.

Chapter 6 describes the systematic literature review which was performed in order to have an answer to the first research question. Identified software development project success criteria from the supplier’s perspective are presented and evaluated.

In Chapter 8 there is an overview of the project start-up phase. Moreover, activities which are performed in a firm which develops software for an external customer are discussed. Furthermore, a three-level understanding which should be gained before any software project starts is presented.

Chapter 9 summarizes the research. This includes an assessment of the research, and the implications for practice and future research.

The original publications are listed in Appendix A including full citation information and abstract. The author's contribution to each publication is also presented. Each publication is referred in this dissertation using name 'Publication' and serial Roman number. Each article is unique and discrete. However, the publications along with Chapters 1-9 form a coherent dissertation.

2 BASIC CONCEPTS ON SOFTWARE DEVELOPMENT PROJECT FAILURE

Although software development projects have been carried out for over 50 years, it seems that we have not yet learned enough to ensure that our software development projects are successful (Cerpa and Verner, 2009). However, during recent decades different ways of how to organize software development have been developed (Cusumano, MacCormack, Kemerer and Crandall, 2003; Gumm, 2006; Meyer, 2006; Prikladnicki, Audy and Shull, 2010), and terminology which is used to describe software development has also evolved. Because of different terminology which has been used to describe different modes to organize software development, research on software development projects is difficult without being strict with concepts and what they mean.

One of the difficulties related to research on software development projects is to recognize whether the study is about the software development project or something else. Moreover, core questions are whether the study is about a project, about software development, or about software development project.

The first concept, 'project', is defined in the standard ISO/IEC 12207 Systems and software engineering – Software life cycle processes as "*an endeavour with defined start and finish dates undertaken to create a product or service in accordance with specified resources and requirements*" (ISO/IEC 12207, 2008, p. 5). Moreover, Newton defines a project describing attributes for work done during a project: "*In a project the work is unique, complex, non-routine, on-time effort limited by time, budget, resources, and performance specifications designed to meet customer needs*" (Newton, 2009, p. 16). Therefore a project should have defined start and finish dates and specific resources should be allocated to do unique and complex work. After accomplishment, project resources will be relieved to other projects or work.

The second concept, 'software development' is as important as the concept of project, and it is understood in this dissertation as the act of working to produce or create software. This means that studies on IT projects may be applicable if they contain software development projects but there may be difficulties in isolating results which discuss only software development projects. One example of such difficulties can be found in a study by Whittaker (1999).

Whittaker's study is about unsuccessful information technology projects. Projects were classified as 'Custom-developed Application', 'Purchased Application Installation', 'New Data Management System', 'New Operating System', 'New Communication System', and 'Other Components'. Without having more knowledge about the projects studied it seems that only the first project type, 'Custom-developed Application', is about software development projects. Therefore only parts of the results attained by Whittaker are applicable if the research focus is on software development projects.

Another difficulty is to distinguish software development projects from software maintenance work which is done on a continuous basis, i.e. without bundling changes or new features together and establishing a project for that. Moreover, confusion exists with continuous services which are related to software but are not software development, e.g. database administrative support and network support. Furthermore, one difficulty is related to software products and their implementation in target organizations. Can a software product implementation project be considered a software development project or not?

Examples of studies where it is difficult to interpret if the focus is on software development project are studies made by Haried and Ramamurthy (2009), Levina and Ross (2003), and Taylor (2005; 2007). These studies are concrete examples about how carefully researchers have to read studies made by other researchers, look for symptoms of projects, and evaluate the usefulness and applicability of the studies from the viewpoint of a software development project.

One example is the study by Haried and Ramamurthy (2009) which evaluates the success in international sourcing of information technology projects. The projects consist of different IT projects – including software development projects – and therefore only part of the study may be applicable. However, in addition to being careful when recognizing software development projects, researchers should recognize that in this study three cases out of eight were actually continuous services (database administrative support, SAP application maintenance support, and network support) – not projects. Therefore careful consideration is needed before using the results gained from this study.

Another example is the study by Levina and Ross (2003) which explores the value proposition in information technology outsourcing. The study consists of one case where contracts are made for several years with a fixed price, with certain value, and an agreement on the level of service to be provided for that price. However, it seems that the contract is a mixture of support, software maintenance and various development projects, and therefore the study is not applicable to research on software development projects.

Although Levina and Ross's study is not applicable for research on software development projects, it is a study of application management outsourcing. As was the case in their study, outsourcing may be a fixed contract for several years including agreed services. On the contrary, it may be that a study of outsourcing may be revealed to be a case where only one project is outsourced to one supplier and therefore the study of outsourcing may be applicable to research on software development projects.

The question of should a software product implementation project be considered a software development project in context of outsourcing is concretized with another example. Taylor (2005; 2007) studied outsourced IT projects from the supplier's perspective and especially risks on package implementation projects. Typically the projects in her study included extensive customization, front-end web development work, and/or major infrastructure upgrades, and only two major projects were described as straightforward package implementations with little or no customization (Taylor, 2005). Therefore the study made by Taylor can be considered applicable to research on software development projects.

These three examples describe the difficulties which researchers face while studying software development projects. In this dissertation the focus is on projects in which software is designed or developed, or major customization is made in software products. Embedded software development is also included. The concept of a software engineering project is used to highlight that some of the projects can be embedded software development projects including appliance design. Any IT projects or continuous services are excluded from this dissertation.

Before it is possible to assess a software development project as a failure, the concept of 'project failure' should be considered. Although software development project failure could be assumed to be intuitively understandable, there is neither a consistent definition of software development project failure or project failure in general.

However, there are two concepts related to project failure/success which are agreed in the project management literature (Belassi and Tukel, 1996; Collins and Baccarini, 2004; de Wit, 1988; Jugdev and Müller, 2005; Munns and Bjeirmi, 1996; Müller and Turner, 2007; Wateridge, 1998):

- Project failure/success factors which are elements of a project that can be influenced to increase the likelihood of failure/success; these are independent variables that make failure/success more likely
- Project failure/success criteria which are the elements by which the outcome of the project is judged; these are dependent variables which measure project failure/success

Project failure/success factors are discussed as such (e.g. Cerpa and Verner, 2009; de Wit, 1988; Munns and Bjeirmi, 1996; Reel, 1999), grouped by process, product, or people factors (e.g. Procaccino, Verner and Lorenzet, 2006), a framework has been used in order to classify the factors (e.g. Belassi and Tukel, 1996; Ewusi-Mensah, 2003; Fortune and White, 2006), or factors have been used as a basis for models to estimate software project outcome (Cerpa, Bardeen, Kitchenham and Verner, 2010).

Instead of project failure/success factors, more interesting from this dissertation's point of view are the criteria which are used to judge the project's failure or success. However, there is no common understanding of either what software development project failure/success criteria are or what project failure/success criteria are in general, and the main reason for the prevailing situation seems to originate from the nature of projects. Each project has different interest groups or stakeholders which view the project from different perspectives (Belassi and Tukel, 1996; Boehm, 1991; de Wit, 1988), and this may be the reason for the inconsistent and diverse criteria which are used. Commonly recognized stakeholders in software development projects are project sponsors, customer, end-users, senior management, suppliers, project managers, and the actual project team which designs and develops software (Boehm, 1991; Jun, Qiuzhen and Qingguo, 2010; Pressman, 2005; Procaccino, Verner, Shelfer and Gefen, 2005; Procaccino and Verner, 2006). For example, developers' perspective is emphasized in (Linberg, 1999; Procaccino et al., 2005) and project managers' perspective in (e.g. Procaccino and Verner, 2006).

Although there is no common understanding of project failure/success criteria between researchers, general project management research has started to distinguish concepts of 'project failure/success' and 'project management failure/success' (PM failure/success). Two recently published reviews on project success, one by Jugdev and Müller (2005) and another by Ika (2009), highlight the distinction between project success and PM success. Moreover, Papke-Shields, Beise and Quan (2010) have noticed and taken into account this distinction while defining measures for their study on assessing the use of project management practices. Other studies differentiating project success from PM success are e.g. Baccarini (1999), Cooke-Davies (2002), de Wit (1988), Dvir, Lipovetsky, Shenhar and Thisler (1998), Dvir et al. (2003), Lipovetsky, Tishler, Dvir and Shenhar (1997), Munns and Bjeirmi (1996), Sadeh, Dvir and Shenhar (2000), and Shenhar, Levy and Dvir (1997). The same distinction is made by Pinto and Prescott (1990), and Pinto and Mantel (1990), who have used the concepts efficiency of the project implementation process and external efficiency. The first concept refers to PM success whereas external efficiency consists of the perceived value of the project and client satisfaction.

The definitions presented by Munns and Bjeirmi (1996) for project and project management clarify the distinction. They define a project as "*achievement of a specific objective, which involves a series of activities and tasks which consume resources*" (Munns and Bjeirmi, 1996, p. 81). This highlights the importance of understanding and attaining the project goals, and a project is a means to achieving those goals. Project management is defined as "*the process of controlling the achievement of the project objectives by applying a collection of tools and techniques*" (Munns and Bjeirmi, 1996, p. 81). Thus PM success is considered to be measurable (e.g. time/cost/quality) while project success goes further, focusing on longer-term and customer-oriented results (Papke-Shields et al., 2010).

Another way to perceive the distinction between project success and PM success is by saying "*the operation was a success, but the patient died*" (Jugdev and Müller, 2005, p. 22). This emphasizes the situation, where a software development project has been accomplished on time, within budget, and with required features. If the software does not fulfil the customer's real needs the reason for the situation is not a consequence of the managerial part of the project but there are more profound reasons for the situation.

Moreover, de Wit (1988) has said that "*A project can be a success despite poor project management performance.*" (de Wit, 1988, p. 165), and one popular example of this is the Sydney Opera House. Although it took 15 years to build and the budget was overrun 14 times, it is now generally agreed to be an engineering masterpiece (Jugdev and Müller, 2005). However, it should be realized that whereas PM success may lead to project success, the opposite is not true (Ika, 2009), as was also pointed out by de Wit (de Wit, 1988, p. 165): "*Good project management can contribute towards project success but is unlikely to be able to prevent failure*". Therefore, PM failure/success and project failure/success should be evaluated as separate but interlinked measures.

One of the studies where there were criteria for both project and project management failure that were precisely expressed was the study by Whittaker (1999) on unsuccessful information technology projects. A project was considered a failure if the project budget was overrun by 30 per cent or more, or the project schedule was overrun by 30 per cent or more, or the project was cancelled or deferred due to its inability to demonstrate or deliver the planned benefits.

On the contrary, only one criterion was used in the study of Sumner, Bock and Giamartino (2006) exploring the linkage between the characteristics of IT project leaders and project success. They adhered to one project management criterion defining project success as: "*…planned versus actual project completion time is a valid measure of project success that essentially encompasses project cost*" (Sumner et al., 2006, p. 46). Finishing a project within budget may mean not that it has been successful, but only that the management of costs has been successful.

In addition to already mentioned studies, there is one study and one definition which should be mentioned if the focus of the research is software development failure. The study is the periodically updated Chaos Report which is made by the Standish Group[1]. Publicly available and largely cited version of the report is from 1995. However, the results and the content are not cited in this dissertation because of the criticism presented against research methods and figures (Eveleens and Verhoef, 2010; Glass, 2005; Glass, 2006; Jørgensen and Moløkken-Østvold, 2006). In addition to the Chaos Report, the definition which has to be mentioned is made by one of the critics, Robert L. Glass. He uses the

[1] http://www.standishgroup.com

concept of ‘runaway project’ meaning “*a project which goes out of control primarily because of the difficulty of building the software needed by the system*” (Glass, 1998, p. 3). This definition encompasses the recognition that the project became unmanageable in terms of meeting its original target goals, and all too often, the project had to be cancelled. These target goals may be schedule, costs, quality, or other essential goals of the project.

As a summary, project success is defined in this dissertation according to Turner: “*Overall the project will be successful if it delivers the desired performance improvement, or better, at a time and a cost that provides value for the organization*” (Turner, 2009, p. 266). This definition emphasizes project success but does not ignore project management success.

Although both project failure and success are discussed above, the focus of this dissertation is on software development project failure. It is not assumed that success and failure are opposite concepts, but by understanding one concept it is likely that some understanding could be gained about the other concept. Hence, when commencing this dissertation, a failed software development project was defined to be a project that is either cancelled before completion or has delivered functionally that was not used.

However, progression through the research showed that this definition does not cover the supplier’s perspective. The systematic literature review was conducted in order to find how software development project success/failure is defined from the supplier’s perspective. This is presented in Chapter 6.

3 RESEARCH PROBLEM AND RESEARCH QUESTIONS

This chapter presents the research problem, the research questions and the decisions made during the research process. The original publications and their contribution to the main research problem are described. A simplified picture shows the research questions, publications, which are part of this dissertation, and their interrelationships (Figure 1).

In this dissertation the focus is on software development project failure. Motivation for the research originated from observations from the literature but after the first steps of the research it became clear that there is an area which is not researched before. The observations from the literature are as follows: Firstly, it seems that we have not yet learned enough to ensure that our software development projects are successful (Cerpa and Verner, 2009). Secondly, the reason for the failure of a specific project is often ignored in studies on failed software projects which have identified various factors that lead to failure (Verner et al., 2008). Thirdly, comprehensive studies on failed software development projects have been made on large projects which have been too massive to hide, or have been public in some legal sense, but despite of frequency of private-sector everyday projects thorough studies on them have been ignored. Therefore there is a need for robust research on common private-sector software development projects which have deemed to have failed.

Based on these observations the overall research problem of this dissertation can be summarized by the question "*Why do software development projects fail?*" The answer to the research problem was considered by looking at firms which develop software. The projects which were applicable should have failed, e.g. a project that was cancelled before completion or delivered functionally that was not used. The use of this definition ensured that the difference between project failure and PM failure was taken into account.

Five cancelled software engineering projects were found from five different firms. Normally, any type of detailed data on software engineering cancellations is strictly confidential and very difficult to get access to. In these cases one of the authors of Publication I had been involved either as an employee of the software supplier or the customer, or as a consultant hired by a firm which was involved in that case. The concept of software engineering was used to highlight that some of the projects were embedded software development including appliance designing.

In order to achieve a better understanding of why software development projects fail a post-mortem analysis of these five cancelled software engineering projects was made. The aim was to find out why these five projects suffered cancellation, and to gain more knowledge about the reasons for cancellation decisions and the causes behind those reasons. The study is described in Publication I, and the main results from the viewpoint of this dissertation are summarized in Chapter 5.

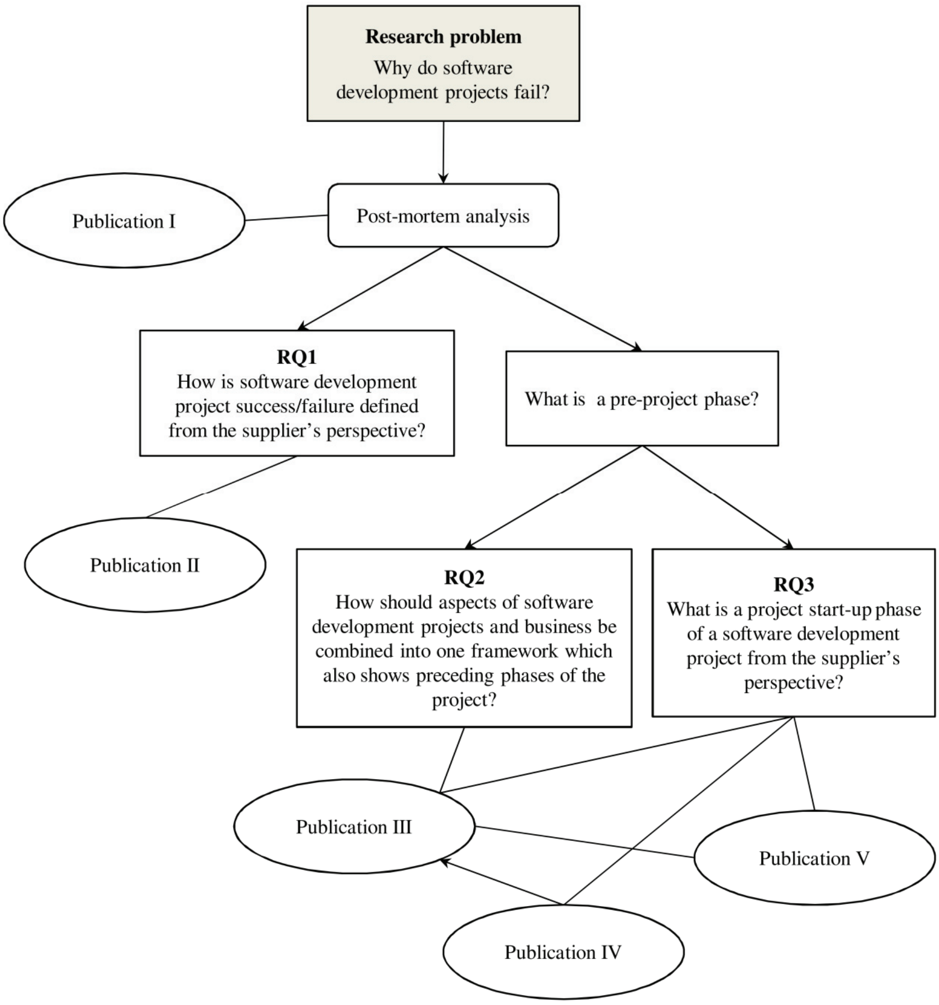

Research problem
Why do software development projects fail?
Publication I
Post-mortem analysis
RQ1
How is software development project success/failure defined from the supplier's perspective?
What is a pre-project phase?
Publication II
RQ2
How should aspects of software development projects and business be combined into one framework which also shows preceding phases of the project?
RQ3
What is a project start-up phase of a software development project from the supplier's perspective?
Publication III
Publication V
Publication IV

FIGURE 1 The research problem, research questions, and their interrelationships

The reasons for cancellation decisions and the causes behind those reasons were identified. The root causes of the cancellation decisions were well known (lack of understanding of the real needs of the customer, the unavailability of experienced people when the project team was selected, risks connected to tight schedules, problems with selected architecture, and a lack of understanding of the technical problem).

However, although reasons for cancellation decisions and the causes behind those reasons were identified, two other issues emerged. The first one is related to concern about the project failure from the perspective of the main stakeholders. Another issue is related to the phase where mistakes were made.

The first issue emerged while analyzing Case C in which there were two organizational stakeholders, a customer and a supplier, and where there was a sub-contracting relationship between these parties. If we look at the case from the customer's perspective the project was a clear failure. However, from the supplier's perspective the case is different. The supplier finished the project practically on time, and the customer paid the invoice although did not implement the system. Was this case also a failure from the supplier's perspective although the project delivered functionally that was not used?

Although different stakeholders have been recognized while studying software development projects, it is not common to distinguish between the customer's and the supplier's perspectives. However, different perspectives are partly taken into account in outsourcing literature, which has traditionally studied outsourcing from the customer's perspective (Dibbern, Goles, Hirschheim and Jayatilaka, 2004), and concentrating on what, why, where, and how to outsource (Dibbern et al., 2004; Hätönen and Eriksson, 2009). Although it has been recognized that there is a need for studies from the supplier's perspective in outsourcing literature (Dibbern et al., 2004; Levina and Ross, 2003), studies have only been published recently which take into account both perspectives equally (e.g. Haried and Ramamurthy, 2009), or the study has been carried out from the supplier's perspective (e.g. Aundhe and Mathew, 2009; Jun et al., 2010; Taylor, 2007).

Hence, when the research problem was studied from the supplier's perspective a more specific question was formulated as "*What is software development project failure from the supplier's perspective?*" Although the overall research problem is about software development project failure it can be supposed that the aim of the supplier is to have a successful project. Because success and failure are not necessary reverse concepts, software development success should also be considered. The first research question is thus defined as

> RQ1: How is software development project success/failure defined from the supplier's perspective?

Another issue emerged while considering the phase in which mistakes were made. In four cases out of five, the outcome of the project was to be delivered to an external customer, and in these cases the cause of the cancellation originated in a phase *before* project had been started. The real cause of the failure in these four cases was a fatal mistake made before the project started, related to the tendering, to the agreement or to something else that happened before the beginning of the project. Hence, in sub-contracting situations there is a pre-project phase in which it is possible to make mistakes resulting in project cancellation.

