THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Towards Managing the Interaction between Manufacturing and Development Organizations in Automotive Software Development

JOAKIM PERNSTÅL

Department of Computer Science & Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2013

Towards Managing the Interaction between Manufacturing and Development Organizations in Automotive Software Development
JOAKIM PERNSTÅL
ISBN 978-91-7385-834-2

Doktorsavhandling vid Chalmers tekniska högskola
Ny serie nr. 3515
ISSN 0346-718X

Technical Report No. 94D
Department of Computer Science and Engineering
Division of Software Engineering
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Telephone + 46 (0)31-772 1000

Contact information:
Volvo Car Corporation
Dept 81451, PV33
SE-405 31 Göteborg, Sweden
Telephone: +46 31 3250242
jpernsta@volvocars.com

Department of Computer Science and Engineering
Chalmers University of Technology
Hörsalsvägen 11
SE-412 96 Göteborg, Sweden
pernstal@chalmers.se

Chalmers Reproservice
Göteborg, Sweden 2013.

Abstract

The dramatic increase and importance of software in automotive systems has created many challenges not only in the development organization, but also in the manufacturing organization. This thesis focuses on the inter-departmental interaction between these two organizations in large-scale development of software-intensive automotive systems. The overall goal of this thesis is to identify the most prevalent challenges in this area, and develop and evaluate possible scalable, efficient and effective solutions addressing said challenges. The thesis work was conducted at two Swedish automotive companies using empirical research methods.

An important contribution of this thesis are the assessment and improvement analysis and planning methods developed in order to meet industrial needs to effectively and systematically conduct process improvements in large-scale software development, while focusing on inter-departmental interaction. This included the tailoring of previously developed lightweight methods, and developing and applying novel approaches for root cause analysis and modeling and analysis of dysfunctional organizational communication patterns. Lessons learned and feedback from practitioners showed that the methods were helpful and useful to enhance the performance of the process improvement initiatives.

Through the application of the assessment methods this thesis identified major challenges related to requirements engineering (RE). The substantial growth of automotive software has led to an increased need for coordination and communication of requirements and associated possible solutions across the manufacturing and development organizations. The main reasons for these challenges are lack of bidirectional communication and the information being communicated is insufficiently specified.

The large amount of specifications also demands effective and efficient ways to enable and improve coordination and communication. To this end a lightweight RE framework was developed, called BRASS. It combines goal oriented requirements communication with lean based concurrent engineering, and promotes communication over achieving perfection in specifications. The applicability of BRASS was evaluated through an industrial validation, which showed that BRASS can be tailored and applied in industry and the practitioners perceived the use of BRASS as useful and effective.

Keywords: Software engineering, manufacturing engineering, automotive software development, Requirements Engineering, Software process improvement, Empirical.

Contents

Acknowledgements

I would like to take the opportunity to express my gratitude to all those who were in some way involved in this research project.

First and foremost, I would like to thank my industrial supervisor Dan Florén, who is the primarily initiator of this research project, for believing in me and giving me the opportunity to become an industrial PhD student.

Second, but not less, I would like to thank my main supervisor Tony Gorschek for adopting a “lost-in-space-PhD student”. In close collaboration with my co-supervisor Robert Feldt, Tony managed with a “full-speed-space-machine-approach” to guide the space ship on course. Both Tony’s and Robert’s exceptional knowledge and experience of research in the complex field of software engineering, and excellent grasp of the grammar of the English language have been invaluable.

I would like to thank my examiner Johan Karlsson for his support throughout my doctoral adventure. I also want to thank Peter Öhman, who supervised me through the initial phases.

I also owe Nader Asnafi and Anders Carlsson special thanks for arranging the financing and anchoring this research project in the manufacturing organization at Volvo Cars Corporation (VCC). Here, I would also like to thank the Volvo Cars Industrial PhD Program for enabling research activities by employees at VCC.

I am also grateful to the friendly and inspiring colleagues of the HOSE research group, Emil Börjesson, Ali Sharohkni, Pariya Kashfi, and Ana Magazinius. Special thanks go to Ana Magazinius for her invaluable support in the initial studies that were carried out. I really enjoyed working with you and I will never forget your interesting ideas and our discussions that have inspired me to broaden my perspective from an engineering level to a scientific view.

My gratitude also to all the anonymous individuals, who took part in the interviews and workshops and contributed valuable data. Special thanks to Björn Villing, Andreas Lundberg, Mats Tharing, and Åke Ljungdahl for allowing me to carry out the studies in the organizations.

I want to extend my appreciation to my colleagues at Department 81451. Despite their sometimes high work load, they understand and accept my almost non-operative role as an industrial PhD.

Finally, a thanks to my family, Helena, Jenny and Frida, for supporting me in rain and shine and putting up with my sometimes absent-minded presence, and last but not least reminding me about things that are more important than any research or title.

List of Publications

Appended Papers

The work presented in this thesis is based on the following papers.

Chapter 2
J. Pernstål, A. Magazinius, T. Gorschek, A Study Investigating Challenges in The Interface Between Product Development and Manufacturing in The Development of Software-Intensive Automotive Systems. International Journal of Software Engineering and Knowledge Engineering 22(7): 965-1004, 2012.