This finding is supported by Gillis who has stated that many of the important decisions and assumptions that affect the eventual success or failure of a project are made in the 'pre-project' phase and are given to the project manager as part of his mandate or as constraints on his authority (Gillis, 1987). Therefore a tentative research question was formulated as, "*What is a pre-project phase?*".

The study around the pre-project phase was further divided into two more specific research questions. One was related to the problem of how different aspects which are present in sub-contracting software development projects are connected together. Another one was related to a certain phase inside the pre-project phase. The viewpoint followed the same perspective as in RQ1, i.e. the supplier's perspectives but not forgetting the existence of the customer.

When there is a sub-contracting relationship, as it was in four analyzed cases out of five (Publication I), there are two parties, a customer and a supplier, of which the customer is acquiring software and the supplier is developing software for the customer. In these situations, the customer and the supplier are from different organizations, and they have made a contract regarding a software development project. According to the contract the supplier has agreed to develop software and deliver the outcome of the software development project to the customer. Consequently, both parties have different perspectives with diverged goals. For the supplier, the project is a way to do business, and for the customer, the benefits gained with help of the output of the project should be worth the price. Hence, at the same time the aim of the customer is to minimize the costs of the project and aim of the supplier is to maximize the profit of the project (de Wit, 1988; Sadeh et al., 2000). This situation adds business elements into projects.

However, the existence of business elements and that they may have an influence on the following software development project has been noticed only recently. For example Anda, Sjøberg and Mockus (2009) have published an article in which they have recognized that bidding is partly a business domain but at the same time the bids for software projects contain a substantial software engineering component. In addition to recognizing a connection between the business field and software engineering field, Anda et al. (2009) suggest that "*there are important drivers of cost and effort that may not be well described using current software engineering concepts*" (Anda et al., 2009, p. 424).

Given connections between business, software development and projects, there should be a framework which connects all elements together and shows the pre-project phase in this context. Therefore a new research question was formulated:

> RQ2: How should aspects of software development projects and business be combined into one framework which also shows preceding phases of the project?

The answer to RQ2 helps to discern differences between two main organizational stakeholders, consider software development project failure from both perspectives, and position pre-project phase in a business context. During this research project it was found that examining software development projects in a business context is quite unusual. Furthermore, the pre-project phase seemed to be an almost unexplored area, but especially when studying software development projects from the supplier's perspective.

Research question three is related to a certain phase inside the pre-project phase. The post-mortem analysis revealed that the real cause of failure in four contractual cases was a serious mistake, which was made *before* the project had been started. Moreover, in three cases out of these four sub-contracting cases a mistake was made during the tendering phase (Case A, Case C, and Case E) while in the fourth case (Case B) the supplier made a mistake *after* receiving the order from a customer but *before* the project was started. This phase can be considered to be a project start-up phase (Fangel, 1991; Turner and Cochrane, 1993).

During the research process it was decided to concentrate on the project start-up phase. The main reason for the decision was that there is no project until the supplier has received on order from the customer or the customer has indicated in other ways that the supplier has been selected to execute the project. Moreover, the project start-up phase precedes every software development project, and there is a scarcity of studies that discuss project start-up as found during the research process. Hence, the third research question was formulated as

> RQ3: What is a project start-up of a software development project from the supplier's perspective?

In brief, the aim of this dissertation is to have answers to the main research problem, i.e. to have more knowledge about software development project failure and to understand why software development projects fail. However, although reasons for cancellation decisions and the causes behind those reasons were identified in five cases, other issues emerged during the post-mortem analysis, which led to the three research questions. Answers to these three research questions provide partial answers to the main research problem. By summing up all partial answers more knowledge on software development failure is attained.

This dissertation consists of five publications and this introduction. Publication I gives explanations for cancellation of five specific cases and therefore provides a partial answer to the overall research problem. The second publication answers RQ1, and therefore provides a partial answer to the overall research problem. Publications III and IV answer RQ2 and RQ3 and therefore provide further partial answers to the overall research problem. Publication V provides information for RQ3 and therefore is yet another partial answer to the overall research problem.

Each research question is answered in its own chapter. Chapter 6 presents an answer to RQ1, Chapter 7 answers RQ2, and Chapter 8 answers RQ3. Summaries to research questions and answers to the overall research problem are presented in Chapter 9.

4 RESEARCH METHODOLOGIES

The research conducted for this dissertation consists of the overall research problem and three research questions. In order to have a better understanding of the subject – Why do software development projects fail? – a post-mortem analysis of five cancelled software engineering projects was conducted. This method is presented in Subchapter 4.1. In order to have an answer for RQ1 a systematic literature review was performed. This is described in Subchapter 4.2.

Before research questions two and three were formulated there was a more general research question "What is a pre-project phase?" Since it became clear that preceding phases of the software development project are an almost unexplored area, a research approach called theory building from case studies was selected. This research approach is considered especially appropriate in new topic areas (Eisenhardt, 1989b). During the research process RQ2 and RQ3 were formulated. These are described in Subchapter 4.3.

A variety of research methods were used as suggested by Myers (2009, p. 10) and Sjøberg, Dybå and Jørgensen (2007) who claim that carefully selected and combined research methods and design elements in one research is a target for improving the quality of empirical studies in software engineering. Answers to the research questions were attained using the research methods described in this chapter, and therefore more knowledge was gained on software development failure.

4.1 Post-mortem analysis

In order to have an answer to the main research problem, cancelled software development projects were analyzed. Details of the method are given in Publication I; this subchapter provides a brief overview. Analysis and results are also described in Publication I.

A natural choice for the research method in order to analyze projects was a case study, and the choice is supported by the definition of a case study research provided by Myers (2009, p. 76):

> "Case study research in business uses empirical evidence from one or more organizations where an attempt is made to study the subject matter in context. Multiple sources of evidence are used, although most of the evidence comes from interviews and documents."

Case studies are a standard method of empirical study in management and related disciplines such as organization development and information systems research (Sjøberg et al., 2007). Runeson and Höst (2009) state that case study is a suitable research methodology in software engineering research also since it studies phenomena in its natural context, although they see that case study methodology is more mature in social sciences or in the information systems field than in the software engineering field. Hence, the guidelines provided by Yin (2009) – social science –, and Myers (2009) – business and management, although the author's background is from information systems – were applied for post-mortem analysis of software engineering projects.

For post-mortem analysis no separate data collection phase was needed. The data consisted of almost complete project documentation of cancelled software engineering projects in which one of the authors of Publication I had been involved, in either as an employee of the software supplier or the customer, or as a consultant hired by the supplier or the customer. The availability of the detailed data is due to this involvement and to the benevolence of the companies involved. The possible impact of this involvement on the validity of the research is discussed in Publication I.

The data included all existing software engineering process documentation such as technical documents, project plans, minutes of meetings, emails related to the project, and different types of memos. The available documentation included everything that can be reasonably expected to be found after a project has been cancelled. In some cases the documentation includes additional information such as emails released for research purposes.

The unit of analysis was one cancelled software engineering project. Every project was from a different company and each company was involved in only one case, either as a supplier, a customer, or a company with an in-house development unit. Four cases out of five involved both the supplier and the customer companies, and the story of each case is the story of the organizations involved, which is a common situation in case studies of projects (Myers, 2009, p. 76).

Because there is no known post-mortem analysis method developed for a situation where only project documentation – but almost complete project documentation – is available, a step-by-step analysis method was developed. Common knowledge of software projects and process models was needed, and the causal reasoning of the analysis was based on the possibility of following individual events and issues backwards in time by using the reconstructed project.

The analysis of each case was performed in two phases. The first phase consisted of project documentation analysis and interviews carried out by an evaluation team. This phase was conducted for purposes other than academic research and an evaluation report of the cancelled case was produced. During the second phase of the analysis all project documentation was analyzed by the authors of Publication I, including reports created during the first phase.

The analysis carried out in the second phase was based on the available documentation. In this case there was none of the normal difficulties in getting access to the relevant documentation. The available documentation was rich and allowed many different types of analysis to be performed. It was authentic and credible, and there was no doubt of the representativeness of the documents. Since information gathered from analysis of project records and documents has been shown to be invaluable in getting at the 'hidden' agenda that may be at the root of the some of the failed development projects in organizations (Ewusi-Mensah, 2003, p. 199) it was possible to trace the root cause of the cancellation reason.

Since almost complete documentation of the projects were available, it was possible to triangulate the analysis by crosschecking each interpretation from several documents. This was possible because there were many documents related to the same subject, and in many cases individual documents had been created by different individuals.

The project was reconstructed from the categorized documents by anchoring individual documents and their contents to the timescale of the project. Several approaches to analyze the documentation were used until the root causes of the cancellation decisions were identified, and this is discussed in detail in Publication I. The validity of the post-mortem analysis is also discussed in Publication I. Results, which are relevant for this study, are presented in Chapter 5.

4.2 Systematic literature review

In order to be able to answer RQ1, a systematic literature review was performed, and analysis and results are presented in Publication II. Kitchenham and Charters (2007) provide a definition for systematic literature review as: "*… a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest.*" (Kitchenham and Charters, 2007, p. 3). However, a pursuit of finding all available studies has commonly been bypassed in practice. Many researchers have decided to use different digital libraries relevant for their subject, have excluded or included conference papers, textbooks, or have chosen to use a limited period of time. Examples of using constraints can be found e.g. in Beecham, Baddoo, Hall, Robinson and Sharp (2008), Hannay, Sjøberg and Dybå (2007), Jørgensen and Shepperd (2007), Kitchenham, Brereton, Budgen, Turner, Bailey and Linkman

(2009), Prikladnicki and Audy (2010), Sjøberg et al. (2005), and Šmite, Wohlin, Gorschek and Feldt (2010).

The most common reasons to undertake systematic reviews are to summarize existing information, indentify gaps in current research, or provide a framework/background in order to appropriately position new research activities (Kitchenham and Charters, 2007). One example of a well done and informative systematic review is one made by Šmite et al. (2010). In addition to it being a well done review, at the same time it shows how resource and time consuming it is to conduct a review. It took about one year to complete the review and there were four researchers involved. Another example of systematic literature reviews is one made by Beecham et al. (2008). Their systematic review was used as the basis for constructing a motivation model (Sharp, Baddoo, Beecham, Hall and Robinson, 2009), which means that the review provides a framework in order to appropriately position new research activities.

However, software researchers have only recently begun to pay attention to how to systematically locate, evaluate, synthesize, and interpret the evidence of past research (Cruzes and Dybå, 2011), and since there is a lack of common terminology and the quality of empirical studies varies, there may be conflict between internal and external validity (Sjøberg et al., 2007), or some other confusion surrounding the interpretation of the primary studies. Therefore the usefulness of systematic literature reviews weakens. One example of confusion can be found from the same systematic review of Šmite et al. (2010) mentioned above. They have classified 59 papers into numerous categories, and their classification either into inter-organizational (chain of collaborating companies) or intra-organizational (international organizations) is essential from this dissertation's point of view. However, a study conducted by Lasser and Heiss (2005) has been classified as being an inter-organizational study although the study describes in-house (intra-organizational) distributed arrangements. This unstable classification is bothersome from this dissertation's point of view but at the same time shows the need for a profound understanding of the collaboration organizations and their roles, and therefore exposes the importance of the customer-supplier framework developed and described in Publication III.

In the systematic literature review completed for this research the guidelines of Kitchenham and Charters (2007) were applied and the process and results are presented in Publication II. The review, analysis, and criteria for project success are presented in Chapter 6 of this introduction. Moreover, Case C is considered using the success criteria.

The analysis of the studies fulfilling the inclusion criteria for the review required interpretation. Since there were only a few papers, the number of the success criteria found from these papers was limited. Generalization of the identified criteria was quite straightforward, and it was possible to define the success criteria from the supplier's perspective. The validity of the literature review is discussed in Publication II.

4.3 Theory building from case studies

The rationale for the third part of the research emerged from the post-mortem analysis while considering the phase in which mistakes were made. Since it became clear that preceding phases of the software development projects are an almost unexplored area, a research approach called theory building from case studies was selected. This research approach is considered especially appropriate in new topic areas (Eisenhardt, 1989b).

Theory building from case studies has an emphasis on developing concepts, measures, and testable theoretical propositions supported by empirical evidence (Eisenhardt and Graebner, 2007). This is consistent with Sjøberg et al. (2007) and Sjøberg, Dybå, Anda and Hannay (2008) who emphasize the need for a common understanding and use of basic terminology, descriptors, and keywords, that theories should be empirically tested, and that theories should have practical value for the software industry.

The aim thus was to create basic concepts and have a better understanding of the activities performed before project start in a supplier firm. The choices and steps taken during the research are described more closely in Publication III, and only a summary of the research process is presented here.

Post-mortem analysis revealed that mistakes were made during the tendering phase and during the project start-up phase. During the research process it was decided to concentrate on the project start-up phase (RQ3) and therefore interviews about this phase were planned and conducted in four other firms which should be valid representatives of software industry in general. It was decided to choose four software engineering firms whose main business was to sell, execute, and deliver the outcome of the project to external customers and employing between 25 and 249 people. The aim was to interview practitioners who were project managers or other people responsible for project management in these firms. The aim was not to reach saturation interviewing as many project managers as possible but to interview at least two project managers from each firm.

As planned, two software engineering firms where interviews were performed were involved in software development projects for various customers. The other two firms were involved with embedded software projects with close cooperation with industrial firms. The main characteristic of all four participant firms is that they deliver unique products (software or embedded software, or in some cases specialized hardware with embedded software) for their customers. For these firms projects are their main way of doing business. The interviews and their roles are illustrated in Table 1. The term ‘Focus’ in the table relates to whether the firm develops mainly software or embedded software.

During the research process, literature on the preceding phases of software engineering projects was sought after but this turned out to be difficult. The pre-project phase seemed to be an almost non-existing area, at least when the focus is on the preceding phases from the supplier's perspective. Therefore literature was sought after from project business (e.g. Artto, Wikström, Hellström and Kujala, 2008; Kujala and Artto, 2000), project marketing (e.g. Cooper and Budd, 2007; Cova and Holstius, 1993; Holstius, 1987; Jalkala, Cova, Salle and Salminen, 2010; Kujala, Murtoaro and Artto, 2007), and project management (e.g. Dvir et al., 1998; Fangel, 1984; Fangel, 1991; PMBOK, 2008; Shenhar et al., 1997). In summary, no framework with which to bind this research was found.

RQ2 was formulated in order to encapsulate the complex situation at this phase of the research process. A customer-supplier framework was developed in order to be able to discuss the phases preceding software engineering projects in the business context and moreover, to show different perspectives.

During the development process basic analysis of interviews were utilized as well as the literature on project marketing, project management and software engineering, especially considering the supplier's perspective. This process is described more closely in Publication III. The customer-supplier framework is an answer to RQ2 and it is presented in Publication III and briefly in Chapter 7.

TABLE 1 The firms and roles of the interviewees

Firm	Focus	The role of the interviewee
Firm A	Software	Project Manager
		Project Manager
		Business Unit manager
Firm B	Software	Project Manager
		Project Manager
		Team Manager
Firm C	Embedded software	Project Manager
		Engineering Manager
Firm D	Embedded software	Project Manager
		Project Manager
		Project Manager
		Project Manager
		Project Manager
		Project Manager

In order to be able to answer RQ3, further analysis of interviews from the project start-up phase was performed. This analysis as well the main results are described more closely in Publication III. A coherent answer for RQ3 is presented in Chapter 8.

5 RESULTS FROM POST-MORTEM ANALYSIS

In order to achieve a better understanding of why software development projects fail a post-mortem analysis of five cancelled software engineering projects was conducted. The aim of the study was to find out why these five projects suffered cancellation to gain more knowledge about the reasons for cancellation decisions and the causes behind those reasons. The study is presented in Publication I and this chapter is a brief summary of the main findings from the perspective of this dissertation.

The root causes of the cancellation decisions were well known ones. The projects suffered from a lack of understanding of the real needs of the customer, the unavailability of experienced people when the project team was selected, risks connected to tight schedules, problems with selected architecture and a profound lack of understanding of the technical problem.

However, in addition to explanations for cancellations of these five cases, there were two important findings. Firstly, when there is a sub-contract relationship between a customer and a supplier it is not clear whether the project is a failure from the supplier's perspective although the project is failure from the customer's perspective. Based on the initial definition of software development project failure presented in Chapter 2 (failed software development project is a project that is cancelled before completion or delivered functionality that was not used), one case, Case C, was a clear failure. However, from the supplier's perspective the case was not so straightforward. The supplier managed to produce the software on time, within budget, and according to the scope agreed with the customer during the course of the project, and the customer paid the invoice. The key question was that whether it is possible to consider the project a failure also from the supplier's perspective, and this vague situation led to RQ1.

The second finding was related to the sub-contracting situation and a pre-project phase in such situations. It was noticed that in four cases out of five there was a sub-contracting relationship between the customer and the supplier. In these cases there was a pre-project phase in which it is possible to make mistakes resulting project cancellation. This finding led to a tentative research question concerning the pre-project phase, but this tentative research question was divided later into two more specific research questions.

These two findings from cases analyzed in Publication I were the rationale for the remainder of the research which provided answers to the overall research problem.

6 SOFTWARE DEVELOPMENT PROJECT SUCCESS FROM THE SUPPLIER'S PERSPECTIVE

The concern of how a supplier considers software development failure emerged from the post-mortem analysis presented in Publication I. In Case C the outcome of the project was delivered almost on time and within budget but the customer did not take the system into use, and therefore the project was a failure from the customer's perspective. However, from the supplier's perspective the case was different. The project was finished practically on time, and the customer paid the invoice, but it was not known whether the supplier considered the project to be a failure or not.

When software is developed through a contractual project there are two parties which have different perspectives and different goals (Collins and Baccarini, 2004; de Wit, 1988; Sadeh et al., 2000; Taylor, 2007). Furthermore, the supplier has the responsibility to develop software for the customer, and at the same time the project is a way of making business for the supplier. Therefore, it is not straightforward to convey how a supplier perceives the software development project failure. The vague situation led to RQ1 "*How is software development project success/failure defined from the supplier's perspective?*" In order to be able to answer the research question a systematic literature review was performed, and this is described in Publication II and summarized in this chapter.

It was found that de Bakker et al. (2010) argue in their recently published article, that, based on their literature review, the use of the traditional project success definition will easily lead to the conclusion that the software development project has failed. They report that the publications investigated for their paper indicate that, during the course of a software development project, the requirements originally made will almost certainly change, and this will influence the schedule and the costs. Therefore it is almost impossible to provide adequate time and budget estimates at the beginning of a software development project.

Despite this contradiction related to unreliable estimates and the need to have fixed timetables and budgets which was earlier noted e.g. by Glass (2001), research on software development project success seems to adhere to the traditional project success definition, and, unfortunately, this implies that almost all software development projects seem to fail although there are many examples of successful software implementation in our modern society.

Based on project management literature and the arguments of de Bakker et al. (2010), definitions for software development project success/failure were sought while not ignoring software development project management success/failure. This is in-line with original definition of software development project failure presented in Chapter 2 which emphasizes project failure, not project management failure.

There were numerous articles discussing project success/failure or software development project success/failure but only seven articles passed the inclusion criteria: the article has to discuss software development project success/failure or software development project management success/failure from the supplier's perspective.

Definitions for software development project success/failure were sought from these seven articles but they were not found. Instead, criteria which were used to evaluate whether software development project was a success from the supplier's perspective were found. Criteria for failure evaluation were not found. However, the main focus was on software development project success from the supplier's perspective only in one article, and furthermore, it was limited to PM success. In other articles the main focus was on risks, or trust and control, for example. This may be a reason why definitions for software development project success/failure were not found.

Three criteria for evaluation of a software development project success from the supplier's perspective were established based on success criteria extracted from the articles. These criteria are:

1) customer satisfaction,
2) short-term business success for the supplier, and
3) long-term business success for the supplier.

Hence, based on the systematic literature review described in Publication II, the answer for RQ1 is that there are no definitions for software development project failure in literature. There was no definition either for project success but these three success criteria were provided.

In order to have a definition for software development project success/failure from the supplier's perspective empirical research should be conducted. Only in one article which was accepted for further analysis in the systematic literature review the main focus was on software development project success from the supplier's perspective. Without complementary empirical research the basis for definition will remain weak.

However, it is possible to evaluate software development project success from the supplier's perspective using the three success criteria which were found from literature. The analyzed Case C in Publication I was a failure from the customer's perspective, but the situation was unclear from the supplier's perspective. The first criterion – customer satisfaction – was not realized because the customer was not satisfied. However, the second criterion – short-term business success for the supplier – was possibly fulfilled because the customer paid the invoices although it is not known if the project was profitable for the supplier. Considering the third criteria – long-term business success for the supplier – it may be assumed that long-term business success for the supplier will not be reached. The basis for this assumption is that because the customer was not satisfied, it is less likely that the customer might have more projects with that supplier in future.

This evaluation indicates that if software development project is considered as a failure when it is cancelled before completion or delivered functionality that was not used, the project may be partially successful from the supplier's perspective. This judgment is based on assumption that short term business benefit was possibly fulfilled, but long term business success might not be fulfilled because of the unsatisfied customer.

Although there were only seven articles which were accepted for analysis and considerable interpretation was required in order to identify success criteria from these articles, this short evaluation shows that these three criteria are usable for judging software development project success from the supplier's perspective. However, although these criteria are usable from the supplier's perspective these are not applicable from the customer's perspective as such. For example criterion customer satisfaction seems irrelevant from the customer's perspective in sub-contracting situations.

7 THE CUSTOMER-SUPPLIER FRAMEWORK

Since a tentative research question was formulated as "*What is a pre-project phase?*" the study around the pre-project phase was divided into two more specific research questions. RQ1 has been discussed in Chapter 6.

This revealed that in sub-contracting situations there are different aspects which should be taken into account when software development projects are studied. Moreover, the pre-project phase, in which it is possible to make such mistakes causing the project to be cancelled, should be considered together with these different aspects. Therefore RQ2 "*How should aspects of software development projects and business be combined into one framework which also shows preceding phases of the project?*" was formulated, and a customer-supplier framework was developed as an answer to RQ2. The development process and the framework itself are described thoroughly in Publication III, and only different perspectives are discussed in Subchapter 7.1 of this introduction.

In order to be able to study software development project failure in different contexts, special attention is needed while recognizing whether there is a sub-contracting relationship or not. For example the article of Lasser and Heiss (2005) presents a classification of in-house collaboration forms with 15 different stages in one single firm but the same article has been misclassified by Šmite et al. (2010) as being a case with a chain of collaboration companies.

Because it is difficult to recognize the existence of the sub-contracting relationship especially in the case of distributed software development, three examples were generated in order to mitigate difficulties in recognition of the sub-contracting relationships between customers and suppliers. These examples are presented in Subchapter 7.2. Subchapter 7.3 is a summary of this chapter.

7.1 The customer-supplier framework

The heart of the customer-supplier framework (Figure 2) is a software development project. The focus in this framework is on software development, including embedded software development, which is made through projects to external customers. The concept 'Software engineering project' is used in the framework to highlight that a project can be an embedded software development project including appliance design.

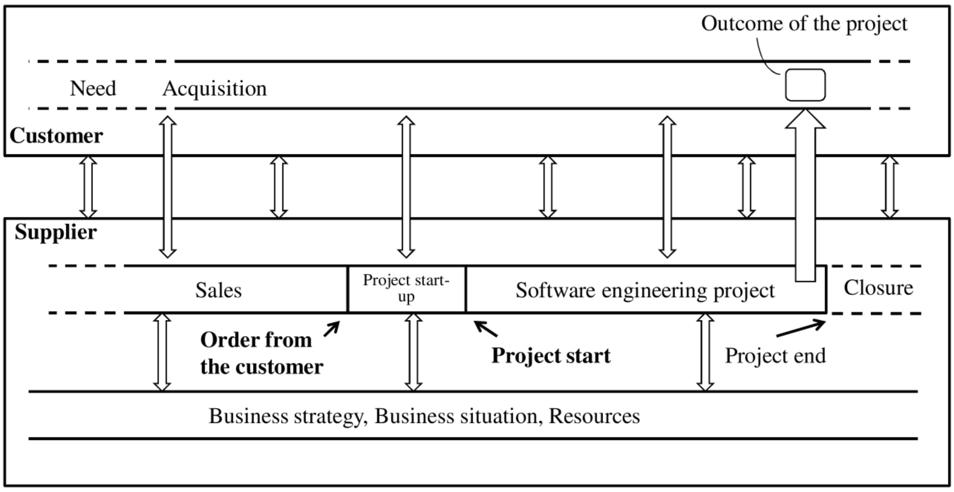


FIGURE 2 The customer-supplier framework

When looking at a software development project from the supplier's perspective there is no project without preceding phases. When the sales effort has been successful, the customer makes an order, and the sales phase ends. The concept 'Order from the customer' means the moment when the customer has made an official order or the customer has indicated in some other way that it will acquire a software engineering project with agreed deliverables from the supplier firm. After that point the responsibility for the case is transferred from sales personnel to project personnel, and the supplier starts preparations for the project execution, i.e. begins the project start-up phase. This start-up phase is described more closely in Chapter 8. The start-up phase ends when the supplier is ready to start the project execution, i.e. starts the project, as found in Publication IV. When the output of the project has been delivered to the customer, the supplier closes the project, disbands the project team, and transfers resources to other projects. The whole process is termed a 'Supply process' in ISO/IEC standard 12207 Systems and software engineering – Software life cycle processes (ISO/IEC 12207, 2008).

From the customer's side the same situation seems different, and the process from the customer's perspective is referred to as an 'Acquisition process' in the same ISO/IEC 12207 standard. From the customer's side everything starts when the customer identifies a need or a supplier succeeds in raising a need inside the customer. If the customer decides that it is neither capable enough nor has enough resources to execute an in-house software development project, the customer starts an acquisition process. It may issue a request for tenders, or start negotiations directly with one supplier which prepares a tender for the customer. Depending on the situation there may be several negotiations with one supplier or with many suppliers. When the customer has selected the supplier, the supplier takes the responsibility for the project execution; the customer monitors the progress of the project, and finally receives the output of the project from the supplier. After acceptance of the project delivery, the customer closes the acquisition process, and takes the responsibility for implementation of the output of the project into its own organization, or merges the output of the project into its own products.

The main point is that when software development is carried out by an external supplier, there are two parties which have their own projects i.e. the customer has an acquisition project and the supplier has a software development project. Both parties have different responsibilities during the whole process, and these duties should be agreed before the project starts. Other concepts which are shown in the framework and contribute to the software development project executed by the supplier are described in Publication III.

7.2 Sub-contracting situations: Examples

This subchapter underlines the importance of recognizing the sub-contracting relationship and therefore describes three basic types of different situations where sub-contracting relationships may exist with the help of three examples. These examples illustrate different sub-contracting situations where the supplier has the responsibility for developing software for the customer based on requirements defined by the buying customer. Instead of describing different software development models these examples show different situations where there may be many different projects under way at the same time, but one of these projects is a sub-contracted software development project given by a supplier to an external customer. These examples are not exhaustive but were developed in order to highlight the variety of situations where sub-contracting relationships may exist. This sub-contracting relationship adds business elements to the project executed by the supplier.

Example 1 (Figure 3) illustrates a situation where Customer A is a firm having its own ICT department. This ICT department takes care of the information systems which have been or will be developed for different departments of Customer A. The ICT department may develop either most of the software itself or buy most of it from various suppliers. Thus *purchased software supports the operations of Customer's A own departments*. Information systems of Customer A may require bespoke software development, or, if Customer A uses software products, tailoring is needed for interfaces between software products. Hence, the content and purpose of the software development project vary depending on the needs of Customer A. The dashed ellipse in Figure 3 highlights the sub-contracting relationship between Customer A and Supplier A and therefore shows where different aspects from the customer-supplier framework should be taken into account while studying software development.

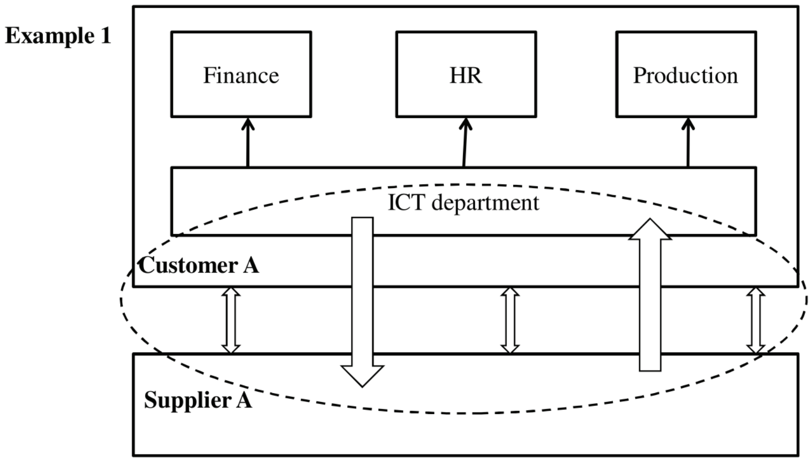


FIGURE 3 Example 1: In-house ICT department – Supplier

Example 2 (Figure 4) illustrates a situation where Customer B develops software products or services (or products with (embedded) software) to various customers. The R&D unit has the responsibility for product development although new features of existing products or definitions of totally new products are based on requirements defined by product managers or marketing people from other departments of Customer B. Concerns over time to market with tempting features is a headache for Customer B, and the timetables are fixed with predefined product releases by marketing people and management. As can be seen in Example 2, in order to keep to the software production timetable a part or most of the development is acquired from Supplier B. The main characteristic of the situation in Example 2 is that *software developed by Supplier B will be a part of a product which will be sold to various customers of Customer B*. Depending on the need of Customer B, the projects which are made by Supplier B may be unique R&D projects or enhancements of an existing software product. The dashed ellipse highlights the sub-contracting relationship between Customer B and Supplier B and this is a simplified illustration of the customer-supplier framework.

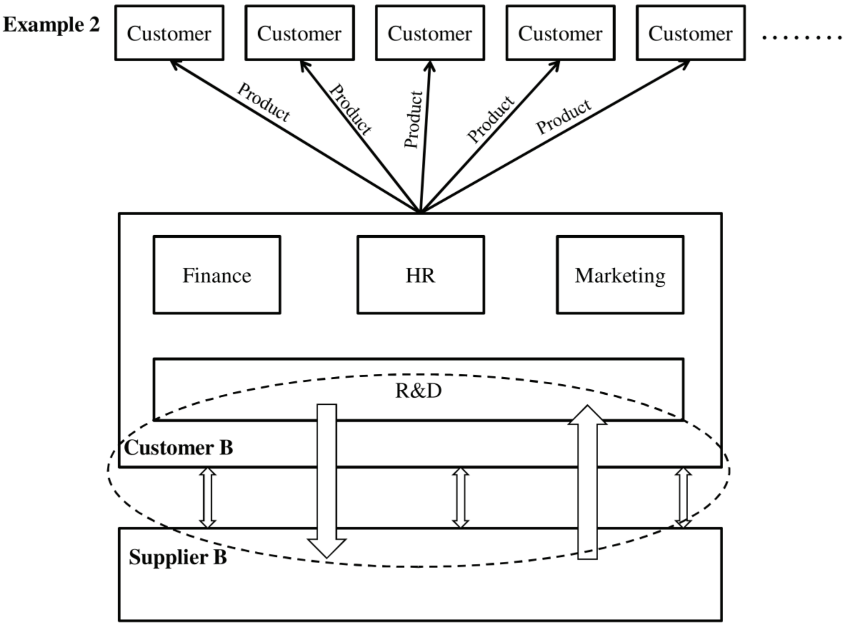


FIGURE 4 Example 2: Product development: R&D – Supplier

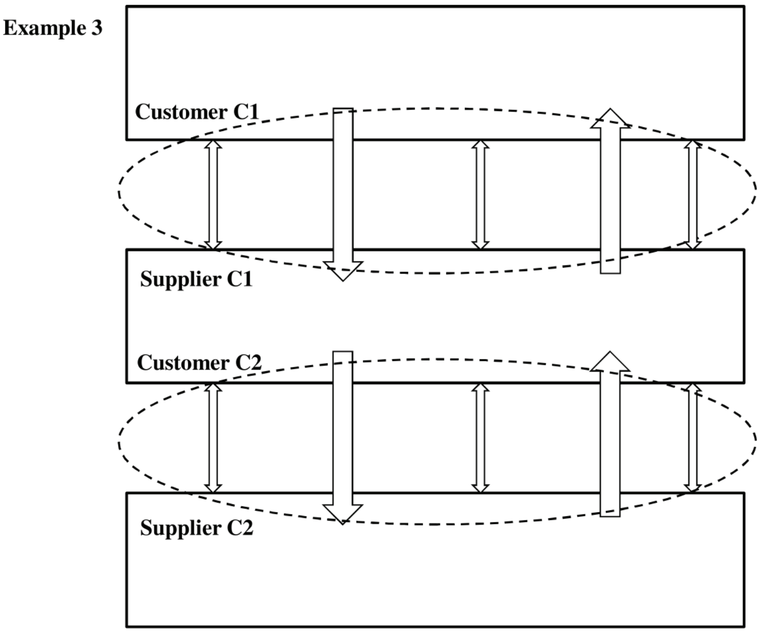


FIGURE 5 Example 3: Sub-contracting at two levels

Example 3 (Figure 5) illustrates *a supply chain* where the main sub-contracting relationship is between Customer C1 and Supplier C1. If Supplier C1 does not have enough resources or knowledge it may acquire a part of the software development from Supplier C2. In this situation Supplier C1 is at the same time a supplier for Customer C1 and a customer for Supplier C2. The dashed ellipse highlights the sub-contracting relationships between Customers and Suppliers. This example of the supply chain does not define whether Customer C1 is purchasing software for internal operations (Example 1) or for products made for its own customers (Example 2).

It has to be noted that these three examples above show different situations where sub-contracting relationships may exist but do not show all variations of distributed software development models. For example, an ICT department may develop a part of the software in-house and these in-house offices may be situated in one country or on other continents. None of the three examples illustrate this situation because there is no sub-contracting relationship between the in-house distributed departments.

Moreover, if Supplier A in Example 1 is situated in the same building in the same town as Customer A, the latter may collaborate closely with the former although they are from different organizations. Furthermore, Supplier A may have located its own offices in other countries and all these relationships differ from each other.

7.3 Summary

The developed customer-supplier framework provides a unified view of different aspects related to projects when software is developed by a supplier in the sub-contracting situation. Using the framework it is possible to distinguish between different perspectives and understand different elements which are present in sub-contracting situations, not forgetting the customer.

Three examples were generated in order to mitigate difficulties in recognition of the sub-contracting relationships between customers and suppliers. The examples illustrate a sub-contracting relationship where a supplier has the responsibility for developing software to a customer based on requirements defined by the purchasing customer. These three cases show different situations where the customer-supplier relationship may exist, and emphasize the importance of recognition of the business relation between the customer and the supplier as a part of a research setting when researching software development project failure.

8 THE PROJECT START-UP

The post-mortem analysis of five cancelled software engineering projects exposed causes for cancellations. Moreover, the post-mortem analysis revealed that serious mistakes were made before the actual project started. It was decided to concentrate on the project start-up phase from the supplier's perspective for this dissertation because there is no project until the supplier has received an order from the customer or the customer has indicated in other ways that the supplier has been selected to execute the project. Moreover, a project start-up phase precedes every software development project, and there is a scarcity of studies that discuss project start-up.

Hence, RQ3 was formulated as *"What is a project start-up of a software development project from the supplier's perspective?"* and empirical research was performed in small to medium sized software companies in Finland. The research followed research methodology that is described in Subchapter 3.3 and in Publication III. This chapter presents the knowledge which was gained during the research process, and therefore this chapter is an answer to the third research question.

Subchapter 8.1 presents an overview of the project start-up phase, provides a definition for what project start and start date are from the supplier's perspective, and defines temporal and operational boundaries of the project start-up phase. This subchapter summarizes Publications III and IV. Subchapter 8.2 of this introduction briefly presents project start-up activities identified from interviews which are described more thoroughly in Publication III. The need to define success criteria before the project has been started is highlighted. Subchapter 8.3 summarizes findings from Publication V and presents a three-level understanding which should be gained in order to enhance overall project success from the supplier's perspective. Subchapter 8.4 is a summary of this chapter.

8.1 Overview

During the research process it became clear that the project start-up phase has remained quite an unknown concept in literature on software development projects. This is supported by searching for the word ‘start-up’ in Project Management Body of Knowledge (PMBOK) which is an American national standard ANSI/PMI 99-001-2008 for project management. Only one occurrence was found, on page 141 (PMBOK, 2008). Moreover, the standard ISO/IEC 12207 Systems and software engineering – Software life cycle processes (ISO/IEC 12207, 2008) which has been used in this dissertation as a reference standard, does not recognize the word ‘start-up’ at all. Furthermore, only a handful of articles which discuss project start-up has been found during this research process. These articles are presented in Publication III. However, Turner states that it should now be widely accepted that a structured start-up process is an essential part of project management (Turner, 2009, p. 266), which is agreed with in this dissertation.

The basis for understanding of project start-up has been made in the INTERNET Committee on Project Start-up which was founded at the end of 1984. The work has been filed in a book called ‘Handbook of Project Start-up: How to launch any phase effectively’ (Fangel, 1990) and consists of 74 articles, abstracts, or reports which have been written for the workshops, congresses, symposiums, or conferences around the theme during 1981-1988. The book contains descriptions of project start-ups in general or in a specific industry, start-up of international projects, research and development projects, contractual projects, governmental projects, and small bespoke projects. There are checklists, guidelines for conducting start-up workshops, models for project start-ups and discussion on competences and skills of project team. Some of this information has been summarized in (Turner, 2009) as general guidelines for project start-up but these guidelines have focused on the project start-up from the customer’s perspective.

It seems that project management literature has almost forgotten the concept of project start-up after an initial active period of research conducted during 1981-1988. This can partly be seen from literature review made by Themistocleous and Wearne (2000). They have conducted a survey of topics of project management in the International Journal of Project Management (IJPM) from 1984 to 1998, and Project Management Journal (PMJ) from 1990 to 1998. They did not use the concept of ‘project start-up’ but identified the concept of ‘project launch’ which to some extent could relate to project start-up activities carried out at the beginning of the project. They found that only 12 papers out of 538 (1%) have discussed project launch in papers published in IJPM. None of the 20 articles published in PMJ were about project launch.

The paucity of research on project start-up is supported in a study of Crawford, Pollack and England (2006). They conducted a study on the trends of emphasis within the project management literature over the period 1994-2003.

This study covered articles published in the IJPM and in the PMJ. They found that only Betts and Lansley (1995) have found increasing interest in project start-up during 1983-1992. Project start-up consists of subjects like feasibility, briefing, requirements, definition, startup activities, and project finance according to a classification by Betts and Lansley (1995), i.e. only some of those studies were about project start-up as understood in this dissertation. At the same time Crawford et al. (2006) interpreted that Themistocleous and Wearne (2000) did not see any interest in project start-up, and this may be due to the absence of the concept of 'project start-up' and on the use of the concept of 'project launch' in topics of project management identified by Themistocleous and Wearne.

During this research process only two decent articles concentrating on project start-up was found, excluding articles published in (Fangel, 1990) or written by Fangel. These were Egginton (1996) and Halman and Burger (2002). These articles are discussed in Publication III. In addition to being a main subject in an article, project start-up can be a minor subject within other project management issues e.g. in Briner, Hastings and Geddes (2009, p. 93-105), Turner (2009, p. 265-278), and Turner and Cochrahe (1993). However, it is more common to encounter the concept of project start-up as a part of some other project management issue as in Taylor (2005) or Thomas and Fernández (2008).

Of concern is that the newest version of PMBOK does not discuss project start-up at all. Instead, PMBOK uses the concept of 'project initiation' which is defined as "*Launching a process that can result in the authorization of a new project.*" (PMBOK, 2008, p. 443). The processes which may result in a new project are 'Develop Project Charter' and 'Identify Stakeholders', and these processes belong to 'Initiating Process Group'. However, from the supplier's perspective this may be understood to mean sales phase, but in that case there is always a risk that there will be no project. Moreover, from the supplier's perspective there will be a project only after the customer has made an order, and therefore the concept of project initiation does not apply to supplier firms. Instead, in supplier firms there is a special start-up phase, and descriptions and guidelines for supplier firms are needed.