Chapter 3
J. Pernstål, T. Gorschek, R. Feldt, D. Florén, Software Process Improvement Planning in Inter-Departmental Development of Software-Intensive Automotive Systems—A Case Study, submitted to 14th International Conference of Product Focused Software Development and Process Improvement - PROFES, 2013.

Chapter 4
J. Pernstål, T.Gorschek, R.Feldt, D.Florén, Flex-RCA—A Lean Based Method for Root Cause Analysis in Software Process, submitted to Journal of Software: Evolution and Process, 2013.

Chapter 5
J. Pernstål, R.Feldt, T.Gorschek, D.Florén, Communication Problems in Software Development—A Model and its Industrial Application, submitted to Empirical Software Engineering Journal, 2013.

Chapter 6
J.Pernstål, T. Gorschek, R. Feldt, The Lean Gap: A Review of Lean Approaches to Large-Scale Software Systems Development, submitted to Journal of Systems and Software, 2013.

Chapter 6
J. Pernstål, T.Gorschek, R.Feldt, D.Florén, Requirements Communication and Balancing in Large-Scale Software-Intensive Product Development, submitted to Requirements Engineering journal, 2013.

Other Papers

Paper 1
J. Pernstål, A. Magazinovic, P. Öhman, A case study of the interaction between development and manufacturing organizations with a focus on software engineering in automotive industry, *Conference on Software Engineering Research and Practice in Sweden* (SERPS), 2007

Paper 2
J. Pernstål, A. Magazinovic, P. Öhman, Manufacturing Engineering Challenges In Product Development Of Software Based Systems In Automotive Industry: A Case Study, *The 18th International Conference on Flexible Automation and Intelligent Manufactuing* (FAIM), 2008.

Paper 3
J. Pernstal, A. Magazinius, P. Öhman, A multiple case study investigating the interaction between manufacturing and development organizations in automotive software engineering, *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement* (ESEM '08), 2008.

Paper 4
J. Pernstal, P. Öhman, An Investigation of manufacturing Challenges in Automotive Software Development from a Lean Perspective, *The 19th International Conference on Flexible Automation and Intelligent Manufacturing* (FAIM 2009), 2009.

Paper 5
A. Magazinius, J. Pernstal, P. Öhman, Prerequisites for Software Coste Estimation in Automotive Development Projects - A Case Study, *Research and Practice in Sweden* (SERPS'07)(2007.

Paper 6
A. Magazinius, J. Pernstal, P. Öhman, Any other cost estimation inhibitors*?", Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement* (ESEM '08), 2008.

Paper 7
A. Magazinius, J. Pernstal, P. Öhman, Software Cost Estimation Inhibitors – A Case Study in Automotive Industry, *Journal of Software Maintenance and Evolution: Research and Practice*, 22(5), 2010.

CHAPTER 1

Introduction

Towards Managing the Interaction between Manufacturing and Development Organizations in Automotive Software Development

1.1 Overview

Software continues to become a substantial component in several industries (e.g., automotive, aerospace and telecommunication systems). Many of them are facing major challenges as large-scale product development is becoming increasingly software-intensive. In these industries competences covering electronics and software engineering (SE) have in recent years become equally important as traditional skills in hardware development.

Software-intensive systems are commonly developed in the context of large-scale development (i.e. systems of systems development), where software constitutes only one, but important, part of the whole. In such development there are many challenges [1-3]. A major challenge is to meet the increasingly needs for communication and coordination between organizational units caused by the complex and interdependent nature of software development [4, 5].

The necessity of achieving a well-functioning interaction between the departments of Product development (PD) and Manufacturing (MAN) throughout the development cycle is central in large scale product development, and in particular in automotive development [3, 6-8].

The overall goal of this thesis is to provide new insights and solutions to improve inter-departmental interaction in industries developing software-intensive systems. Two main objectives are addressed in this thesis with a focus on the automotive domain: (1) how can process assessment and improvement be systematically and effectively performed in large-scale software-intensive system development with a focus on inter-departmental interaction, and (2) finding out what the core challenges are in the inter-departmental interaction between PD and MAN in development of software-intensive automotive systems and how they can be improved.

1.2 Background and Related Work

1.2.1 Research Setting

Typically, large-scale product development projects are complex with a high degree of interdependency between many different departments and engineering disciplines. As the relative contribution and size of software increases in such development, the entire complexity escalates, introducing new challenges to the development organizations [1, 9]. The effectiveness of coordination and both formal and informal communication within and across organizational boundaries have been recognized as one of the most critical challenges [4, 5]. Requirements engineering (RE), and in particular requirements communication, is also commonly identified as crucial and plays a vital role for the success of such development efforts [1, 10, 11].

To achieve efficient development and manufacturing of new products a well-functioning inter-departmental coordination and communication between PD and

MAN has been recognized in research and industry as a main challenge [3, 6-8]. This, in order to prevent problems related to the fit of new product technologies (e.g., navigation, infotainment, engine control, and active safety systems in a vehicle) and the manufacturing processes producing the products (e.g., vehicles).