The prevailing situation emphasizes the work carried out by the INTERNET Committee on Project Start-up (Fangel, 1990). In addition to numerous articles published by the committee, they have defined the focus of their work. Project start-up has been defined as: "*Project start-up is a short-term systematic process designed to promote mutual understanding and cooperation among project participants.*" (Fangel, 1991, p. 5). Moreover, another definition is given as "*Project start-up is a unified and systematic management process which quickly generates a platform for taking off and for getting going effectively.*" (Fangel, 1991, p. 6). Its purpose has been presented as "*To create a solid base for management of the project – both through systematic project planning and through effective team building.*" (Fangel, 1984, p. 242). The objectives of a project start-up can been expressed as: to create a shared vision for the project, to focus the attention of the project team on the project's purpose and the method of achieving it, to gain acceptance of the plans, and to get the project team functioning (Turner, 2009, p. 266-267; Turner and Cochrane, 1993).

Both Fangel (1987) and Turner (2009, p. 266) has highlighted the need for effective start-up on projects and this may be due to:

- the lack of time after the contract,
- the increasing complexity of projects,
- the need for implementation of qualified project management earlier in the life cycle,
- the need for team building and cross-cultural cooperation,
- the need for increased effectiveness caused by shorter product life cycles, and
- changes in the way projects are managed, including goal-directed approaches, which reinforce the setting of objectives, the use of group methods for building cooperation, and the management of the team through the use of a clear and common mission.

When the INTERNET Committee on Project Start-up named their book, they noted that a project start-up may take place at the start of any of the stages in the project life cycle (Fangel, 1990). Turner agrees with them expressing that start processes may be conducted whenever there is a significant change in the project team, either in its composition or structure, or when the project team's attention needs refocusing on the objectives of the stage ahead (Turner, 2009). Moreover, Barry et al. (2002) showed how even a small sized project which is completed over a long time period can require stops and re-starts in order to implement changing requirements. Because these re-starts require much more effort than anticipated due to the time required, efficient start-ups are needed during the course of the project implementation.

However, in this dissertation any project start-ups are not discussed but the concentration is on the project start-up phase which precedes the project and ends with project start. As a consequence, there may be confusion over the concepts of *project start* and *project start-up*. Fangel draws the analogy with starting the engine of a car, and starting-up the diesel engine in a ship, described in (Fangel, 1991; and Publication III). The former is an action at an instant of time, and the latter is a structured series of activities, a start-up process, which gives the most efficient and economical operation (Turner, 2009, p. 266). In this dissertation both concepts are important, and their meanings need to be clarified.

The concept of project start may not be very important if we are discussing an in-house project or a product development project. However, in the case of a sub-contracted bespoke project the situation is different. The supplier and the customer should have a common understanding regarding the scope, timetable, and cost of the project. This understanding should come into existence during negotiations between the supplier and the customer before the project is allowed to start, and this understanding is usually clarified in the commercial and legally binding agreement and the project plan.

When doing software projects as a business, the supplier has a need to estimate effort as realistically as possible, and the conceivable delivery date is estimated using effort estimation as a basis. Consequently, from the supplier's perspective it is important to have an original baseline and to fasten a date from which the project is considered started, and moreover, allocate resources to be ready from that date. From that moment of time the project should be under way and upcoming changes are made against that moment of time. If the project starts late, it is illogical to expect that later tasks will be completed earlier in order to make up the difference (Kappelman, McKeeman and Zhang, 2006). Moreover, allocated resources may already be reassigned to other projects. Especially from the supplier's perspective it is important to complete the project on time and within budget in order to have a profitable project (Collins and Baccarini, 2004; Mao, Lee and Deng, 2008), and therefore the project start should be agreed with the customer.

However, although the start date is important from the supplier's perspective and standards call for the definition of project start date defining a project as an "*Endeavour with defined start and finish dates undertaken to create a product or service in accordance with specified resources and requirements*" (ISO/IEC 12207, 2008, p. 5), it was found that literature does not provide any definition for project start. This vague situation is discussed in Publication IV.

Moreover, it was found that there is no common understanding of project start inside any company where project managers were interviewed. Every interviewed project manager seemed to have their own definition for the project start. Therefore each project manager has his/her own definition of what constitutes the start date of the project. This type of ambiguity may result in a situation in which the supplier performs work that is necessary for the project but is not included in the agreement, and therefore not only the project start but also project start date and the work included in a project should be defined.

Different definitions for project start were provided by interviewed project managers. After analysis similar definitions were grouped together, and finally there were five different definitions. These definitions were placed in a time scale representing the relative order of the definitions and the result is seen in Figure 7. Moreover, there is a suggestion for internal project start as well as a suggestion from where the project start-up phase starts and where it ends. These suggestions for definitions are discussed next.

The definition 'We got the order' is defined to be the start of the project start-up phase of the supplier firm. The supplier needs an order or some other indication that the customer has selected the supplier for its contractor. Without the order or some other indication the supplier takes a risk that the customer changes its mind, may decide not to start the project at all, or select another supplier.

After the supplier has got the order, the project manager is selected. Selection of the project manager is important but the selection is seen here mainly as managerial work. The next step is the moment when 'Project work has been started'. This means that someone – it may be a project manager or other person – who is appointed to the project team starts working on the project. It was suggested in Publication IV that the moment when 'Project work has been started' should be fastened to the internal project start / internal project start-date.

Finally, it was suggested in Publication IV that the official project start / project start date should be fastened to the date where a kick-off meeting with the customer has been conducted, and it is the moment when the project start-up phase ends. After that meeting the customer and the supplier should have a common understanding of the project, its objectives, budget, and schedule, and the customer may expect the supplier to have the project team up and running. Consequently, all project start-up activities that are required should be performed before that meeting. The interviews conducted with Project Managers identified eleven of these activities and these are described in Subchapter 8.2 and in Publication IV.

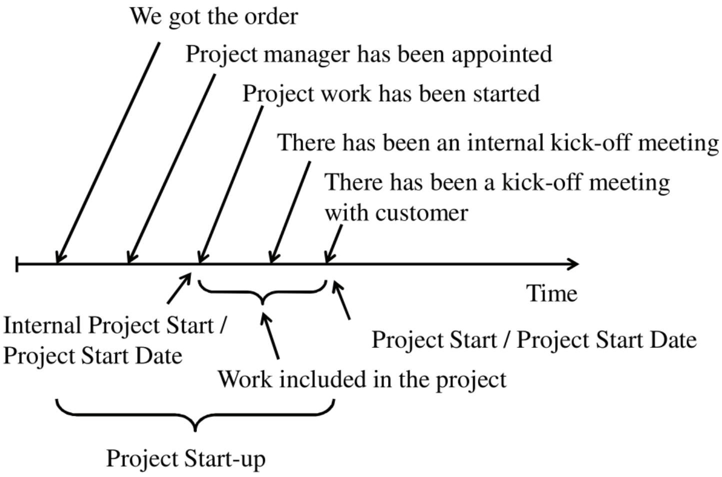

We got the order
Project manager has been appointed
Project work has been started
There has been an internal kick-off meeting
There has been a kick-off meeting with customer
Time
Internal Project Start / Project Start Date
Project Start / Project Start Date
Work included in the project
Project Start-up

FIGURE 6 The definitions for internal project start and project start

After the supplier has received an order from the customer, the supplier is often under pressure to start the project as soon as possible. Silvasti (1987) has described the problems which the supplier encounters during the start-up process as:

- a lack of time after the contract,
- a lack of start-up resources,
- multi project environment,
- many potential projects needing start-up activities, and
- uncertainties concerning the starting projects.

He describes further: "*The work has to be started immediately – in one or two weeks. That would be possible if we could do all the preparation work before the contract. But* *we don't because for this we need costly resources and there is still a risk that we will lose the contract"*.

The supplier suffers from a contradiction. Although there is pressure to start the project as soon as possible, the supplier should concentrate on understanding what kind of project they are starting, and that will take time. Barry et al. (2002) have studied the relationship between the software project duration and project effort, and they have noted that for a given project a minimum time is required to get complex work started. An organization needs time to set up the project team, train them and allow them to become familiar with the project.

For the supplier, project start-up typically means the mobilization activities that they must engage upon following the award of the contract (Morris, 1987). However, no studies or textbooks were found which describe activities that are performed by software development supplier firms during the start-up phase. The few studies which discuss activities of any start-up phase are presented in Publication IV, but activities identified from interviews conducted for this research are presented in the next subsection.

8.2 Activities

Eleven activities were identified from interviews of project managers during the research conducted for this dissertation. The research process, including the analysis of the interviews, is described more detailed in Publication III, and the activities are briefly presented in this subchapter. With the help of these activities it is possible to understand what happens in a software development firm during the start-up phase in practice.

Software development project management has been discussed in textbooks (e.g. Pressman, 2005; Royce, 1998). However, no studies have been found discussing activities which are, or should be, performed before project start when there is a sub-contracting relationship between the customer and the supplier. In these situations there is pressure to start the project as soon as possible because of commercial obligations to the customer, but at the same time the project manager and the project team should have enough time to familiarize themselves with the project.

Interviews carried out for this research revealed that project managers' relationship with upcoming projects varies. Some project managers did not hear about a project until they were nominated as the project manager for it. On the other hand, one individual acted as a salesman, sold the project, got the deal, gathered as good a project group as possible, and continued as the project manager. Hence, considering their roles and possibilities to get familiar with the project before project start, there are differences between the project managers. This difference between the project managers' roles and responsibilities was found inside all firms. Although only project managers were interviewed for the start-up phase, due to these differences it was possible to get a wide perspective into the project start-up phase from the supplier's perspective.

Altogether eleven activities were identified from the interviews with project managers. The activities are a fusion of all activities described by project managers, and that means that all activities were not described by all project managers. The activities identified from the interviews are:

- select the project manager,
- gather extant material,
- familiarize oneself with the project,
- select the project team,
- administer the project,
- plan (re-plan) the project,
- set up a development environment,
- make the handover from sales to project,
- plan/negotiate with a customer,
- run an internal kick-off meeting, and
- run a customer kick-off meeting.

These activities are a summary of operations which are, or should be, performed during the project start-up phase in supplier firms which develop software. Each activity is described more closely in Publication III in which there is also a comparison between these activities and existing studies found from literature on general project start-up activities.

The activities which are performed during the start-up phase aim mainly to quickly generate a platform for taking off and for getting going effectively (Fangel, 1984; 1991). However, when projects are made for external customers and the projects are a way to do business, goals exist which should be considered before project start.

Based on the systematic literature review described in Publication II and briefly in Chapter 5, there are three success criteria for software development projects from the supplier's perspective. Project outcomes should please the customer, the project should be profitable and moreover, long term benefits should also be reached. Therefore at the same time there is a concern of customer satisfaction and the possibility of re-buying (Cova and Salle, 2007), but projects should also bring value to the supplier (Shenhar et al., 1997). Moreover, there may be possible disparity between meeting short-term project goals and satisfying the customer, and disparity between meeting short-term and long-term benefits during the project execution, and these contradictory situations should be recognized. Hence, project managers who have the responsibility of managing the software development project should be aware of the results expected from their projects not only in the short- but also in the long run.

Consequently, the success criteria from the supplier's perspective should be defined before the project has even begun and prior to committing an organization's resources to it (Shenhar et al., 1997). To bear business objectives in mind, management should specify project objectives as early as possible and focus the project manager's and team members' attention on the expected results from the project.

If the success criteria are not agreed, different members may pursue their own goals instead of project's goals (Mahaney and Lederer, 2003). If the success criteria are agreed at the start of the project, project management adjustments will be made during the project execution (Thomas and Fernández, 2008), and therefore project objectives are presumably easier to reach. Moreover, Kappelman et al. (2006) have stated that projects with undefined success criteria by definition cannot succeed.

However, one of the important findings from this research is that none of the interviewed project managers mentioned spontaneously that the success criteria for their projects are defined before project start. This finding may indicate that it is not common for supplier firms to actively aim at defining success criteria for each project and therefore do not encourage their project teams to engage themselves in such a way that the project success criteria will be met

Thus there may be a contradiction between practices in supplier firms and the studies of Shenhar et al. (1997), Thomas and Fernández (2008), and Wateridge (1998), which suggest that the success criteria should be agreed before the project start, and if not, the possibility to reach project success is weakened. Moreover, if project success criteria are not defined and agreed, it is not possible to evaluate whether the project was a success or not (Kappelman et al., 2006). This may be one explanation why software development projects have a reputation for failure.

8.3 Three-level understanding

The overall research problem of this dissertation was to understand why software development projects fail. Although the overall research problem is about software development project failure, it can be supposed that the aim of the supplier is to have a successful project. Therefore during basic analysis identification of the key activities of the software project start-up phase that enhances overall project *success* from the supplier's perspective was striven for. This part of the analysis is presented in Publication V.

The research process progressed and it was understood that the key activities which are discussed in Publication V describe a *three-level understanding* which should be attained during the project start-up phase in order to enhance project success. Due to this advanced understanding, the concept of 'understanding' is used in this dissertation instead of the concept of 'key activities' which has been used in Publication V.

More understanding on the project start-up was gained from the Handbook of Project start-up (Fangel, 1990). At the project start-up phase the scope of work and constraints of quality, cost and time are fixed, either known by the customer or simply estimated (Passalacqua, 1987; Turner, 2009, p. 276-277), and this means among others preparation for changes which will emerge during the course of the project. Therefore starting-up the project means understanding its objectives and making them flow throughout the project team (Passalacqua, 1987). However, it was identified from interviews that understanding should be attained not only at one level (project team) but at three levels which are project manager, project team, and customer/supplier. These three levels are discussed here, and also in Publications III and IV.

Project manager. The Project manager is the person who is assigned by the performing organization to achieve the project objectives (PMBOK, 2008, p. 444). In supplier firms, until the customer has made an order, it is unsure whether there will be any project at all. The interviews revealed that some project managers did not hear of the project until they were nominated as its manager while others were already involved during the sales process. Although the situation varies among project managers, 10 project managers out of the 14 emphasized the importance of achieving a holistic understanding of the project and its objectives. The project manager should formulate a big picture of the project and its goals, as formulated in Publication III. Without that understanding it is not possible to write the project plan or rewrite any project plan already created during the tendering phase. This applies especially to the schedule because only with a holistic understanding it is possible to estimate whether the original schedule is realistic or not. Moreover, without holistic understanding the project should not be allowed to start, described by one project manager:

> The customer had a tight schedule and we needed three – four designers and to work with existing specifications. When looking back we should not have done any design work on the first day. Instead, I should have moved to the customer for a while to fathom out what was the need and if the specifications were finished or not.

Some project managers commented that at this stage, it is not possible to understand in detail what will be done during a project but a more general understanding has to be gained. It is essential to understand early what is really important so that it is possible to guide the project team to devote its time to important matters instead of less important details.

Project team. A Project team consists of the project management and other team members who carry out the management and project work. The team is comprised of individuals from different groups with knowledge of a specific subject matter or with a specific skill set who carry out the work of the project (PMBOK, 2008, p. 26). At the project start-up phase this novel organization – project organization – is created for undertaking a unique task (Turner, 2009, p. 265), and therefore just as a project manager needs to achieve a holistic understanding of the project and its objectives, the whole project team also needs to understand and agree on what tasks have to be done in order to accomplished the project (Kappelman et al., 2006). One project manager described this as follows:

> It’s better that we don’t do anything but sit on our hands for two weeks so that it’s clear to everyone what we are aiming for and what are the goals.

The interviewees provided examples of how this understanding might be attained, and these are discussed in Publication V.

Supplier/customer. When software is developed for an external customer, the customer and the supplier should have a common understanding of the project and its objectives. However, from the supplier’s side there are sales representatives who negotiate with the customer for the content of the project. When the project manager is nominated they take responsibility, and start to familiarize themselves with the project. The project manager has to evaluate whether it is even theoretically possible to implement the desired outcome, or realistic to execute the project within the given timetable, costs, and scope. Therefore the understanding about the project gained during the sales process between the customer and the supplier, may no longer be valid, or need renegotiating with the customer. It was identified from the interviews that project managers want to ensure a common understanding between the customer and the supplier before the project start point. If the customer and the supplier have a common understanding of the project and its objectives before project start, it is possible to mitigate any problems connected to change management and negotiations on how to compensate for the changes. It was noted that the project managers had a realistic attitude that there will be changes during project execution and they must prepare for future changes during the start-up phase.

The need to gain this three-level understanding was identified from the interviews in order to enhance the overall success of software development projects of supplier firms. However, many project managers said “…. *we should have done / we should do* ….” when describing experiences of the projects related to gaining this three-level understanding. Therefore more empirical research is needed on sub-contracted software development projects in order to first identify the problems related to this three-level understanding and later provide guidelines related to gaining three-level understanding in contractual business situations.

8.4 Summary

This chapter addresses RQ3. In addition to an overview of project start-up, it is an exposition of the start-up phase of a supplier firm that develops software for an external customer in a sub-contracting relationship. During the research conducted for this dissertation, eleven activities were identified from interviews with project managers. In addition to the activities, it was identified that there is a three-level understanding which should be attained during the project start-up phase in order to enhance project success from the supplier’s perspective. Moreover, some of the results of the research conducted for this dissertation were to discover definitions for internal project start and internal project start date as well as project start and project start-date. Furthermore, activities which should be included in project work were suggested.

In addition to the overview of the start-up phase of a software development firm, it was found that activities described by interviewed project managers did not mention that the success criteria for their projects are defined before project start. This important finding may indicate that it is not common for supplier firms to actively aim at defining success criteria for each project and encourage project team to engage themselves in working for a project in such way that its success criteria will be met. Consequently, there may be a contradiction between practices in supplier firms and the studies of Shenhar et al. (1997), Thomas and Fernández (2008), and Wateridge (1998), which suggest that the success criteria should be agreed on before the project start.

9 SUMMING UP THE RESEARCH

This dissertation presents many issues related to software development project failure. This chapter provides answers to the research questions in Subchapter 9.1, and the research problem in Subchapter 9.2. Validity and assessment of the research conducted for this dissertation are discussed in Subchapter 9.3. Implications for practice are considered in Subchapter 9.4. Subchapter 9.5 discusses limitations, and implications for future research are presented in Subchapter 9.6.

9.1 Answers to the research questions

The overall research problem was "*Why do software development projects fail?*". More specific research questions for this dissertation were formulated during the research process and they were:

- How is software development project success/failure defined from the supplier's perspective? (RQ1)
- How should aspects of software development projects and business be combined into one framework which also shows preceding phases of the project? (RQ2)
- What is a project start-up of a software development project from the supplier's perspective? (RQ3)

Answer to RQ1

In order to answer RQ1 a systematic literature review was performed. There were four articles which discuss software development projects from the supplier's perspective, and three more articles that took both perspectives into account. Based on these articles the criteria for software development project success from the supplier's perspective are:

1) customer satisfaction,
2) short-term business success for the supplier, and
3) long-term business success for the supplier.

Definitions for software development project failure were not provided in these seven articles which were accepted for analysis.

In conclusion, the answer for RQ1 is that, based on systematic literature review, there are no definitions for software development project failure. There were no definitions either for project success, but three project success criteria were indentified from the literature and these criteria are presented above.

In addition to presenting the identified success criteria from the supplier's perspective, usability of these criteria was evaluated. Case C, which was analyzed in Publication I, was a failure from the customer's perspective. If project success is judged against these criteria, Case C may be considered partially successful from the supplier's perspective because the project might have been successful in the short-term. Therefore these three success criteria are usable for judging software development project success from the supplier's perspective.

Moreover, it was found that distinction between *project success* and *project management success* is not commonly used regarding research on software development project success or failure. The use of traditional project success definition will easily lead to the conclusion that the software development project has failed.

Answer to RQ2

The customer-supplier framework developed during this research provides the answer to RQ2. This is described briefly in Chapter 7.1 and more thoroughly in Publication III. With help of the framework it is possible to discern different perspectives and elements which are present in sub-contracting situations, and therefore the framework supports the evaluation of project success/failure from each perspective in sub-contracting situations.

In addition to the customer-supplier framework, three examples were provided in order to show different situations where a sub-contracting situation may exist. Before it is possible to recognize whether there is a sub-contracting relationship or not, it is essential to recognize who are the organizations that participate in software development projects. If organizations are from different firms, a sub-contracting relationship exists between these parties. Consequently, software development projects made by suppliers contain business elements which should be taken into account while researching software development project failure.

Answer to RQ3

In this dissertation, Chapter 8, complemented by Publications III, IV, and V, provides the answer to RQ3. For the supplier, project start-up typically means the mobilization activities that they must engage in following the award of the contract (Morris, 1987), and these activities have been identified through a study in four software development firms. It has been noted that suppliers suffer from a contradiction because, although there is pressure to start the project as soon as possible, the supplier should concentrate on understanding what kind of project they are starting, and that will take time. The need to gain a three-level understanding before project start was identified.

In addition to the overview, activities, and the three-level understanding discussed in Chapter 8, it was discovered that it is important to agree success criteria before project start in order to commit the project team to project goals. However, it was found that project managers did not mention spontaneously that the success criteria for their projects are defined before project start. This finding may indicate that it is not common for supplier firms to actively aim at defining success criteria for each project and encourage project teams to engage themselves in working for a project in such way that its success criteria will be met. Moreover, without defined success criteria there are not any success criteria which to pursue. This may lessen the successfulness of software development projects, and, in addition, final evaluation of the project success is not possible.

9.2 Answers to the research problem

The overall research problem was "*Why do software development projects fail?*", and it was approached by conducting a post-mortem analysis of five cancelled software engineering projects. The root causes of the cancellation decisions in these cases provide explanations for these five cases, and these were lack of understanding of the real needs of the customer, the unavailability of experienced people when the project team was selected, risks connected to tight schedules, problems with selected architecture, and a lack of understanding of the technical problem.

In addition to explanations for the failure of specific projects, this dissertation provides four other explanations for the research problem.

Firstly, a software development project may fail because of serious mistakes which were made before the project, especially in sub-contracting situations. The supplier is often under pressure to start the project as soon as possible during the project start-up phase (Silvasti, 1987), although the supplier should concentrate on understanding what kind of project they are starting, as found in Publication V. Therefore this phase is especially favourable for mistakes.

Secondly, whether a software development project has been a failure or not depends on the perspective from which the evaluation has been made especially in sub-contracting situations. If the project is a failure from the customer's perspective, it is not straightforward to evaluate whether the project is also a failure from the supplier's perspective. Three success criteria from the supplier's perspective were identified, and these are not necessarily applicable when considering software development project failure from the customer's perspective. Therefore, it is essential to define the perspective from which the failure/success is evaluated, and clearly emphasize whether a software development project was either a failure or success from each perspective respectively.

Thirdly, a software development project which has been considered a failure may not be a failure at all. A reason for this explanation is that the software development project may have failed against *project management* success criteria but not against *project* success criteria. Without making a distinction between project failure and project management failure it is easy to conclude that the software development project has failed.

Fourthly, software development projects may fail because supplier firms do not seem to actively aim at defining success criteria for each project during the start-up phase. Without defined success criteria different members may pursue their own goals instead of project's goals (Mahaney and Lederer, 2003), it is difficult to make such project management adjustments during the project execution which ensure project success (Thomas and Fernández, 2008), and evaluation of the project success is not possible (Kappelman et al., 2006).

9.3 Validity and assessment of the research

This subsection discusses validity issues, possible weaknesses, biased perception and interpretation which were identified or which may have occurred during the course of the research. These may have had an influence on the results and therefore are discussed here. Validity issues related to the post-mortem analysis of five software engineering projects are considered in Publication I and are not discussed here. Similarly, validity issues related to the three research questions are considered in Publication II and Publication III and are not discussed here.

The research on software development project failure in this dissertation is based on certain concepts and how these concepts are defined and understood. Defining and understanding of concepts refers to construct validity which means "*identifying correct operational measures for the concepts being studied*" (Yin 2009, p. 40). The main concepts of the research carried out for this dissertation are project, supplier, customer, and sub-contracting relationship.

The first concept 'projects' is considered in Chapter 2 and therefore is not discussed here. The concepts of 'supplier' and 'customer' are understood as representing organizational stakeholders of a project, and sub-contracting relationship describes the relationship between them. Standard ISO/IEC 12207 defines a supplier (or contractor, producer, seller, or vendor) as "*organization or individual that enters into an agreement with the acquirer for the supply of a product or service*" (ISO/IEC 12207, 2008, p. 7). Customer (or acquirer, buyer, and purchaser) is defined in the same standard as "*organization or person that receives a product or service*" (ISO/IEC 12207, 2008, p. 3). In this standard it was noted that the supplier or the customer may be internal or external to the organization, i.e. sometimes both parties may be part of the same organization. However, when the supplier and the customer are from different organizations, there is a sub-contracting relationship between these parties. This sub-contracting relationship is not defined more precisely in this dissertation but the nature of the relationship is commercial and therefore there are business elements which are added to every software development project made by the supplier. The developed customer-supplier framework illustrates some elements which are present in such relationships.

The customer-supplier framework which was developed during the research process is partially validated. It has been shown to project managers who have given supporting comments on it and no conflicting opinions have been received during these discussions. The customer-supplier framework is a means to understand the complicated sub-contracting situation which has an influence on software development project failures from the supplier's perspective, not a main result as such.

The customer-supplier framework includes the concepts of project, supplier, customer, and sub-contracting relationship, and there are relationships between these main concepts. When software development project failure is studied in a sub-contracting context, there is a concern of internal validity which refers to "*seeking to establish a causal relationship, whereby certain conditions are believed to lead to other conditions, as distinguished from spurious relationships*" (Yin, 2009, p. 40). Hence, it is assumed that the sub-contracting relationship between customer and supplier is an explanatory factor for certain actions which may lead to software development project failure, either from the customer's or the supplier's perspective.

While no specific theories were used in this dissertation, both the research on the project start-up phase from the supplier's perspective and the customer-supplier framework laid the foundation for more theoretical research in future. The main perspective of this dissertation on software development project failure has been the suppliers' perspective. Therefore, instead of being grounded on existing theories e.g. agency theory (Eisenhardt, 1989a), it may be rewarding to drawn on quite a new research area - project business (Artto and Kujala, 2008; Kujala, Artto, Aaltonen and Turkulainen, 2010). This is supported by Taylor (2007, p. 20):

> However, the core business of the contracting organization or vendor is the delivery of projects to customers, and hence *vendor* firms have the additional goal of completing their projects so successfully that their reputation is enhanced and prospects for future business and on-going revenue are improved.

Hence, when software is developed by an external supplier, success/failure of these projects should be researched in the context of project business.

Both empirical parts conducted for this dissertation have been made in software firms operating mainly in Finnish markets. One exception was a firm whose project was analyzed in Publication I. That firm operates in European markets. In Finland most of the firms are small, and there are large general IT service firms that operate on a project basis and provide solutions to larger companies and the public sector (Software Industry Survey, 2011). Thus, the results may reflect the features of Finnish business life derived from project documentation of the post-mortem analysis and interviews.

The number of interviewees (14 project managers from four software engineering firms) and the fact that only project managers were interviewed are a limitation of this research. However, each firm was selected to be valid representatives of the software industry as they develop bespoke software to external customers through projects and at least two project managers were interviewed from each firm. Although only project managers were interviewed, due to differences between project managers' roles and responsibilities a wide perspective into the project start-up phase from the supplier's perspective was attained.

Interviewees provided descriptions of the project start-up phase and these descriptions reflect the perceptions of the firms in question as well as Finnish society and culture. However, the activities identified from the project start-up interviews were compared with the activities found from literature, and no cultural characteristics were found. The reason for that may be that the literature and the interviews originate in the western world, and there is no research on the project start-up phase e.g. from Asian countries. Nevertheless, the results may be applicable in European countries having a software industry with a strong presence of SMEs, e.g. Ireland (Coleman and O'Connor, 2008), or Australia (Cater-Steel, 2001). Hence, it is suggested that results are applicable to firms which develop bespoke software for external customers in Europe and Australia.

The author's own background is in the software industry. This has affected what has been researched for this dissertation and how this has been done. The main effect has been the interest in the supplier's perspective in general, trying to have answers for supplier firms. Another embodiment has been catching the pre-project phase and especially the project start-up phase during the research process. From the supplier's perspective the start-up phase is hectic work which sometimes may seem chaotic.

9.4 Implications for practice

In addition to answers to the research questions and the overall research problem, this dissertation has many implications for practice, and these are presented as recommendations to supplier firms:

- **Success criteria for software development projects from the supplier's perspective should be defined beforehand during the start-up phase, before the project has been started.** If success criteria are agreed at the start of the project, project management adjustments will be made during the project execution, and therefore project objectives are presumably easier to reach. The systematic literature review showed that success criteria may contain business objectives both in the short- and long-term, and also other success criteria as customer satisfaction.

- **The difference between project success criteria and project management success criteria should be understood.** Instead of striving for project management success, project managers should become aware of the results expected from their projects from the business perspective. This situation is quite complicated because there may be a possible disparity between meeting short-term project goals and satisfying the customer, and a disparity between meeting short- and long-term benefits during the project execution. Although there may exist such contradictory goals, they should be recognized, and project and project managements success criteria should be agreed before project start.
- **Suppliers should keep the agreed success criteria in mind and make decisions during the project start-up phase which do not hinder project success but rather contribute to the project positively to ensure its success.** The post-mortem analysis revealed that it is possible to make mistakes during the pre-project phase that ultimately result in project cancellation. Therefore the project start-up phase was examined more closely, and it was found that many important decisions are made during the project start-up phase.
- **Supplier firms should invest in planning practicable start-up processes for different projects, and decide whether this work is included in the project or not.** During the start-up phase there is a need for a rapid start-up of the project after the customer has made an order (Silvasti, 1987), and there is pressure to start the project as soon as possible. The need for a rapid start-up may lead the supplier to rush and forget that time is needed in order to set up the project team, train them, and allow them to become familiar with the project (Barry et al., 2002). Moreover, it is tempting to start the project as soon as possible without performing adequate preparatory activities because if the project start-up activities are not chargeable there is the threat of losing profitability on the project before it has begun. Since project start-up activities are visible and there is defined and clear start-up process, it is easier to realize how much preparation effort and calendar time is required from the supplier to perform the start-up of a project. Moreover, if suppliers are prepared to carry out that phase efficiently, customers will not probably question whether that work is chargeable or not.
- **Supplier firms should carry out post-mortem analysis of their projects more often, analysing also the pre-project phase.** The post-mortem analysis of five cancelled projects was a rationale for this research. Without post-mortems little understanding is gained from past projects, and therefore, it is easy to repeat the same mistakes over again.

In addition to the recommendations for supplier firms, this dissertation presents explanations of why five software engineering projects failed. Especially, the four cases in which there was a sub-contracting relationship between the customer and the supplier provide a benchmarking opportunity for customers and suppliers. Each case is different, and they cover a variety of aspects which are present in practice. Therefore the project results and the post-mortem analysis both provide valuable information for both customers and suppliers.

The last implication for practice is the customer-supplier framework which was developed. Education and training are mostly based on situations where software is developed in internal units, through in-house projects, without recognizing commercial boundaries and their influence on software development. Moreover, project management training does not recognize projects as a mean for business. Therefore, the framework provides an alternative outlook on the situation where supplier firms operate.

9.5 Limitations

Some of the limitations have been discussed in the validity sections in Publication I, II, and III, and in Subchapter 9.3 in this dissertation, but some other limitations have to be considered when assessing the results of this research. First, results from the analysis of five cancelled software engineering projects and work experience of the author of this dissertation together guided the rest of the research process. Software development has been approached from a commercial viewpoint, emphasising the business relationship between customers and suppliers. It asserts that this commercial relationship has an influence on software development projects, and concludes that the research community has almost ignored this influence. Because of this strong emphasis on linear view from sales to project closure, which is common in supplier firms, then traditional software engineering process view has almost totally been hidden. However, this process view has not been ignored but it has been understood that the reconciliation of these both approaches would demand a different stream of research. All software engineering processes have been embedded into one category, “Other software life-cycle processes”, which has been mentioned in Publication III while describing development of the customer-supplier framework.

Another limitation is concentration on the project start-up phase rather than the sales phase although results from analysis of five cancelled software engineering projects showed that in three cases out of five serious mistakes were made during the sales phase. Research on the sales phase would have revealed other results.

Furthermore, a definition for software development project success/failure from the supplier’s perspective based on empirical research has not been provided in this dissertation. Not including this empirical part in this dissertation was one of the decisions made during the research process. However, it was learned that understanding of the concepts project success and failure is evolving and research on software development project success/failure from the supplier's perspective has only started.

9.6 Implications for future research

This research raises many opportunities for future research and four of them are as follows:

- The project start-up phase,
- Identifying project success criteria by perspective in sub-contracting situations, and by project type,
- The customer-supplier framework, and
- Distinguishing between perspectives.

The project start-up phase. Potentially the most interesting track is the project start-up phase itself, its influence on the subsequent project, and the interfaces around the phase. Although the project start-up phase is recognized in practice, it is a little known concept in literature. Moreover, only one study was found presenting a study on project start-up, in a high technology innovating and manufacturing company (Halman and Burger, 2002, summarized in Publication III). It was not found in any studies concentrating on the project start-up phase of a software development project. This dissertation provides first insights into the start-up phase of a software development project, especially from the supplier's perspective, and therefore it is discussed here from the supplier's perspective only.

Eleven activities were identified from the interviews. However, the focus of this dissertation was not to build a well-defined start-up process based on the interviewees. Questions such as are some activities more important than others, are there any interrelationships between activities, or are there any causalities between activities need to be investigated further. Moreover, it was found that during the start-up phase it is possible to make such mistakes that result in project cancellation, but the connection between the project start-up phase and project success it not yet known. With the help of the customer-supplier framework it is possible to recognize some subjects which should be examined more closely. These are related to interfaces within the firm and interfaces between the supplier and the customer.

Within the firm there are interfaces before, during, and after the project start-up phase. One of the interfaces within the firm is that between two units, i.e. the sales unit and the unit which takes the responsibility of executing the project, named here as production. At the project level this means an interface between sales team and a certain project manager and the whole project team. This research has not established what information and how is transferred from the sales team to the project manager and to the project team, what difficulties are related to that interface and what should be the best practices in that interface. Moreover, it is not known whether there are difficulties related to organizational differences between two units, sales and production units.

Other interfaces within the firm are as unknown as the interface between sales and production. These are interfaces between the project start-up phase and management, the project start-up and administration, the project start-up and the project, and the start-up and other projects within the supplier firm. The interface between the start-up phase and the project is related to the three-level understanding which should be attained at project manager and at project team levels during the start-up phase in order to enhance project success. It is not known yet what are the problems related to attaining understanding at project manager and at project team levels in sub-contracting situations, but the importance of the subject from the viewpoint of project success was recognized from the interviews.

In addition to these interfaces within the firm, there are interfaces between the supplier and the customer. Two levels were recognized and are presented in the customer-supplier framework. These are interfaces between the supplier and the customer at project level and at customer-supplier relationship level. In addition, these interfaces are related to the three-level understanding which should be attained between the customer and the supplier during the start-up phase in order to enhance project success.

Identifying project success criteria by the perspective in sub-contracting situations and by project type. This dissertation has discussed software development project failure/success in sub-contracting situations, mainly from the supplier's perspective but not forgetting the customer's perspective. By understanding and distinguishing different project types and perspectives and using proper success criteria it is possible to get more relevant results for practitioners who manage different projects in different organizations.

During the research process it became clear that it should be recognized whether software is developed through in-house project or through a commercial project executed by an external supplier. Taylor (2007) says that most typically project management studies in the field of information systems have investigated in-house projects, where the implementation team and the project manager belong to the organization implementing the project. Therefore, the perspective of most studies is organizations whose overall goal of a successful project is to deliver an information system that will support and strengthen the organization's own business, the customer's perspective, not the supplier's perspective which does its' own business enhancing the customer's business.

When software development is carried out by an external supplier, there are two parties which have their own goals, i.e. the aim of the customer is to minimize the costs of the project whereas the aim of the supplier is to maximize the profit (de Wit, 1988). For the supplier, software development projects are a way to do business and therefore the business context has to be taken into account. This has been mentioned by Anda et al. (2009) referring to cost and effort drivers which may not be well described, and by Taylor (2007) who has found that the supplier has additional goals and new risks which are not discussed in literature.

When looking the situation from the customer's perspective there may be a large information systems development project where software development is acquired from an external supplier, and this acquired software has to be implemented into the customer's own information systems. Alternatively, the customer may acquire software which will be integrated into the customer's own software products. Hence, there are two main project types conducted by the customer, named as in-house bespoke software development project and software product development project. In both cases there may be a sub-contracting relationship with one or more suppliers.

The need to distinguish between different projects has been highlighted by Pinto and Mantel (1990) who have found that the factors that predict project failure vary depending upon the project type. This is supported by Müller and Turner (2007) who claim that there are differences between success criteria by project type. Hence, in order to be able to research software development project success/failure project type and in the case of sub-contracting situations, the perspective should also be recognized and business context should be taken into account.

During the research process three project success criteria from the supplier's perspective from seven articles were identified. However, considerable interpretation was required in order to identify these success criteria. Hence, more empirical studies on software development project success or failure in practice are needed before it is possible to gain more knowledge about how suppliers perceive software development project success or failure. Moreover, more rigorous empirical research is needed before it is possible to establish concepts through which success or failure of various types of software development projects from different perspectives can be defined or assessed. When project success criteria are distinguished by perspective in sub-contracting situations, and by project type, it is possible to compare e.g. different software development models, developers' motivation, and risk management between unsuccessful and successful projects.

The customer-supplier framework. The customer-supplier framework provides a research opportunity. For this research the customer-supplier framework has been a means to understand the complicated sub-contracting situation. The framework was partly validated during this research, but more thorough validation should be done by other researchers, as Kitchenham, Linkman and Linkman stated: "*A particular problem is the tendency for experimenters to ignore evidence that contradicts their preconceptions and resolve any ambiguous evidence in favour of their preconceptions.*" (Kitchenham et al., 2005, p. 773). Moreover, the perspective in this research has been the supplier's and therefore the customer's part needs to be extended. Other frameworks or models describing mainly the customer-supplier relationship can be found e.g. in (Bergkvist, 2011; Kern and Willcocks, 2000).

Distinguishing between perspectives. The framework also provides support for distinguishing between perspectives in sub-contracting situations. However, the framework does not define how closely the customer is involved during this phase and how roles and responsibilities are later divided between both parties. Therefore one more research opportunity is to have more empirical research on different cases studying how roles, responsibilities, and actual software development work are divided between both parties in sub-contracting situations from both perspectives especially taking into account process view as defined e.g. in (ISO/IEC 12207, 2008).

The research area of requirements engineering would be closely related to this as this distinction of responsibilities between customer and supplier in sub-contracting situations is important for requirements. It has been noted that at least some requirements are defined already before the project start-up phase but it is not clear who has done that work: the customer or the supplier, sales persons or experts. Moreover, additional work for having more precise requirements is done during the start-up phase and this work continues during the course of the project. Therefore one future research context is requirements engineering, addressing the sub-contracting situation.

The division of roles, responsibilities and actual software development work between customer and supplier in practice has been published in the context of global software development (GSD), e.g. in (Prikladnicki, Audy and Evaristo, 2003; Šmite, 2005; Šmite, 2006). It seems that research on GSD does not yet emphasise properly whether software is developed globally but in-house (e.g. Lasser and Heiss, 2005), or distributed globally between two or more or firms (e.g. in Paasivaara, 2003; Prikladnicki, Audy, Damian and Oliveira, 2007). The difference between intra-organizational projects (in-house projects) and inter-organizational projects (projects with sub-contracting relationships) is addressed in (Poikolainen and Paananen, 2007) and noted also in (Šmite et al., 2010). However, the commercial sub-contracting situation has not yet been generally established as a factor which should be taken into account in GSD research and whose influence on globally distributed software development projects should be examined more closely. Hence, GSD is another research context for future research on the project start-up phase of software development projects and project success from different perspectives.