In order to improve the performance in manufacturing and product development processes, many industries developing large-scale software-intensive products (e.g., automotive and aerospace) have implemented lean manufacturing (LM) and started to adopt some inherent principles and practices of lean product development (LPD) (e.g., cross-functional teams, concurrent engineering visual management, and continuous improvement (Kaizen)) [6-8, 12]. LM focuses on achieving efficient manufacturing systems where removing waste in the manufacturing processes, i.e. everything that does not contribute to the creation of value for the customer is central [13]. In LPD the main focus is on removing waste in the product development processes, but also to make full use of LM implementations [8].

This thesis focuses on the inter-departmental interaction between PD and MAN in development of software-intensive automotive systems. PD is concerned with the design and development of software-intensive automotive systems (e.g., development of power train and chassis control systems for vehicles). MAN is concerned with the manufacturing of vehicles with a focus on the manufacturing operations affected by software-intensive automotive systems. The inter-departmental interaction studied in this thesis includes all the phases of development, from concept (e.g., exploration of requirements and solutions) to design, implementation and validation, but also manufacturing involving pre-production verification and validation of the manufacturing processes (see Chapter 2 and cf., for example, [14-16]). In a new car model project, these activities commonly span over three to four years [14, 15]. The work presented in this thesis focuses on the inter-departmental interaction between PD and MAN, since it has been shown to be critical in the automotive domain while it is of general interest for SE in understanding the collaboration between different organizational units. The PD and MAN interface is of particular interest since these two types of units have been characterized as being different both culturally and personality-wise [17].

The research presented in this thesis was performed as a process improvement initiative at two Swedish automotive companies, namely Volvo Car Corporation (VCC) and Volvo Truck Corporation (VTC). VCC is a premium car manufacturing company and has approximately 22,000 employees all over the world and produces roughly 450,000 cars per year (2011) [18]. VTC is a global automotive company that focuses on the development and production of medium and heavy-duty trucks. The number of employees is about 17,000 and approximately 75,000 trucks are produced in 16 countries (2010) [19]. Both companies are organized as matrix organizations and use a traditional plan-based approach including a stage gate model for governing the development of the complete car and use the V-model [20] to present an overview of design and verification of inherent software-intensive systems. Requirements are mainly specified in written text and administrated by computer-aided RE tools (see Chapter 7). This is a typical setting in the automotive industry [1, 8, 11, 21].

1.2.2 Challenges

Five challenges (C1-C5) are primarily addressed in this thesis, which are divided into two main categories. The first category concerns challenges in the practical application of software process improvement (SPI) methods in large-scale software-intensive systems development focusing on inter-departmental interaction. In response to industry needs of systematically and effectively conducting process improvement initiatives, and based on SPI challenges identified in earlier work, the most relevant challenges (C1-C3) to the research presented in this thesis are described in Section 1.2.2.1. The second category includes the challenges in the inter-departmental interaction between PD and MAN in development of software-intensive-automotive systems identified as most critical in the research project. Section 1.2.2.2 describes these challenges (C4 and C5) and related work. The five challenges are linked to the contributions of this thesis and set its main focus.

1.2.2.1 SPI Challenges

Several general challenges for the application of SPI have been identified in the literature (e.g., [22-27]). The most relevant ones addressed in this thesis are based on the main needs raised by the companies involved. This comprises the SPI methods' ability to suit different settings, to enable effective resource utilization, to define clear goals and problems, and to measure the effects of improvements. The following describes and exemplifies the SPI challenges in relation to the SPI frameworks presented in Section 1.2.3.

C1—SPI initiation threshold

Attaining effective and efficient use of the resources required to conduct SPI initiatives is challenging. Traditional SPI frameworks, such as the Capability Maturity Model Integration (CMMI) [28] and ISO/IEC 15504 [29] (a.k.a. SPICE), are often criticized for being too extensive and resource consuming to implement [22]. Commencing SPI requires long-term commitment of resources that are specifically dedicated for the SPI effort as the expectation that SPI will happen in addition to regular work is an often mentioned problem [24]. A typical assessment-improvement cycle using CMM [30] can take anything from 18 to 24 months [31]. This means a long time to return on investment (TTROI), which is a major concern in small and medium-sized enterprises (SMEs) [22], but also in larger organizations with separate cost center [32].

Consequently, to achieve SPI in software companies with limited resources dedicated to SPI efforts, lightweight SPI frameworks have been proposed in earlier work. Most of these frameworks have been applied to SMEs and many of them adopt a prescriptive approach (see Section 1.2.3.1) using elements from CMMI and ISO/IEC 15504 [33] (e.g., MESOPyME [34], ASPE-MSC [35], and IDEAL [36]). In addition, there are lightweight frameworks taking an inductive approach (see Section 1.2.3.2), such as the IMPACT project [37] inspired by the Quality Improvement Paradigm

(QIP) [38]. IFLAP (improvement Framework utilizing Light weight Assessment and improvement Planning) [39] and its predecessors [40, 41] are other examples, which have also been proven to be scalable and useful in industries developing software on a large-scale, for example, in the automotive domain [39, 42].