A third research context where research can be done is outsourcing. It has been observed that there is a lack of studies in the outsourcing literature on software development which consider the supplier's perspective (e.g. Dibbern et al., 2004; Goles and Chin, 2005; Levina and Ross, 2003; Taylor, 2007) and this was found also in Publication II. The newest reported research on software development outsourcing (Khan, Niazi and Ahmad, 2011a; Khan, Niazi and Ahmad, 2011b) strengthens this mainstream of selecting the customer's perspective for the research. Moreover, research focus in this dissertation has been on software development projects, not any IT projects, yearly billed software maintenance work, nor different outsourced support activities. Therefore, in carrying out outsourcing research, a focus could be taken from the supplier's perspective or on the project start-up phase of software development projects from both perspectives.

10 CONCLUSION

The focus of this dissertation has been on software development project failure. Despite software having been successfully applied to almost all possible areas software development projects have a reputation that they fail. Therefore, it seems that we have not yet learned enough to ensure that our software development projects are successful (Cerpa and Verner, 2009). Moreover, if the focus of a specific study has been on identification of various factors that lead to failure, the reason for the failure of a specific project is often ignored (Verner et al., 2008). There have been studies on unsuccessful software development projects which have been publicly available, but thorough studies on failed private-sector everyday projects are rare. Hence, instead of research on large and massive projects, the research for this dissertation concentrated on private-sector everyday projects that are not high profile enough to be reported in the literature or press. The aim of the research was to extend the understanding of software development project failure and to answer the research problem: Why do software development projects fail?

The research problem was approached by conducting a post-mortem analysis of five cancelled software engineering projects. The reasons for cancellation decisions and the causes behind those reasons were identified but other issues emerged. One issue was related to concern about the project failure - whether the failed project was also a failure from the supplier's perspective. Another issue was related to the pre-project phase in sub-contracting situations. Interest in that phase was based on the post-mortem analysis which revealed that in the pre-project phase it is possible to make mistakes that result in project cancellation. The study around the pre-project phase was further divided into two more specific research questions. One research question was related to the problem of how different aspects which are at present in sub-contracting software development projects are connected to each other. Another research question was related to the project start-up phase.

Answers to overall research problem and more specific research questions were sought performing a systematic literature review on software development project failure/success from the supplier's perspective, interviewing project managers in four software engineering firms, and looking for literature on the preceding phases of the project in general but especially on the project start-up phase. Although the main subject of this dissertation was a research on software development project failure, the contribution of this dissertation is wider. The main contributions of this dissertation are as follows:

- **Results from post-mortem analysis of five cancelled software engineering projects.** This dissertation provided a thorough study on five cancelled software engineering projects and has given explanations as to why these projects failed. Each case was different, and they covered a variety of aspects which are present in practice. Therefore, the result of the post-mortem analysis provides valuable information for both customers and suppliers.
- **New answers to old problem of software development failure.** The viewpoint of this research was unusual and new answers were found. An important outcome was the examination of the subject from the supplier's perspective as well as distinguishing the concepts of project success/failure from project management success/failure.
- **Customer-supplier framework.** Using the framework developed during this research it is now possible to discern different perspectives and elements which are present in sub-contracting situations. Therefore, the framework provides support when researching software development projects in sub-contracting situations. Moreover, the framework provides supplier firms with an outline of the situation in which they operate.
- **Highlighting the need for distinguishing between perspectives in sub-contracting situations, between different project types and usage of proper success criteria.** Project managers of both in-house projects and supplier firms need relevant research and valid results. Given perspectives and project types while studying software development projects, this is possible.
- **Project start-up phase.** This research has made the project start-up phase, a very important but previously hidden phase, visible.
- **Practical recommendations to supplier firms were provided.**

In summary, this dissertation provides new information on software development project failures. In addition to having answers to research problem and research questions, this dissertation has shown research gaps not only on the research on software development project failure but also on the preceding phases of software development projects, especially the project start-up phase. Moreover, this research emphasises research on software development projects from the supplier's perspective.

YHTEENVETO (FINNISH SUMMARY)

Yli viidenkymmenen vuoden historiastaan huolimatta ohjelmistoala kärsii epäonnistuvien projektien maineesta. Intensiivisestä projektien hallintaan, riskien hallintaan, menetelmiin ja prosesseihin kohdistuvasta tutkimuksesta huolimatta ohjelmistoprojektit myöhästelevät, ylittävät budjettejaan eikä asiakaskaan ole aina tyytyväinen.

Tässä väitöskirjatutkimuksessa etsitään vastausta kysymykseen: Miksi ohjelmistoprojektit epäonnistuvat? Viiden erilaisen epäonnistuneen ohjelmistoprojektin dokumentaation analyysillä löydettiin syitä kyseisten projektien epäonnistumiselle. Yhdessä tapauksessa, joka oli sisäinen tuotekehitysprojekti, ratkaiseva virhe tehtiin projektin aloituksen jälkeen. Neljässä muussa tapauksessa ratkaisevat virheet tehtiin ennen kuin projekti oli edes alkanut. Näissä neljässä tapauksessa yhteistä oli kaupallinen asiakas-toimittaja -suhde, minkä vuoksi väitöskirjatutkimuksessa keskityttiin vain vastaaviin projekteihin.

Yhdessä edellä mainituista epäonnistuneista projekteista toimittaja onnistui toteuttamaan projektin lähes toivotussa aikataulussa, ei ylittänyt budjettia ja projektin lopputulos oli lähes se, mistä oli sovittu. Tästä huolimatta asiakas ei ottanut järjestelmää käyttöön, vaikka maksoi toimittajan laskut. Asiakkaan näkökulmasta projekti oli epäonnistunut, mutta toimittajan näkökulmasta tilanne ei ollut niin selkeä. Näkökulmien selkeyttämiseksi selvitettiin systemaattisen kirjallisuuskatsauksen avulla, miten ohjelmistoprojektin onnistuminen tai epäonnistuminen on aikaisemmissa tutkimuksissa määritelty toimittajan näkökulmasta. Näkökulma oli sama koko raportoidun väitöskirjatutkimuksen ajan.

Systemaattisessa kirjallisuuskatsauksessa löydettiin vain seitsemän tutkimusta, joissa käsiteltiin ohjelmistoprojekteja toimittajan näkökulmasta. Näistä tutkimuksista tunnistettiin kolme kriteeriä ohjelmistoprojektin onnistumiselle toimittajan näkökulmasta. Tunnistetut onnistumiskriteerit ovat asiakastyytyväisyys ja toimittajan sekä lyhyen että pitkän tähtäimen liiketoiminnallinen menestys. Mikään näistä seitsemästä tutkimuksesta ei määritellyt ohjelmistoprojektin epäonnistumista.

Onnistumiskriteerien lisäksi huomattiin, että ohjelmistoalalla ei ole tavallista erotella projektin onnistumista ja projektin hallinnan onnistumista toisistaan. Koska eroa ei tehdä, on helppo pitää ohjelmistoprojekteja epäonnistuneena.

Neljässä edellä mainitussa analysoidussa tapauksessa kyseessä oli kaupallinen suhde asiakkaan ja toimittajan välillä, millä on vaikutusta projektiin. Koska kaupallisten elementtien huomioiminen ohjelmistoprojekteihin liittyvässä tutkimuksessa on vähäistä, tutkimuksessa kehitettiin asiakas-toimittaja -viitekehys, joka tunnistaa erilaisia elementtejä, jotka ovat läsnä kaupallisessa asiakas-toimittaja -suhteessa. Viitekehyksen avulla on mahdollista tarkastella kokonaisvaltaisesti tilannetta, jossa toimittajalla on vastuu asiakkaalle toteutettavasta ohjelmistoprojektista, tunnistaa eri osapuolia ja siten arvioida ohjelmistoprojekti ja sen onnistumista tai epäonnistumista eri näkökulmista, sekä havaita projektia edeltäviä vaiheita. Tutkimuksen perusteella projektia edeltävistä vaiheista erityisesti projektin käynnistysvaihe näyttää olevan lähes tunnistamaton vaihe ohjelmistoalan kirjallisuudessa.

Tutkimuksen toinen empiirinen osa keskittyi projektia edeltäviin vaiheisiin ja erityisesti projektin käynnistysvaiheeseen. Kirjallisuuden ja tehtyjen haastattelujen perusteella projektin käynnistysvaiheen aloitus kiinnitettiin hetkeen, jolloin asiakas tekee tilauksen tai jollain muulla tavalla ilmoittaa, että tilaa projektin toimittajalta. Käynnistysvaiheen lopetus ja siten projektin virallinen aloitus kiinnitettiin asiakkaan kanssa pidettävään kick-off -tilaisuuteen.

Haastatteluista tunnistettiin 11 projektin käynnistysvaiheen aikana tehtävää aktiviteettia. Lisäksi haastatteluista tunnistettiin pyrkimys saavuttaa ymmärrys projektin sisällöstä kolmella tasolla, jotta projektilla olisi paremmat edellytykset onnistua. Nämä tasot ovat projektipäällikkö, projektiryhmä ja asiakas-toimittaja. Edelleen havaittiin, että toimittajayritykset eivät aktiivisesti pyri määrittelemään projekteille onnistumiskriteereitä.

Mikäli onnistumiskriteereitä ei määritellä, projektiryhmän jäsenet voivat tavoitella omia tavoitteitaan projektin tavoitteiden sijaan. Ilman onnistumiskriteereiden määrittelyä projektin toteutuksen aikana on vaikeampaa tehdä toimenpiteitä, joilla varmistettaisiin projektin onnistuminen. Edellä mainittujen vaikutusten lisäksi projektin onnistumisen arviointi ei ole mahdollista.

Väitöskirjatutkimuksen tavoitteena oli löytää vastauksia kysymykseen, miksi ohjelmistoprojektit epäonnistuvat. Vastaukset tiivistettynä ovat:

- Ohjelmistoprojektit voivat epäonnistua, koska ratkaisevia virheitä tehdään ennen projektin alkua.
- Ohjelmistoprojektin pitäminen epäonnistuneena riippuu näkökulmasta.
- Ohjelmistoprojektia voidaan pitää epäonnistuneena, koska se on epäonnistunut projektin hallinnan näkökulmasta.
- Ohjelmistoprojekti voi epäonnistua siksi, että yritykset eivät näytä aktiivisesti toimivan siten, että projekteille määriteltäisiin onnistumiskriteerit. Jos näitä ei määritellä, muun muassa projektin onnistumisen arviointia ei voida tehdä.

Väitöskirjatutkimuksessa annetaan lukuisia suosituksia toimittajayrityksille ja ehdotetaan useita jatkotutkimusaiheita erityisesti projektin käynnistysvaiheesta ja painottaen toimittajan näkökulmaa.

APPENDIX A: PUBLICATIONS AND AUTHOR’S CONTRIBUTION

This dissertation draws upon two conference papers, two journal papers, and one journal paper, which is under review. In this appendix summary of each of the publications and author’s contribution are presented.

Publication I

Ahonen, J.J, Savolainen, P. 2010. Software engineering projects may fail before they are started: Post-mortem analysis of five cancelled projects. Journal of Systems and Software. 83 (11), 2175–2187.

Abstract

Context: Software project cancellations are often caused by mistakes made during the project, and such cancellations make a strong economic impact. We analyzed five cancelled software engineering projects. One case was an internal product development project of a company that sells products to its customers. The other four cases were different software engineering projects, and outcomes of these projects were planned to be delivered to external customers.

Objective: This study reports a post-mortem analysis of five software engineering projects with the aim of providing more knowledge about the reasons for cancellation decisions and the causes behind those reasons.

Methods: The research method is case study. A method for a document-based post-mortem analysis was developed and post-mortem analysis was performed. All project documentation was available for analysis.

Results: The reasons for the cancellation decisions were well-known ones. In four cases of five, the outcome of the project was to be delivered to an external customer, but in these cases the causes of the cancellation reasons were not found from the normal project documentation. In these cases the cause of the cancellation originated in a phase before the start of the project and therefore the project was doomed before it was started.

Conclusion: It is reasonable to suggest that a remarkable portion of project cancellations are due to mistakes made before the project is started in the case of contract-based software engineering projects

Author’s contribution

The author of this dissertation defined the overall structure of the publication and how to present the results. Ahonen developed the analysis method used based on his review and analysis of post-mortem analysis methods. A person with access rights to the original documentation analyzed the cases. The author of this dissertation jointly with Ahonen performed the final interpretation of the results and wrote the conclusions.

Publication II

Savolainen, P., Ahonen, J. J., Richardson, I. 2011. Software development project success and failure from the supplier's perspective: A systematic literature review. International Journal of Project Management. Accepted for publication.

Abstract

In this paper, we consider software development project success and failure from the supplier’s perspective. First we clarified concepts in order to be able to exclude review articles on in-house projects, continuous services, the customer’s perspective, and software product development, with the aim of providing valid results for supplier firms. We divided success criteria into project success and project management (PM) success, and, in seven articles, identified three success criteria from the supplier’s perspective: customer satisfaction, short-term business benefits, and long-term business benefits. In contrast, no definition of software development project failure was found. Articles were found in seven different journals, showing that knowledge on software development project success from the supplier’s perspective is fragmented. This impedes the growth of knowledge on this topic.

Author’s contribution

The author of this dissertation was one of the persons who searched, selected, and evaluated articles for further review. She and Ahonen were main contributors in the analysis and writing process; Richardson reviewed the publication and made some minor changes.

Publication III

Savolainen, P., Richardson, I. The unknown project start-up and the customer-supplier framework. 9.8.2011. Submitted.

Abstract

Context: An earlier study we conducted revealed that in sub-contracting situation it is possible to make serious mistakes before the actual project has started. In the study presented in this paper, we concentrate on the pre-project phase, particularly the project start-up phase from the supplier’s perspective.
Objective: Our initial aim was to find out what exactly is the pre-project phase of a software development project. During the research process, we focused on the project start-up phase and the activities performed during this phase.
Method: The research method was case study.
Results: During our research, the pre-project phase and the project start-up phase seemed to be an almost non-existing area. Thus, we did not find a published framework which showed these phases connected to a software development project. We therefore developed the customer-supplier framework in order to fill this gap. The framework was developed through analysis of literature and interviews which we conducted in four software engineering firms. We found only one study on the project start-up phase; other articles were descriptions of the phase or the start-up phase was mentioned because of its connection to the main topic of the article. We interviewed 14 project managers and, from these interviews, identified 11 activities in the start-up phase. The resultant framework

provides a unified view of different aspects related to projects when software is developed in sub-contracting situations. These activities 1) show the variety of actions undertaken in supplier firms just before project start, normally a short time period, 2) describe important decisions and choices which are made in a supplier firm from the point of view of the project being undertaken, 3) describe how unsettled the situation may be although the project is about to start, and 4) reveal an interface where knowledge is transferred from the sales team to the project team.
Conclusion: The project start-up phase is an important and complex phase which has almost totally been unrecognized and unexplored phase in software engineering field. The framework which we present fills this significant gap.

Author's contribution
The author of this dissertation planned and conducted the research, developed the framework, and was the main contributor in the writing process. Richardson reviewed the publication and suggested modifications.

Publication IV
Savolainen, P. 2010. Vague Project Start Makes Project Success of Outsourced Software Development Projects Uncertain. In Ali Babar, M., Vierimaa, M., Oivo, M. (Eds.), Product-Focused Software Process Development, Vol. 6156. Berlin Heidelberg: Springer, 351-365.

Abstract
A definition of a project success includes at least three criteria: 1) meeting planning goals, 2) customer benefits, and 3) supplier benefits. This study aims to point out the importance of the definition of the project start, the project start date, and what work should be included in the project effort in order to ensure the supplier's benefits. The ambiguity of the project start risks the profitability of the project and therefore makes project success at least from supplier's point of view uncertain. Moreover, vague project start makes it more difficult to compare project management metrics, such as duration and effort, between projects. There is no clear definition for the project start either in literature or practice. Based on interviews, the definitions are provided for project start, project start date, and project start-up effort included in the project.

Author's contribution
The author of this dissertation was the author of this publication. The publication is based on the research made for Publication III.

Publication V
Savolainen, P., Verner, J. M., Land, L. P. W., Low, G. C. 2011. What happens before a project starts? – Project start-up from the supplier perspective. In Pokorny, J.; Repa, V.; Richta, K.; Wojtkowski, W.; Linger, H.; Barry, C.; Lang, M. (Eds.), Information Systems Development. Springer: New York, 647- 657.

Abstract
Before an outsourced software project officially begins the contracting or supplier organization has already expended effort. Although project start and start-up effort impact on project success in

most cases these are undefined concepts. There are no clear definitions of project start, start-up or the activities that should be completed before project start either in the literature or in practice. Ambiguity around project start sets up risks to the profitability of a project and therefore makes the real success of a project not only uncertain but difficult to measure. A vague project start also makes comparisons between projects and between organizations unreliable. In this paper, we describe a pilot study that reviews project start, project start-up, and project start date, and then investigates what the key activities of the supplier are normally performed by the end of the project start-up phase. We use interviews with software supplier practitioners to define those key activities.

Author's contribution

The publication is based on the research made for Publication III. Verner outlined the publication and the author of this dissertation made analysis of the data. The author of this dissertation, Verner, and Land contributed in the writing process while Low reviewed the publication.

REFERENCES

Anda, B.C.D., Sjøberg, D.I.K., Mockus, A. 2009. Variability and Reproducibility in Software Engineering: A Study of Four Companies that Developed the Same System. IEEE Transactions on Software Engineering. 35 (3), 407-429.

Artto, K., Kujala, J. 2008. Project business as a research field. International Journal of Managing Projects in Business. 1 (4), 469-497.

Artto, K., Wikström, K., Hellström, M., Kujala, J. 2008. Impact of Services on Project Business. International Journal of Project Management. 26 (5), 497-508.

Aundhe, M.D., Mathew, S.K. 2009. Risks in offshore IT outsourcing: A service provider perspective. European Management Journal. 27 (6), 418-428.

Baccarini, D. 1999. The logical framework method for defining project success. Project Management Journal. 30 (4), 25-32.

Bannerman, P.L. 2008. Risk and risk management in software projects. Journal of Systems and Software. 81 (12), 2118-2133.

Barki, H., Rivard, S., Talbot, J. 2001. An Integrative Contingency Model of Software Project Risk Management. Journal of Management Information Systems. 17 (4), 37-69.

Barry, E.J., Mukhopadhyay, T., Slaughter, S.A. 2002. Software Project Duration and Effort: An Empirical Study. Information Technology and Management. 3 (1-2), 113-136.

Beecham, S., Baddoo, N., Hall, T., Robinson, H., Sharp, H. 2008. Motivation in Software Engineering: A systematic literature review. Information and Software Technology. 50 (9-10), 860-878.

Belassi, W., Tukel, O.I. 1996. A new framework for determining critical success/failure factors in projects. International Journal of Project Management. 14 (3), 141-151.

Bergkvist, L. 2011. Conditions Influencing Client–IS Supplier Interactions During IS Outsourcing. In J. Pokorny, V. Repa, K. Richta, W. Wojtkowski, H. Linger, C. Barry, M. Lang (Eds.) Information Systems Development. New York: Springer, 125-136.

Betts, M., Lansley, P. 1995. International Journal of Project Management: A review of the first ten years. International Journal of Project Management. 13 (4), 207-217.

Boehm, B.W. 1991. Software Risk Management: Principles and Practices. IEEE Software. 8 (1), 32-41.

Briner, W., Hastings, C., Geddes, M. 2009. Project Leadership, 2nd edition. UK: Gower Publishing Company.

Cater-Steel, A.P. 2001. Process improvement in four small software companies. In D. Grant, L. Sterling (Eds.) Proceedings of the 13th Australian Software Engineering Conference, 2001. USA:IEEE Computer Society, 262-272.

Cerpa, N., Bardeen, M., Kitchenham, B., Verner, J. 2010. Evaluating logistic regression models to estimate software project outcomes. Information and Software Technology. 52 (9), 934-944.

Cerpa, N., Verner, J.M. 2009. Why Did Your Project Fail?. Communications of the ACM. 52 (12), 130-134.

Charette, R.N. 2005. Why Software Fails?. IEEE Spectrum. 42 (9), 42-49.

Chua, A.Y.K. 2009. Exhuming IT Projects from Their Graves: An Analysis of Eight Failure Cases and Their Risk Factors. Journal of Computer Information Systems. (Spring 2009), 31-39.

Coleman, G., O'Connor, R. 2008. Investigating software process in practice: A grounded theory perspective. Journal of Systems and Software. 81 (5), 772-784.

Collins, A., Baccarini, D. 2004. Project success - A survey. Journal of Construction Research. 5 (2), 211-231.

Cooke-Davies, T. 2002. The "real" success factors on projects. International Journal of Project Management. 20 (3), 185-190.

Cooper, M.J., Budd, C.S. 2007. Tying the pieces together: A normative framework for integrating sales and project operations. Industrial Marketing Management. 36 (2), 173-182.

Cova, B., Holstius, K. 1993. How to Create Competitive Advantage in Project Business. Journal of Marketing Management. 9 (2), 105-121.

Cova, B., Salle, R. 2007. Introduction to the IMM special issue on 'Project marketing and the marketing of solutions' A comprehensive approach to project marketing and the marketing of solutions. Industrial Marketing Management. 36 (2), 138-146.

Crawford, L., Pollack, J., England, D. 2006. Uncovering the trends in project management: Journal emphases over the last 10 years. International Journal of Project Management. 24 (2), 175-184.

Cruzes, D.S., Dybå, T. 2011. Research synthesis in software engineering: A tertiary study. Information and Software Technology. 53 (5), 440-455.

Cusumano, M., MacCormack, A., Kemerer, C.F., Crandall, B. 2003. Software development worldwide: the state of the practice. IEEE Software. 20 (6), 28-34.

de Bakker, K., Boonstra, A., Wortmann, H. 2010. Does risk management contribute to IT project success? A meta-analysis of empirical evidence. International Journal of Project Management. 28 (5), 493-503.

de Wit, A. 1988. Measurement of project success. International Journal of Project Management. 6 (3), 164-170.

Dibbern, J., Goles, T., Hirschheim, R., Jayatilaka, B. 2004. Information systems outsourcing: A survey and analysis of the literature. The DATA BASE for Advances in Information Systems. 35 (4), 6-102.

Dvir, D., Lipovetsky, S., Shenhar, A., Tishler, A. 1998. In search of project classification: a non-universal approach to project success factors. Research Policy. 27 (9), 915-935.

Dvir, D., Raz, T., Shenhar, A.J. 2003. An empirical analysis of the relationship between project planning and project success. International Journal of Project Management. 21 (2), 89-95.

Egginton, B. 1996. The project start-up process – Getting it to work better. Engineering Management Journal. 6 (2), 88-92.

Eisenhardt, K.M. 1989a. Agency Theory: An Assessment and Review. Academy of Management Review. 14 (1), 57-74.

Eisenhardt, K.M. 1989b. Building Theories from Case Study Research. Academy of Management Review. 14 (4), 532-550.

Eisenhardt, K.M., Graebner, M.E. 2007. Theory Building from Cases: Opportunities and Challenges. Academy of Management Journal. 50 (1), 25-32.

Eveleens, J.L., Verhoef, C. 2010. The rise and fall of the Chaos report figures. IEEE Software. 27 (1), 30-36.

Ewusi-Mensah, K. 2003. Software Development Failures: Anatomy of abandoned projects. USA: MIT Press.

Fangel, M. 1984. Planning project start-up. International Journal of Project Management. 2 (4), 242-245.

Fangel, M. 1987. The Essence of Project Start-up. In M. Fangel (Ed.) Handbook of Project Start-Up. Denmark: INTERNET Committee on Project Start-up.

Fangel, M. 1990. (Ed.) Handbook of Project Start-up. Denmark: INTERNET Committee on Project Start-up.

Fangel, M. 1991. To start or to start-up?: That is the key question of project initiation. International Journal of Project Management. 9 (1), 5-9.

Fortune, J., White, D. 2006. Framing of project critical success factors by a systems model. International Journal of Project Management. 24 (1), 53-65.

Gillis, R.B. 1987. Strategies for successful project implementation. In M. Fangel (Ed.) Handbook of Project Start-Up. Denmark: INTERNET Committee on Project Start-up.

Glass, R.L. 1998. Software Runaways. USA: Prentice Hall.

Glass, R.L. 2001. Frequently forgotten fundamental facts about software engineering. IEEE Software. 18 (3), 112,110-111.

Glass, R.L. 2005. IT Failure Rates – 70% or 10-15%?. IEEE Software. 22 (3), 112-111.

Glass, R.L. 2006. The Standish report: does it really describe a software crisis?. Communications of the ACM. 49 (8), 15-16.

Goles, T., Chin, W.W. 2005. Information systems outsourcing relationship factors: detailed conceptualization and initial evidence. The DATA BASE for Advances in Information Systems. 36 (4), 47-67.

Gumm, D.C. 2006. Distribution Dimensions in Software Development Projects: A Taxonomy. IEEE Software. 23 (5), 45-51.

Halman, J.I.M., Burger, G.T.N. 2002. Evaluating effectiveness of project start-ups: an exploratory study. International Journal of Project Management. 20 (1), 81-89.

Hannay, J.E., Sjøberg, D.I.K., Dybå, T. 2007. A Systematic Review of Theory Use in Software Engineering Experiments. IEEE Transactions on Software Engineering. 33 (2), 87-107.

Haried, P., Ramamurthy, K. 2009. Evaluating the success in international sourcing of information technology projects: The need for a relational client-vendor approach. Project Management Journal. 40 (3), 56-71.

Hätönen, J., Eriksson, T. 2009. 30+ years of research and practice of outsourcing – Exploring the past and anticipating the future. Journal of International Management. 15 (2), 142-155.

Holstius, K. 1987. Project Export. Lappeenranta University of Technology.

Ika, L.A. 2009. Project success as a topic in project management journals. Project Management Journal. 40 (4), 6-19.

ISO/IEC 12207. 2008. ISO/IEC 12207:2008: Systems and software engineering — Software life cycle processes. Geneva, Switzerland: ISO/IEC.

Jalkala, A., Cova, B., Salle, R., Salminen, R.T. 2010. Changing project business orientations: Towards a new logic of project marketing. European Management Journal. 28 (2), 124-138.

Jørgensen, M., Moløkken-Østvold, K. 2006. How large are software cost overruns? A review of the 1994 CHAOS report. Information and Software Technology. 48 (4), 297-301.

Jørgensen, M., Shepperd, M. 2007. A systematic review of software development cost estimation studies. IEEE Transactions on Software Engineering. 33 (1), 33-53.

Jugdev, K., Müller, R. 2005. A retrospective look at our evolving understanding of project success. Project Management Journal. 36 (4), 19-31.

Jun, L., Qiuzhen, W., Qingguo, M. 2010. The effects of project uncertainty and risk management on IS development project performance: A vendor perspective. International Journal of Project Management. (2010), in press, doi:10.1016/j.ijproman.2010.11.002.

Kappelman, L.A., McKeeman, R., Zhang, L. 2006. Early warning signs of IT project failure: The dominant dozen. Information Systems Management. 23 (4), 31-36.

Kern, T., Willcocks, L. 2000. Exploring information technology outsourcing relationships: theory and practice. The Journal of Strategic Information Systems. 9 (4), 321-350.

Khan, S.U., Niazi, M., Ahmad, R. 2011a. Barriers in the selection of offshore software development outsourcing vendors: An exploratory study using a systematic literature review. Information and Software Technology. 53 (7), 693-706.

Khan, S.U., Niazi, M., Ahmad, R. 2011b. Factors influencing clients in the selection of offshore software outsourcing vendors: An exploratory study using a systematic literature review. Journal of Systems and Software. 84 (4), 686-699.

Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J., Linkman, S. 2009. Systematic literature reviews in software engineering – A systematic literature review. Information and Software Technology. 51 (1), 7-15.

Kitchenham, B., Charters, S. 2007. Guidelines for performing systematic literature reviews in software engineering. EBSE-2007-01, School of

Computer Science and Mathematics, Keele University, downloaded from the homepage of Information and Software Technology 2nd February 2010.

Kitchenham, B., Linkman, S., Linkman, S. 2005. Experiences of using an evaluation framework. Information and Software Technology. 47 (11), 761-774.

Kujala, J., Artto, K. 2000. Criteria for Project Performance in Business Context. Project Management. 6 (1), 46-53.

Kujala, J., Murtoaro, J., Artto, K. 2007. A Negotiation Approach to Project Sales and Implementation. Project Management Journal. 38 (4), 33-44.

Kujala, S., Artto, K., Aaltonen, P., Turkulainen, V. 2010. Business models in project-based firms – Towards a typology of solution-specific business models. International Journal of Project Management. 28 (2), 96-106.

Lasser, S., Heiss, M. 2005. Collaboration maturity and the offshoring cost barrier: the tradeoff between flexibility in team composition and cross-site communication effort in geographically distributed development projects. In Proceedings of the International Professional Communication Conference, IPCC 2005. Piscataway, N.J.:IEEE, 718-728.

Levina, N., Ross, J.W. 2003. From the vendors perspective: Exploring the value proposition in information technology outsourcing. MIS Quarterly. 27 (3), 331-364.

Linberg, K.L. 1999. Software developer perceptions about software project failure: A case study. Journal of Systems and Software. 49 (2-3), 177-192.

Lipovetsky, S., Tishler, A., Dvir, D., Shenhar, A. 1997. The relative importance of project success dimensions. R&D Management. 27 (2), 97-106.

Mahaney, R.C., Lederer, A.L. 2003. Information systems project management: an agency theory interpretation. Journal of Systems and Software. 68 (1), 1-9.

Mao, J., Lee, J., Deng, C. 2008. Vendors' perspectives on trust and control in offshore information systems outsourcing. Information & Management. 45 (7), 482-492.

Meyer, B. 2006. The unspoken revolution in software engineering. Computer. 39 (1), 124, 121-123.

Morris, P.W.G. 1987. Introductory Paper on Project Start-up. In M. Fangel (Ed.) Handbook of Project Start-Up. Denmark: INTERNET Committee on Project Start-up.

Müller, R., Turner, R. 2007. The influence of project managers on project success criteria and project success by type of project. European Management Journal. 25 (4), 298-309.

Munns, A.K., Bjeirmi, B.F. 1996. The role of project management in achieving project success. International Journal of Project Management. 14 (2), 81-87.

Myers, M.D. 2009. Qualitative Research in Business & Management. UK: SAGE Publications Ltd.

Newton, R. 2009. The practice and theory of project management: creating value through change. UK: Palgrave Macmillan.

Paasivaara, M. 2003. Communication Needs, Practices and Supporting Structures in Global Inter-Organizational Software Development Projects. In Proceedings of the ICSE International Workshop on Global Software Development, GSD'03. 59-63. Available http://gsd2003.cs.uvic.ca/gsd2003proceedings.pdf

Papke-Shields, K.E., Beise, C., Quan, J. 2010. Do project managers practice what they preach, and does it matter to project success?. International Journal of Project Management. 28 (7), 650-662.

Passalacqua, U. 1987. Project Strart-up of Office Automation. In M. Fangel (Ed.) Handbook of Project Start-Up. Denmark: INTERNET Committee on Project Start-up.

Pinto, J.K., Mantel Jr., S.J. 1990. The Causes of Project Failure. IEEE Transactions on Engineering Management. 37 (4), 269-276.

Pinto, J.K., Prescott, J.E. 1990. Planning and tactical factors in the project implementation process. Journal of Management Studies. 27 (3), 305-327.

PMBOK. 2008. A Guide to the Project Management Body of Knowledge, 4th edition. USA: Project Management Institute.

Poikolainen, T., Paananen, J. 2007. Performance Criteria in Inter-Organizational Global Software Development Projects. In Proceedings of the International Conference on Global Software Engineering, ICGSE 2007. Los Alamitos, Calif.:IEEE Computer Society, 60-70.

Pressman, R.S. 2005. Software Engineering: A Practitioner's Approach, 6th edition. USA: McGraw-Hill.

Prikladnicki, R., Audy, J.L.N. 2010. Process models in the practice of distributed software development: A systematic review of the literature. Information and Software Technology. 52 (8), 779-791.

Prikladnicki, R., Audy, J.L.N., Damian, D.de Oliveira, T.C. 2007. Distributed Software Development: Practices and challenges in different business strategies of offshoring and onshoring. In Proceedings of the International Conference on Global Software Engineering, ICGSE 2007. Los Alamitos, Calif.:IEEE Computer Society, 262-274.

Prikladnicki, R., Aundy, J.L.N., Evaristo, R. 2003. Global software development in practice lessons learned. Software Process: Improvement and Practice. 8 (4), 267-281.

Prikladnicki, R., Audy, J.L.N., Shull, F. 2010. Patterns in Effective Distributed Software Development. IEEE Software. 27 (2), 12-15.

Procaccino, J.D., Verner, J.M. 2006. Software project managers and project success: An exploratory study. Journal of Systems and Software. 79 (11), 1541-1551.

Procaccino, J.D., Verner, J.M., Lorenzet, S.J. 2006. Defining and Contributing to Software Development Success. Communications of the ACM. 49 (8), 79-83.

Procaccino, J.D., Verner, J.M., Shelfer, K.M., Gefen, D. 2005. What do software practitioners really think about project success: an exploratory study. Journal of Systems and Software. 78 (2), 194-203.

Reel, J.S. 1999. Critical Success Factors in Software Projects. IEEE Software. 16 (3), 18-23.

Royce, W. 1998. Software project management: A Unified Framework. USA: Addison-Wesley Longman, Inc.

Runeson, P., Höst, M. 2009. Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering. 14 (2), 131-164.

Sadeh, A., Dvir, D., Shenhar, A. 2000. The Role of Contract Type in the Success of R&D Defence Projects Under Increasing Uncertainty. Project Management Journal. 31 (3), 14-22.

Schmidt, R., Lyytinen, K., Keil, M., Cule, P. 2001. Indentifying Software Project Risks: An International Delphi Study. Journal of Management Information Systems. 17 (4), 5-36.

Sharp, H., Baddoo, N., Beecham, S., Hall, T., Robinson, H. 2009. Models of motivation in software engineering. Information and Software Technology. 51 (1), 219-233.

Shenhar, A.J., Levy, O., Dvir, D. 1997. Mapping Dimensions of Project Success. Project Management Journal. 28 (2), 5-13.

Silvasti, J. 1987. Start-up of Small Delivery Projects. In M. Fangel (Ed.) Handbook of Project Start-Up. Denmark: INTERNET Committee on Project Start-up.

Sjøberg, D.I.K., Dybå, T., Anda, B.C.D., Hannay, J.E. 2008. Building Theories in Software Engineering. In F. Shull, J. Singer, D.I.K. Sjøberg (Eds.) Guide to Advanced Empirical Software Engineering. UK: Springer, 312-336.

Sjøberg, D.I.K., Dybå, T., Jørgensen, M. 2007. The Future of Empirical Methods in Software Engineering Research. In L.C. Briand, A.L. Wolf (Eds.) Proceedings of the Future of Software Engineering, FOSE '07. Los Alamitos, CA:IEEE Computer Society, 358-378.

Sjøberg, D.I.K., Hannay, J.E., Hansen, O., Kampenes, V.B., Karahasanovic, A., Liborg, N., Rekdal, A.C. 2005. A survey of controlled experiments in software engineering. IEEE Transactions on Software Engineering. 31 (9), 733-753.

Šmite, D. 2005. A Case Study: Coordination Practices in Global Software Development. In F. Bomarius, S. Komi-Sirviö (Eds.) Product Focused Software Process Improvement, Vol. 3547. Berlin / Heidelberg: Springer, 25-46.

Šmite, D. 2006. Global software development projects in one of the biggest companies in Latvia: is geographical distribution a problem?. Software Process: Improvement and Practice. 11 (1), 61-76.

Šmite, D., Wohlin, C., Gorschek, T., Feldt, R. 2010. Empirical evidence in global software engineering: a systematic review. Empirical Software Engineering. 15 (1), 91-118.

Sumner, M., Bock, D., Giamartino, G. 2006. Exploring the linkage between the characteristics of it project leaders and project success. Information Systems Management. 23 (4), 43-49.

Software Industry Survey 2011. [Accessed 5/26/2011]. Available http://www.softwareindustrysurvey.org/.

Taylor, H. 2005. Congruence between risk management theory and practice in Hong Kong vendor-driven IT projects. International Journal of Project Management. 23 (6), 437-444.

Taylor, H. 2007. Outsourced IT Projects from the Vendor Perspective: Different Goals, Different Risks. Journal of Global Information Management. 15 (2), 1-27.

Themistocleous, G., Wearne, S.H. 2000. Project management topic coverage in journals. International Journal of Project Management. 18 (1), 7-11.

Thomas, G., Fernández, W. 2008. Success in IT projects: A matter of definition?. International Journal of Project Management. 26 (7), 733-742.

Turner, J.R. 2009. The Handbook of Project Based Management, 3rd edition. eBook: McGraw-Hill.

Turner, J.R., Cochrane, R.A. 1993. Goals-and-methods matrix: coping with projects with ill defined goals and/or methods of achieving them. International Journal of Project Management. 11 (2), 93-102.

Verner, J.M., Evanco, W.M. 2005. In-House Software Development: What Project Management Practices Lead to Success?. IEEE Software. 22 (1), 85-93.

Verner, J.M., Sampson, J.Cerpa, N. 2008. What factors lead to software project failure? In C. Rolland (Ed) Proceedings of the 2nd International Conference on Research Challenges in Information Science, RCIS 2008. Piscataway, NJ:IEEE, 71-80.

Wateridge, J. 1998. How can IS/IT projects be measured for success?. International Journal of Project Management. 16 (1), 59-63.

Whittaker, B. 1999. What went wrong? Unsuccessful information technology projects. Information Management & Computer Security. 7 (1), 23-29.

Yin, R.K. 2009. Case study research: design and methods, 4rd edition. USA: Sage Publications.

JYVÄSKYLÄ STUDIES IN COMPUTING

1 Ropponen, Janne, Software risk management - foundations, principles and empirical findings. 273 p. Yhteenveto 1 p. 1999.

2 Kuzmin, Dmitri, Numerical simulation of reactive bubbly flows. 110 p. Yhteenveto 1 p. 1999.

3 Karsten, Helena, Weaving tapestry: collaborative information technology and organisational change. 266 p. Yhteenveto 3 p. 2000.

4 Koskinen, Jussi, Automated transient hypertext support for software maintenance. 98 p. (250 p.) Yhteenveto 1 p. 2000.

5 Ristaniemi, Tapani, Synchronization and blind signal processing in CDMA systems. - Synkronointi ja sokea signaalinkäsittely CDMA järjestelmässä. 112 p. Yhteenveto 1 p. 2000.

6 Laitinen, Mika, Mathematical modelling of conductive-radiative heat transfer. 20 p. (108 p.) Yhteenveto 1 p. 2000.

7 Koskinen, Minna, Process metamodelling. Conceptual foundations and application. 213 p. Yhteenveto 1 p. 2000.

8 Smolianski, Anton, Numerical modeling of two-fluid interfacial flows. 109 p. Yhteenveto 1 p. 2001.

9 Nahar, Nazmun, Information technology supported technology transfer process. A multi-site case study of high-tech enterprises. 377 p. Yhteenveto 3 p. 2001.

10 Fomin, Vladislav V., The process of standard making. The case of cellular mobile telephony. - Standardin kehittämisen prosessi. Tapaustutkimus solukkoverkkoon perustuvasta matkapuhelintekniikasta. 107 p. (208 p.) Yhteenveto 1 p. 2001.

11 Päivärinta, Tero, A genre-based approach to developing electronic document management in the organization. 190 p. Yhteenveto 1 p. 2001.

12 Häkkinen, Erkki, Design, implementation and evaluation of neural data analysis environment. 229 p. Yhteenveto 1 p. 2001.

13 Hirvonen, Kullervo, Towards better employment using adaptive control of labour costs of an enterprise. 118 p. Yhteenveto 4 p. 2001.

14 Majava, Kirsi, Optimization-based techniques for image restoration. 27 p. (142 p.) Yhteenveto 1 p. 2001.

15 Saarinen, Kari, Near infra-red measurement based control system for thermo-mechanical refiners. 84 p. (186 p.) Yhteenveto 1 p. 2001.

16 Forsell, Marko, Improving component reuse in software development. 169 p. Yhteenveto 1 p. 2002.

17 Virtanen, Pauli, Neuro-fuzzy expert systems in financial and control engineering. 245 p. Yhteenveto 1 p. 2002.

18 Kovalainen, Mikko, Computer mediated organizational memory for process control. Moving CSCW research from an idea to a product. 57 p. (146 p.) Yhteenveto 4 p. 2002.