C2—Tailorability of SPI methods
Aligning SPI initiatives with company's strategies, goals, size and setting is a major challenge which requires tailoring of the SPI methods. Many SPI frameworks adopt a one-size-fits-all view across organizations and projects and no special consideration is given to specific organizational settings and needs. Prescriptive frameworks guide the order in which practices in the reference model should be implemented. This can lead to that unnecessary practices are implemented or issues that are mission-critical for the companies strategies, business goals and contextual setting are omitted [23, 27]. CMMI and ISO/IEC 15504 are also domain specific as they primarily address process improvement in systems engineering and SE, and have limited capability of being an enabler for successful implementation of domain-specific technologies in related domains (e.g., manufacturing engineering) on which systems engineering and SE are dependent [43]. Furthermore, many software organizations have recently started to adopt agile and lean methods to increase development speed and reduce waste. Even though, for example, eXtreme Programming (XP) [44] addresses many CMMI Levels 2 and 3 practices [45], agile methods are in general difficult to assess with traditional SPI frameworks (e.g., CMMI and ISO/IEC 15504).

Even if improvements can target identified problem areas based on specific organizational needs in CMMI through continuous representation (see Section 1.2.3.1), it still prescribes the order of what practices should be improved/added in each process area as it still is prescriptive in nature [46]. Inductive frameworks (e.g. QIP and iFLAP), on the other hand, base the improvements on an in-depth understanding of the industrial setting assessed, and thus the order of implementing the improvements can be tailored to match organizational needs. However, they rely on the experiences and knowledge of the organization and there is not a set of predefined practices, implying that the improvements may be limited to the maturity of the organization assessed.

C3—SPI Goals and Measurements
A critical challenge is to clearly define and formulate problems and expected goals, and to measure the effects of the SPI initiative in order to determine whether it was successful. Proper preparations and planning, and a clear focus on what is critical to improve and how are keys to achieve this [24, 47]. However, a majority of the SPI frameworks are on a high level (e.g., CMMI, ISO/IEC 15504 and QIP) providing guidance for what to do, but not how the actual implementation should be done. For example, when assessing and planning improvements possibilities in the context of large-scale software-intensive systems development, the complexity of such development often result in high-level improvement issues (items of problems that should be improved) (see [39] and Chapters 2 and 3). However, irrespective of SPI framework used, they give little further guidance on how to decompose the issues into

focused and defined sub-problems, and uncover their causes and detail what is most critical to improve (see Chapters 4 and 5).

In order to demonstrate whether an SPI initiative has been successful, the ability of measuring its effects is crucial [25, 26]. Prescriptive frameworks measure this through assessing the gap between the practices used in an organization and a set of predefined best practices, while inductive frameworks like QIP analyzes the used practices ability to achieve collected and defined goals for successful project performance (e.g., through GQM). For measuring the success of SPI efforts different performance indicators can be used (e.g., productivity, reduction of cost, and improved quality) [48, 49]. However, this is not an easy task as indicators are given different priorities in different companies, but also within companies, and there are several confounding factors influencing the measurements [49-51]. Besides, studies reporting on SPI initiatives often lack complete description of the context in which they have been performed and evaluated, lessening the possibility to judge whether the presented solutions can be successfully adopted in other settings [52].

1.2.2.2 Challenges in the PD and MAN interface

The first part of the research project resulted in several challenges (key improvement issues) (see Chapter 2). Of these, requirements engineering (RE) were shown to be the central and most critical one (see Chapter 3), and was broken down into two core challenges (see Chapters 4, 5, and 7). The following briefly describes them and related work.

C4—Coordinating and communicating enlarged requirements and solution space

The automotive companies face the challenge of managing the high complexity of their organizational structure while the uncertainty and interdependency of requirements and development tasks are increasing. Usually, to attain economy of scale the development work is organized in complex matrix systems with dual-hierarchical form where development tasks are performed in several project teams, which are staffed with hundreds of people from several parts of the functional organization. The complexity is elevated as the product development process is accelerated by performing upstream and downstream development tasks concurrently in order to minimize time-to market [8, 53]. In addition, *the substantial growth of software increases the interdependences between requirements and development tasks and their uncertainty, leading to major challenges in coordinating and communicating a much larger requirements and solution space within and across departments* [1, 9, 11]. Cross departmental coordination and communication challenges are not unique for the automotive industry as it has been recognize in development of other large software-intensive systems [4, 5, 10], and in general product development projects (e.g., [53-55]).

C5—Achieving effective requirements and solution specifications

Attaining sufficient specification quality is critical for successful coordination and communication between the many organizational parts involved in automotive development projects [1, 11]. However, in such projects an enormous amount of product documentation must be handled requiring a lot of resources. Typically, luxury cars are built of over 300 systems and 2500 functions, and can be produced in huge amount of variants [1]. For designing and integrating these systems and functions into a complete vehicle and managing the variants, a large amount of product documentation, mainly consisting of requirements in natural language (about 100 000 textual requirements in VCC), must be administrated [11]. In addition, the documentation must be maintained and updated during the lifecycle of the vehicle [1, 11]. This causes large overhead costs and, for example, it is estimated that if the number of variants is doubled, the overhead costs rise between 20 and 30 percent because of increased complexity [9]. *For achieving improved coordination communication within and across departments, a critical challenge is to efficiently manage specifications of requirements and their solutions while it is vital to ensure the quality of them.*

1.2.3 Software Process Improvement

In order to increase efficiency and quality when creating software-intensive systems, both industry and researchers have acknowledged the importance of SPI and continuous assessment and improvement of processes and practices [39-41, 56, 57]. Most SPI frameworks are rooted in Shewhart-Deming's PDCA (Plan-Do-Check-Act) paradigm, which emphasizes the need of cyclic and continuous process improvement. They can be divided into two main categories: top-down or prescriptive frameworks and bottom-up or inductive frameworks.

1.2.3.1 Prescriptive Frameworks

Prescriptive frameworks are the most widely used in industry. The top-down approach entails that an organization's processes are compared with some framework of predefined and standardized best-practices and the improvement planning is guided by eliminating the gaps between an existing process and a standard one. Examples of widely used prescriptive frameworks, in particularly development of software intensive systems, are CMMI [28] and ISO/IEC 15504 [29].

CMMI combines three models: Systems Engineering Capability Model (SECM), Integrated Capability Maturity Model (IPD-CMM), and Capability Maturity Model for software (CMM-SW). CMMI focuses on maturity levels and the base reference model comprises 22 key process areas (KPAs) (e.g., requirements development and requirements management). In addition, there are three process areas that cover integrated product and process development (IPPD) (e.g., Integrated Project Management for IPPD) and one that covers supplier sourcing. The maturity is defined by five levels: initial, repeatable, defined, managed and optimized and the KPAs are structured into the different levels of maturity. To improve current practices in an organization, CMMI guide further improvement initiatives by focusing on the KPAs on the next level of process maturity. The appraisal can be based on a staged representation aiming to achieve an overall increase in organizational maturity by strictly guiding the order of practices that should be improved. A continuous representation can also be used, allowing targeted improvements in specific process areas that do not conform to the order given by the staged representation.

When assessing an organization with CMMI, the official appraisal method is called Standard CMMI Appraisal Method for Process Improvement (SCAMPI) [58][59]. In general, there are three classes of appraisal: (1) Class A—full and comprehensive appraisal, (2) Class B—focusing on specific process areas and less rigorous assessment than Class A, and (3) Class C—often called a "quick look" on specific areas and the least rigorous class. The estimated effort for the assessment team is 800 – 1600 person hours for Class A, 80 – 640 person hours for Class B, and 60 – 80 person hours for Class C [58][59]. In addition, there is time spent by the staff being assessed. It is only the full assessment (CMMI Class A) that results in a maturity rating. If the continuous representation is used for an appraisal, the rating is a capability level profile. If the staged representation is used for an appraisal, the rating is a maturity level (e.g., maturity level 3) rating [28].

While CMMI provides one reference model with which organizations are compared to, the ISO/IEC 15504 sets requirements on the reference models and assessment models used. In particular, the ISO standards for systems lifecycle processes ISO/IEC 15288 [60] and software life cycle processes ISO/IEC 12207 [61] are used. To suit domain specific needs tailored variants of the reference models have been developed, such as Automotive SPICE [62]. The maturity level of the process assessed is determined through continuous representation, including six capability levels. To achieve a capability level, a number of characteristics gathered in process attributes (PAs) should be fulfilled. An external assessor driving the assessment is required and the effort for the assessors ranges from 33 to 824 person hours [63].

1.2.3.2 Inductive Frameworks

Inductive frameworks take a bottom-up approach which assumes that process change must be driven by specific organizational needs based on comprehensive understanding of the current processes [64]. Thus the improvements are based on the experiences and knowledge from using established processes in projects rather than a pre-defined set of best practices.

Basili's QIP [38] is an example of a well-known inductive framework basing improvements on experiences from executing processes in projects. It consists of two closed-loop cycles: a project cycle that provides feedback to the project executing the suggested improvements, and an organizational cycle that drives the SPI efforts throughout the organization.

LPD can also be classified as an inductive framework as its core principle of continuous improvement emphasizes the value of building and diffusing knowledge in product development by turning new experiences and knowledge from projects into standards that can be reused in subsequent projects [65]. In practice, lean companies commonly accomplish this by establishing know-how databases evolved from checklists [8, 65].

IFLAP [39] and its predecessors [40, 41] are examples of lightweight inductive frameworks that have been developed in order to provide systematic and practical guidance for SPI projects in software companies with limited resources. Primarily based on the inherent knowledge of organizations, iFLAP aims to uncover the most critical issues and establish a realistic implementation plan for organizations. The iFLAP process consists of three main consecutive steps: (1) Selection—which includes the selection of relevant cases such as organizations, projects and roles for the assessment; (2) Process assessment (PA)—which embodies data collection and analysis by using multiple data sources such as interviews and documents that are triangulated, yielding a set of confirmed improvement issues; and 3) Improvement planning (IP)—which involves the prioritization and dependency mapping of the confirmed improvement issues that generate packages of improvement issues and outlines the agenda for what should be improved first.

1.2.4 Organizational Communication and Coordination

In organization theory, it is well established that increased interdependency between development tasks leads to higher demands on effective coordination and communication of information [53-55]. Galbraith [55] argued that the need of information processing increases as a function of increasing complexity, uncertainty, and interdependence of work-flows.