19 Hämäläinen, Timo, Broadband network quality of service and pricing. 140 p. Yhteenveto 1 p. 2002.

20 Martikainen, Janne, Efficient solvers for discretized elliptic vector-valued problems. 25 p. (109 p.) Yhteenveto 1 p. 2002.

21 Mursu, Anja, Information systems development in developing countries. Risk management and sustainability analysis in Nigerian software companies. 296 p. Yhteenveto 3 p. 2002.

22 Seleznyov, Alexandr, An anomaly intrusion detection system based on intelligent user recognition. 186 p. Yhteenveto 3 p. 2002.

23 Lensu, Anssi, Computationally intelligent methods for qualitative data analysis. 57 p. (180 p.) Yhteenveto 1 p. 2002.

24 Ryabov, Vladimir, Handling imperfect temporal relations. 75 p. (145 p.) Yhteenveto 2 p. 2002.

25 Tsymbal, Alexey, Dynamic integration of data mining methods in knowledge discovery systems. 69 p. (170 p.) Yhteenveto 2 p. 2002.

26 Akimov, Vladimir, Domain decomposition methods for the problems with boundary layers. 30 p. (84 p.). Yhteenveto 1 p. 2002.

27 Seyukova-Rivkind, Ludmila, Mathematical and numerical analysis of boundary value problems for fluid flow. 30 p. (126 p.) Yhteenveto 1 p. 2002.

28 Hämäläinen, Seppo, WCDMA Radio network performance. 235 p. Yhteenveto 2 p. 2003.

29 Pekkola, Samuli, Multiple media in group work. Emphasising individual users in distributed and real-time CSCW systems. 210 p. Yhteenveto 2 p. 2003.

30 Markkula, Jouni, Geographic personal data, its privacy protection and prospects in a location-based service environment. 109 p. Yhteenveto 2 p. 2003.

31 Honkaranta, Anne, From genres to content analysis. Experiences from four case organizations. 90 p. (154 p.) Yhteenveto 1 p. 2003.

32 Raitamäki, Jouni, An approach to linguistic pattern recognition using fuzzy systems. 169 p. Yhteenveto 1 p. 2003.

33 Saalasti, Sami, Neural networks for heart rate time series analysis. 192 p. Yhteenveto 5 p. 2003.

34 Niemelä, Marketta, Visual search in graphical interfaces: a user psychological approach. 61 p. (148 p.) Yhteenveto 1 p. 2003.

35 You, Yu, Situation Awareness on the world wide web. 171 p. Yhteenveto 2 p. 2004.

36 Taatila, Vesa, The concept of organizational competence – A foundational analysis. - Perusteanalyysi organisaation kompetenssin käsitteestä. 111 p. Yhteenveto 2 p. 2004.

JYVÄSKYLÄ STUDIES IN COMPUTING

37 Lyytikäinen, Virpi, Contextual and structural metadata in enterprise document management. - Konteksti- ja rakennemetatieto organisaation dokumenttien hallinnassa. 73 p. (143 p.) Yhteenveto 1 p. 2004.

38 Kaario, Kimmo, Resource allocation and load balancing mechanisms for providing quality of service in the Internet. 171 p. Yhteenveto 1 p. 2004.

39 Zhang, Zheying, Model component reuse. Conceptual foundations and application in the metamodeling-based systems analysis and design environment. 76 p. (214 p.) Yhteenveto 1 p. 2004.

40 Haarala, Marjo, Large-scale nonsmooth optimization variable metric bundle method with limited memory. 107 p. Yhteenveto 1 p. 2004.

41 Kalvine, Viktor, Scattering and point spectra for elliptic systems in domains with cylindrical ends. 82 p. 2004.

42 Dementieva, Maria, Regularization in multistage cooperative games. 78 p. 2004.

43 Maaranen, Heikki, On heuristic hybrid methods and structured point sets in global continuous optimization. 42 p. (168 p.) Yhteenveto 1 p. 2004.

44 Frolov, Maxim, Reliable control over approximation errors by functional type a posteriori estimates. 39 p. (112 p.) 2004.

45 Zhang, Jian, Qos- and revenue-aware resource allocation mechanisms in multiclass IP networks. 85 p. (224 p.) 2004.

46 Kujala, janne, On computation in statistical models with a psychophysical application. 40 p. (104 p.) 2004.,

47 Solbakov, Viatcheslav, Application of mathematical modeling for water environment problems. 66 p. (118 p.) 2004.

48 Hirvonen, Ari P., Enterprise architecture planning in practice. The Perspectives of information and communication technology service provider and end-user. 44 p. (135 p.) Yhteenveto 2 p. 2005.

49 Vartiainen, Tero, Moral conflicts in a project course in information systems education. 320 p. Yhteenveto 1p. 2005.

50 Huotari, Jouni, Integrating graphical information system models with visualization techniques. - Graafisten tietojärjestelmäkuvausten integrointi visualisointitekniikoilla. 56 p. (157 p.) Yhteenveto 1p. 2005.

51 Wallenius, Eero R., Control and management of multi-access wireless networks. 91 p. (192 p.) Yhteenveto 3 p. 2005.

52 Leppänen, Mauri, An ontological framework and a methodical skeleton for method engineering – A contextual approach. 702 p. Yhteenveto 2 p. 2005.

53 Matyukevich, Sergey, The nonstationary Maxwell system in domains with edges and conical points. 131 p. Yhteenveto 1 p. 2005.

54 Sayenko, Alexander, Adaptive scheduling for the QoS supported networks. 120 p. (217 p.) 2005.

55 Kurjenniemi, Janne, A study of TD-CDMA and WCDMA radio network enhancements. 144 p. (230 p.) Yhteenveto 1 p. 2005.

56 Pechenizkiy, Mykola, Feature extraction for supervised learning in knowledge discovery systems. 86 p. (174 p.) Yhteenveto 2 p. 2005.

57 Ikonen, Samuli, Efficient numerical methods for pricing American options. 43 p. (155 p.) Yhteenveto 1 p. 2005.

58 Kärkkäinen, Kari, Shape sensitivity analysis for numerical solution of free boundary problems. 83 p. (119 p.) Yhteenveto 1 p. 2005.

59 Helfenstein, Sacha, Transfer. Review, reconstruction, and resolution. 114 p. (206 p.) Yhteenveto 2 p. 2005.

60 Nevala, Kalevi, Content-based design engineering thinking. In the search for approach. 64 p. (126 p.) Yhteenveto 1 p. 2005.

61 Katasonov, Artem, Dependability aspects in the development and provision of location-based services. 157 p. Yhteenveto 1 p. 2006.

62 Sarkkinen, Jarmo, Design as discourse: Representation, representational practice, and social practice. 86 p. (189 p.) Yhteenveto 1 p. 2006.

63 Äyrämö, Sami, Knowledge mining using robust clustering. 296 p. Yhteenveto 1 p. 2006.

64 Ifinedo, Princely Emili, Enterprise resource planning systems success assessment: An integrative framework. 133 p. (366 p.) Yhteenveto 3 p. 2006.

65 Viinikainen, Ari, Quality of service and pricingin future multiple service class networks. 61 p. (196 p.) Yhteenveto 1 p. 2006.

66 Wu, Rui, Methods for space-time parameter estimation in DS-CDMA arrays. 73 p. (121 p.) 2006.

67 Parkkola, Hanna, Designing ICT for mothers. User psychological approach. – Tieto- ja viestintätekniikoiden suunnittelu äideille. Käyttäjäpsykologinen näkökulma. 77 p. (173 p.) Yhteenveto 3 p. 2006.

68 Hakanen, Jussi, On potential of interactive multiobjective optimization in chemical process design. 75 p. (160 p.) Yhteenveto 2 p. 2006.

69 Puttonen, Jani, Mobility management in wireless networks. 112 p. (215 p.) Yhteenveto 1 p. 2006.

70 Luostarinen, Kari, Resource , management methods for QoS supported networks. 60 p. (131 p.) 2006.

71 Turchyn, Pavlo, Adaptive meshes in computer graphics and model-based simulation. 27 p. (79 p.) Yhteenveto 1 p.

72 Zhovtobryukh, Dmytro, Context-aware web service composition. 290 p. Yhteenveto 2 p. 2006.

JYVÄSKYLÄ STUDIES IN COMPUTING

73 Kohvakko, Nataliya, Context modeling and utilization in heterogeneous networks. 154 p. Yhteenveto 1 p. 2006.

74 Mazhelis, Oleksiy, Masquerader detection in mobile context based on behaviour and environment monitoring. 74 p. (179 p). Yhteenveto 1 p. 2007.

75 Siltanen, Jarmo, Quality of service and dynamic scheduling for traffic engineering in next generation networks. 88 p. (155 p.) 2007.

76 Kuuva, Sari, Content-based approach to experiencing visual art. - Sisältöperustainen lähestymistapa visuaalisen taiteen kokemiseen. 203 p. Yhteenveto 3 p. 2007.

77 Ruohonen, Toni, Improving the operation of an emergency department by using a simulation model. 164 p. 2007.

78 Naumenko, Anton, Semantics-based access control in business networks. 72 p. (215 p.) Yhteenveto 1 p. 2007.

79 Wahlstedt, Ari, Stakeholders' conceptions of learning in learning management systems development. - Osallistujien käsitykset oppimisesta oppimisympäristöjen kehittämisessä. 83 p. (130 p.) Yhteenveto 1 p. 2007.

80 Alanen, Olli, Quality of service for triple play services in heterogeneous networks. 88 p. (180 p.) Yhteenveto 1 p. 2007.

81 Neri, Ferrante, Fitness diversity adaptation in memetic algorithms. 80 p. (185 p.) Yhteenveto 1 p. 2007.

82 Kurhinen, Jani, Information delivery in mobile peer-to-peer networks. 46 p. (106 p.) Yhteenveto 1 p. 2007.

83 Kilpeläinen, Turo, Genre and ontology based business information architecture framework (GOBIAF). 74 p. (153 p.) Yhteenveto 1 p. 2007.

84 Yevseyeva, Iryna, Solving classification problems with multicriteria decision aiding approaches. 182 p. Yhteenveto 1 p. 2007.

85 Kannisto, Isto, Optimized pricing, QoS and segmentation of managed ICT services. 45 p. (111 p.) Yhteenveto 1 p. 2007.

86 Gorshkova, Elena, A posteriori error estimates and adaptive methods for incompressible viscous flow problems. 72 p. (129 p.) Yhteenveto 1 p. 2007.

87 Legrand, Steve, Use of background real-world knowledge in ontologies for word sense disambiguation in the semantic web. 73 p. (144 p.) Yhteenveto 1 p. 2008.

88 Hämäläinen, Niina, Evaluation and measurement in enterprise and software architecture management. - Arviointi ja mittaaminen kokonais- ja ohjelmistoarkkitehtuurien hallinnassa. 91 p. (175 p.) Yhteenveto 1 p. 2008.

89 Ojala, Arto, Internationalization of software firms: Finnish small and medium-sized software firms in Japan. 57 p. (180 p.) Yhteenveto 2 p. 2008.

90 Laitila, Erkki, Symbolic Analysis and Atomistic Model as a Basis for a Program Comprehension Methodology. 321 p. Yhteenveto 3 p. 2008.

91 Nihtilä, Timo, Performance of Advanced Transmission and Reception Algorithms for High Speed Downlink Packet Access. 93 p. (186 p.) Yhteenveto 1 p. 2008.

92 Setämaa-Kärkkäinen, Anne, Network connection selection-solving a new multiobjective optimization problem. 52 p. (111p.) Yhteenveto 1 p. 2008.

93 Pulkkinen, Mirja, Enterprise architecture as a collaboration tool. Discursive process for enterprise architecture management, planning and development. 130 p. (215 p.) Yhteenveto 2 p. 2008.

94 Pavlova, Yulia, Multistage coalition formation game of a self-enforcing international environmental agreement. 127 p. Yhteenveto 1 p. 2008.

95 Nousiainen, Tuula, Children's involvement in the design of game-based learning environments. 297 p. Yhteenveto 2 p. 2008.

96 Kuznetsov, Nikolay V., Stability and oscillations of dynamical systems. Theory and applications. 116 p. Yhteenveto 1 p. 2008.

97 Khriyenko, Oleksiy, Adaptive semantic Web based environment for web resources. 193 p. Yhteenveto 1 p. 2008.

98 Tirronen, Ville, Global optimization using memetic differential evolution with applications to low level machine vision. 98 p. (248 p.) Yhteenveto 1 p. 2008.

99 Valkonen, Tuomo, Diff-convex combinations of Euclidean distances: A search for optima. 148 p. Yhteenveto 1 p. 2008.

100 Sarafanov, Oleg, Asymptotic theory of resonant tunneling in quantum waveguides of variable cross-section. 69 p. Yhteenveto 1 p. 2008.

101 Pozharskiy, Alexey, On the electron and phonon transport in locally periodical waveguides. 81 p. Yhteenveto 1 p. 2008.

102 Aittokoski, Timo, On challenges of simulation-based globaland multiobjective optimization. 80 p. (204 p.) Yhteenveto 1 p. 2009.

103 Yalaho, Anicet, Managing offshore outsourcing of software development using the ICT-supported unified process model: A cross-case analysis. 91 p. (307 p.) Yhteenveto 4 p. 2009.

104 K ollanus, Sami, Tarkastuskäytänteiden kehittäminen ohjelmistoja tuottavissa organisaatioissa. - Improvement of inspection practices in software organizations. 179 p. Summary 4 p. 2009.

105 Leikas, Jaana, Life-Based Design. 'Form of life' as a foundation for ICT design for older adults. - Elämälähtöinen suunnittelu. Elämänmuoto ikääntyville tarkoitettujen ICT tuotteiden ja palvelujen suunnittelun lähtökohtana. 218 p. (318 p.) Yhteenveto 4 p. 2009.

JYVÄSKYLÄ STUDIES IN COMPUTING

106 Vasilyeva, Ekaterina, Tailoring of feedback in web-based learning systems: Certitude-based assessment with online multiple choice questions. 124 p. (184 p.) Yhteenveto 2 p. 2009.

107 Kudryashova, ElenaV., Cycles in continuous and discrete dynamical systems. Computations, computer assisted proofs, and computer experiments. 79 p. (152 p.) Yhteenveto 1 p. 2009.

108 Blackledge, Jonathan, Electromagnetic scattering and inverse scattering solutions for the analysis and processing of digital signals and images. 297 p. Yhteenveto 1 p. 2009.

109 Ivannikov, Andriy, Extraction of event-related potentials from electroencephalography data. - Herätepotentiaalien laskennallinen eristäminen EEG-havaintoaineistosta. 108 p. (150 p.) Yhteenveto 1 p. 2009.

110 kalyakin, Igor, Extraction of mismatch negativity from electroencephalography data. - Poikkeavuusnegatiivisuuden erottaminen EEG-signaalista. 47 p. (156 p.) Yhteenveto 1 p. 2010.

111 Heikkilä, Marikka, Coordination of complex operations over organisational boundaries. 265 p. Yhteenveto 3 p. 2010.

112 Fekete, Gábor, Network interface management in mobile and multihomed nodes. 94 p. (175 p.) Yhteenveto 1 p. 2010.

113 Kujala, Tuomo, Capacity, workload and mental contents - Exploring the foundations of driver distraction. 146 p. (253 p.) Yhteenveto 2 p. 2010.

114 Lugano, Giuseppe, Digital community design - Exploring the role of mobile social software in the process of digital convergence. 253 p. (316 p.) Yhteenveto 4 p. 2010.

115 Kampylis, Panagiotis, Fostering creative thinking. The role of primary teachers. - Luovaa ajattelua kehittämässä. Alakoulun opettajien rooli. 136 p. (268 p.) Yhteenveto 2 p. 2010.

116 Toivanen, Jukka, Shape optimization utilizing consistent sensitivities. - Muodon optimointi käyttäen konsistentteja herkkyyksiä. 55 p. (130p.) Yhteenveto 1 p. 2010.

117 Mattila, Keijo, Implementation techniques for the lattice Boltzmann method. - Virtausdynamiikan tietokonesimulaatioita Hila-Boltzmann -menetelmällä: implementointi ja reunaehdot. 177 p. (233 p.) Yhteenveto 1 p. 2010.

118 Cong, Fengyu, Evaluation and extraction of mismatch negativity through exploiting temporal, spectral, time-frequency, and spatial features. - Poikkeavuusnegatiivisuuden (MMN) erottaminen aivosähkönauhoituksista käyttäen ajallisia, spektraalisia, aikataajuus - ja tilapiirteitä. 57 p. (173 p.) Yhteenveto 1 p. 2010.

119 Liu, Shenghua, Interacting with intelligent agents. Key issues in agent-based decision support system design. 90 p. (143 p.) Yhteenveto 2 p. 2010.

120 Airaksinen, Tuomas, Numerical methods for acoustics and noise control. - Laskennallisia menetelmiä akustisiin ongelmiin ja melunvaimennukseen. 58 p. (133 p.) Yhteenveto 2 p. 2010.

121 Weber, Matthieu, Parallel global optimization Structuring populations in differential evolution. - Rinnakkainen globaalioptimointi. Populaation rakenteen määrittäminen differentiaalievoluutiossa. 70 p. (185 p.) Yhteenveto 2 p. 2010.

122 Väärämäki, Tapio, Next generation networks, mobility management and appliances in intelligent transport systems. - Seuraavan sukupolven tietoverkot, liikkuvuuden hallinta ja sovellutukset älykkäässä liikenteessä. 50 p. (111 p.) Yhteenveto 1 p. 2010.

123 Viukari, Leena, Tieto- ja viestintätekniikkavälitteisen palvelun kehittämisen kolme diskurssia. - Three discourses for an ICT-service development . 304 p. Summary 5 p. 2010.

124 Puurtinen, Tuomas, Numerical simulation of low temperature thermal conductance of corrugated nanofibers. - Poimutettujen nanokuitujen lämmönjohtavuuden numeerinen simulointi matalissa lämpötiloissa . 114 p. Yhteenveto 1 p. 2010.

125 Hiltunen, Leena, Enhancing web course design using action research . - Verkko-opetuksen suunnittelun kehittäminen toimintatutkimuksen keinoin . 192 p. Yhteenveto 2 p. 2010.

126 Aho, Kari, Enhancing system level performance of third generation cellular networks through VoIP and MBMS services. 121 p. (221 p.). Yhteenveto 2 p. 2010.

127 Häkkinen, Markku, Why alarms fail. A cognitive explanatory model. 102 p. (210 p.). Yhteenveto 1 p. 2010.

128 Pennanen, Anssi, A graph-based multigrid with applications. - Graafipohjainen monihilamenetelmä sovelluksineen. 52 p. (128 p.). Yhteenveto 2 p. 2010.

129 Ahlgren, Riikka, Software patterns, organizational learning and software process improvement. 70 p. (137 p.). Yhteenveto 1 p. 2011.

130 Nikitin, Sergiy, Dynamic aspects of industrial middleware architectures 52 p. (114 p.). Yhteenveto 1 p. 2011.

131 Sindhya, Karthik, Hybrid Evolutionary Multi-Objective Optimization with Enhanced Convergence and Diversity. 64 p. (160 p.). Yhteenveto 1 p. 2011.

132 MALI, OLLI, Analysis of errors caused by incomplete knowledge of material data in mathematical models of elastic media. 111 p. Yhteenveto 2 p. 2011.

133 MÖNKÖLÄ, SANNA, Numerical Simulation of Fluid-Structure Interaction Between Acoustic and Elastic Waves. 136 p. Yhteenveto 2 p. 2011.

134 PURANEN, TUUKKA, Metaheuristics Meet Metamodels. A Modeling Language and a Product Line Architecture for Route Optimization Systems. 270 p. Yhteenveto 1 p. 2011.

135 MÄKELÄ, JUKKA, Mobility Management in Heterogeneous IP-networks. 86 p. (145 p.) Yhteenveto 1 p. 2011.

136 SAVOLAINEN, PAULA, Why do software development projects fail? Emphasising the supplier's perspective and the project start-up. 81 p. (167 p.) Yhteenveto 2 p. 2011.