Division of labor occurs prior to coordination, where development work is divided into tasks and sub-units (individuals, groups, and departments are assigned to each task [66]. To reduce the interdependences between requirements and development tasks, the idea of modularization has been promoted in systems engineering [67] and SE [68]. It is a useful approach for dividing the development of complex products into independent requirements and development tasks. However, the growth of software in many products (e.g., vehicles) has led to that previous independent functions, sub-systems and components, but also several engineering fields and departments, must now interact with each other [1, 9]. This has lessened the possibilities to modularize and thus prompting the need of coordination and communication within and across organizational units in product development projects. In particular, attaining effective inter-departmental interaction between PD and MAN has been identified as critical (see Section 1.2.1). The following sections present barriers and mechanisms affecting the interaction between PD and MAN identified in previous work.

1.2.4.1 Coordination and Communication Barriers between PD and MAN

Effective inter-departmental interaction between PD and MAN is difficult to achieve owing to a number of generic barriers [17, 69]. Vandevelde and Van Dierdonk [17] examined earlier studies on the PD and MAN interface and summarized and divided these barriers into personality, cultural, language, physical, and organizational barriers. Table 1 shows some personality and cultural differences between PD and MAN. Personality differences are, for example, that MAN is primarily working with well-defined tasks and is output-oriented while PD often works with more abstract tasks and aspires knowledge development. Cultural differences are, for example, that the degree of structure is higher in MAN than in PD and MAN has shorter time horizon than PD.

When communicating information, the specific language and the level of detail used by different departments and disciplines can cause language barriers. In addition, communicating intangible or non-standardized information usually increases the difficulties in exchanging information between people with different backgrounds.

Table 1 Personality and cultural differences between PD and MAN adopted from [17]

Barrier	Characteristics	PD	MAN
Personality	Goals and aspirations	Knowledge as a source of value to mankind. Research for research's sake. Peer evaluation and recognition.	Delivering quality/volume on time. Minimizing waste and scrap. Clear tasks, relevant to senior management.
	Needs	Autonomy, creative environment Peer recognition. Education, personal development. Support for advancing knowledge in society.	Analyzable, transparent tasks. Increased organizational status. Organizational recognition.
	Motivation	Service to mankind Publications, patent, and professional recognition,. Freedom to solve problems and advance knowledge.	Rewards and sanctions systems for the production volume, quality, and flexibility.
Cultural	Time orientation	Long	Short
	Projects preferred	Advanced	Incremental
	Ambiguity tolerance	Low	Low
	Departmental structure	Low	High
	Bureaucratic orientation	Less	High
	Orientation to others	Permissive	Less permissive
	Professional orientation	Science	Process

Organizational barriers concern unwillingness to change (e.g., adoption of new technologies and processes in MAN) and lack of clarity of goals and, roles and responsibilities within PD and MAN and in the boundaries between them.

Long physical distance and badly designed work places are examples of physical barriers. Furthermore, when development is geographically distributed, the communication gets more complicated (e.g., manufacturing units are spread over the world).

The above mentioned barriers hinders effective inter-departmental communication and sharing of knowledge and experiences between PD and MAN, which in turn leads to misunderstandings of each other, but also deviating interpretations of common corporate goals, capabilities and solutions. Several mechanisms affecting the interplay between PD and MAN have been identified in earlier work.

1.2.4.2 Coordination and Communication Mechanisms between PD and MAN

Mechanisms and approaches explicitly addressing coordination and communication between PD and MAN have been suggested in earlier work, but the empirical evidence of their practical impact is limited [17].

Trygg [70] divides the mechanisms into technological mechanisms (e.g., new components, tools, production methods, and IT-based support tools) and organizational mechanisms including people, structure and culture.

Adler [71] presents a list of coordination mechanisms that were ordered in a descriptive taxonomy based on four generic coordination approaches: (1) standards, (2) schedules, (3) mutual adaptation and (4) teams. To deal with the dynamic conditions in product development projects over time, the taxonomy distinguishes the coordination possibilities in three different temporal phases: (1) pre-project, (2) design and (3) manufacturing. For example, the taxonomy suggests that standards such as design rules assuring manufacturability, schedules (e.g.; sign-off procedures through which MAN accepts responsibility for making a product to the design specifications), mutual adaptations such as manufacturability design reviews, and teams, (e.g., early manufacturing involvement through cross-functional teams), are appropriate coordination mechanisms in the design phase. However, the cost efficiency of using different coordination mechanisms is dependent on the degree of uncertainty of the fit between product and process parameters.

Vandevelde and Van Dierdonk [17] claim that formalization and empathy on the part of PD towards MAN are contributors to a “smooth start of production”. Formalization entails clear goals, roles and responsibilities and empathy means that the product developers consider manufacturing aspects during the design stage. Similarly, Nihtilä [3] and Lakemond et al. [72] emphasize the need for formalization and empathy and observed that such as early and active involvement of MAN, balanced recruitment between PD and MAN, and continuous communication are critical factors. An interesting conclusion in [3] is that due to the increased amount of software in products, there is emerging need for integrating software development operations to the project as a whole, indicating an important direction in future research.

Many lean principles and practices originating from LPD have had a strong influence on approaches that have been developed to reinforce organizational communication between PD and MAN. Establishing cross-functional teams, consisting of members representing different departments (e.g., MAN and PD) and roles (e.g., design and manufacturing engineers) is a core practice in LPD [8, 12]. The purpose is to intensify the communication, rather than coordinating development tasks and departments, groups and individuals. Karlsson and Åhlström [12] found benefits of cross-functional teams, such as better product solution and improved communication of project information to other department than PD, but also difficulties in creating and maintaining them. One main reason for this was that other departments than PD had difficulties in selecting and allocating required resources to actively work with the projects, and in particular, in the beginning of the projects

In order to elevate active cross-departmental development work, practices such as integrated problem-solving [7, 53] and Set Based Concurrent Engineering (SBCE) [4] are used in LPD. Integrated problem-solving and SBCE emphasize the need of communication between upstream and downstream members of a development project (e.g., design and manufacturing engineers).

Attaining simple visual communication through effective visual management is one of the most critical success factors in concurrent development [8, 53]. For this, such as trade-off curves and simple decision matrices, visualizing and communicating evaluations of requirements and alternative solutions are used.

Both SBCE, integrated problem-solving, and visual management have been successfully applied to primarily hardware development in the automotive domain. However, it is unclear how the practices address the fact that software as an artifact, and SE as a discipline, are becoming a central component in the products developed ([73] and see Chapter 6).

1.2.5 Requirements Communication and Coordination in Software Development

Several approaches have been proposed to resolve requirements coordination and communication by either using techniques for enhancing requirements specification quality, standards prescribing best requirements practices, or intensifying requirements communication.

1.2.5.1 Requirements Specification Quality

The requirements specification quality is a critical success factor in large-scale software development projects [1, 10, 11]. Therefore, adopting effective techniques and standards (e.g., IEEE std 830-1998 [74]) that help to produce clear and precise requirements (e.g., correct, unambiguous, complete, consistent, and verifiable) is considered as crucial. Requirements can be documented in several forms, such as use-cases, requirements modeling [75, 76], and formal specifications [77], but specification in natural language is most common. However, in practice, specifying precise and understandable requirements for large and complex systems is impossible to achieve [78]. For example, Weber and Weisbrod [11] found that automotive development is too complex to be managed by just textual requirements. Techniques for modeling and validating requirements have also been proposed (e.g., Unified Modeling Language (UML) [79], Matlab/Simulink and Hardware In the Loop (HIL). However, model-based development and testing of software-intensive automotive systems are in their infancy. For example, Broy et al. [1] point out that, because of the lack of a formalized modeling language, modeling is only applied to certain steps in product development projects. Insufficient integration possibilities, such as linking engineering data to models and compatibility between different tools, also need also to be resolved.

Furthermore, the cost of improving the specification quality of complex systems through detailing textual requirements or modeling is likely to be high, and must be evaluated in relation to the benefits [1, 9].

1.2.5.2 Standardization

To reach a more standardized and mature RE process, there are several well-known SPI frameworks, such as CMMI and ISO/IEC 15504. These frameworks provide high level guidance regarding what to do, but do not detail how the actual implementation should proceed and no special consideration for specific organizational needs are given [22, 31]. For example, the KPA of requirements development in CMMI only prescribe to use proven models and perform risk analysis when analyzing requirements to balance stakeholder needs and constraints. Furthermore, these standards look too narrowly into the engineering aspects within a single project, which is no guarantee for a successful product as project measures (e.g., level of requirements fulfillment) are only the first level to consider in RE [80]. As mentioned above, there are also standards and guidelines for specifying requirements. In addition, there are standards for requirements and design of software-intensive automotive systems, such as Automotive Open System Architecture (Autosar) [81] and ISO 14229 [82]. Even though more standards are used, they allow variability and it is estimated that currently 90 percent of the software must be changed from one generation of vehicles to the next [1].

1.2.5.3 Requirements Communication

Lean and agile software development methods, for example, Lean Software Development (LSD) [83] and Scrum [84] promote coordination and communication of requirements through practices, such as daily Scrum, product backlogs, story cards, and screen mock-ups. There are only a few studies on lean and agile RE. Building on visual management, Peterson and Wohlin [85] measured and displayed the number of requirements, revealing undesired behavior of the development (e.g., bottlenecks). Ramesh et al [86] found that obtaining intensive communication is the most important factor for successful agile RE. However, if there are communication breakdowns caused by, for example, rapid turnover of personnel and growing complexity of the products, the minimization of design specifications can give problems, such as lack inability to scale the software, evolve the application over time and inducting new members into the development team. This is line with Salvonen et al [87], who found that highly skilled people especially in RE is a prerequisite when implementing agile development of embedded systems. Even though organizational communication is central in lean and agile methods, and they are also influencing more traditional industries, such as the automotive, the transition to more lean and agile methods have only started and is not yet widespread in many large-scale SE contexts ([73, 88], and see Chapter 6).

A pragmatic technique called Handshaking with implementation proposals developed by Fricker et al. [89] is the only method found that explicitly aims at intensifying requirements communication. It organizes the requirements communication into a goal seeking element (program manager) and a goal-implementation element (development team). The handshaking process consists of three main phases: (1) taking position, (2) negotiation and (3) confirming agreement. With promising results, the handshaking was applied to a large-scale industrial setting for improving inter-departmental requirements communication problems between marketing and PD (product managers and development teams). In this setting the roles of goal seeker (product manager) and implementer (developer) are rather clear and static over time. However, in other settings the roles the actors play may vary depending on, such as the reason for the communication and what is communicated, and can shift during the development cycle (see Chapters 5 and 7).

1.2.6 Summarizing Challenges

The challenges together with the industry needs identified and the chapters in this thesis that address these challenges are summarized in Table 2.

Table 2 Challenges in relation to industry needs identified and chapters in the thesis

Challenge	Industry need	Chapter
C1—SPI initiation threshold	The effort required to use the SPI method should be reasonable in relation to ROI as the case companies involved had limitations in allocating resources for the SPI project.	2,3,and 4
C2—Tailorability of SPI methods	Given the specific research setting investigated in this thesis (see Section 1.2.1), the SPI method should be possible to tailor and provide detailed guidance on how it was conducted.	2,3,4,and 5
C3—SPI goals and measurements	The SPI method should support planning and detailing of what are most important and beneficial to improve based on organizational needs, and provide measurable indicators for evaluating of the effects of the solutions implemented.	4 and 5
C4—Coordinating and communicating enlarged requirements and solution space.	Identifying and implementing solutions to improve coordination and communications of requirements and alternative solutions between PD and MAN.	6 and 7
C5—Achieving effective requirements and solution specifications	Identifying and implementing solutions to improve the quality of the information being communicated between PD and MAN required for clarifying trade-offs between requirements and alternative solutions.	6 and 7

1.3 Research Method

This section presents the research questions and describes the research process and research methods applied to the studies included in this thesis.

1.3.1 Research Questions

The automotive industry is facing a tremendous growth in the engineering of software-intensive systems, giving rise to various challenges. Typically, these systems are developed in the context of large-scale software development. To ensure alignment of new software technologies in vehicles with the manufacturing processes, a well-functioning interaction between the departments of PD and MAN is crucial. Empirical research focusing on the PD and MAN interface in development of software-intensive automotive systems has not been found. This led to RQ1, which was posed in order to explore and identify the most challenging issues experienced by industry professionals, and to detail and select focus for research.

RQ1: What areas are most important software engineering challenges in the interface between PD and MAN in the automotive domain?

Studies on process improvement initiatives explicitly targeting the process area of inter-departmental interaction in large-scale software development are very few. Furthermore, a majority of the SPI frameworks are too extensive or provide guidance at a high level that does not match the specific needs of an organization. Even though inductive and lightweight frameworks have emerged to help practitioners identify and package improvement issues based on specific organizational needs, the packaged improvement issues are usually on a high level, indicating the symptoms, but not the causes and the core needs of the organization. This gave rise to RQ2

RQ2: How can the challenges identified through a process improvement initiative be prioritized and analyzed to reflect the core needs of an organization?

There are several studies on Lean Product Development (LPD) at Japanese automakers (e.g., [6-8, 12]) that have had a strong influence on approaches that have been developed to reinforce communication and coordination across departments in the product development process. Several of them can be explicitly associated with inter-departmental interaction between PD and MAN (e.g., [6, 7]). Even though lean principles and practices have been translated to the context of software development [83] it was not clear to what extent they have been applied and studied in large-scale software development [73, 88].

To solve the core challenges and needs identified in RQ1, there was an expressed industrial need of evaluating the strength of evidence and potential industrial value of state of the art building on LPD with a focus on large-scale software development. This led to RQ3.

RQ3: Can current lean practices alleviate the core challenges (RQ1) identified?

State of the art explicitly addressing how the core challenges answering RQ1 can be resolved in an efficient and effective way is limited (see Chapters 2, 6 and 7). This gave rise to RQ4.

RQ4: How can the core challenges identified (RQ1) be efficiently and effectively resolved?

1.3.2 Research Process Overview

Technology transfer from research to industry is a problem in itself in SE research as many of the studied problems are formulated in academia and not relevant for industry, and thus the industrial applicability of the results is limited [90]. In order to produce more industry-relevant research results, there is a need of close cooperation and collaboration between industry and academia throughout the entire research process [91, 92].

The research process used in this thesis follows the technology transfer model [91], which is a holistic research cycle model with an aim at transferring research results into industrial use. This line of thought has been adopted by other research models as well. In information systems development research, Mathiassen [93] propose a framework called Collaborative Practice Research (CPR). Overall, CPR adopts an action research approach and suggests a way to organize and conduct applied research based on close collaboration between practitioners and researchers. It has been successfully applied in SPI projects involving practitioners from software organizations and researchers from universities and technology institutes [93]. In design engineering research, Blessing and Chakrabarti [94] present a design research methodology framework (DRM). DRM emphasizes the necessity of developing and implementing solution based on a thorough understanding of real problems and the critical factors influencing the criteria for success through rigorous descriptive studies.

In this thesis, the technology transfer model was mainly chosen because it has been successfully applied to industries developing large-scale software-intensive systems [91, 95]. An overview of the model and its seven steps and are shown in Figure 1.

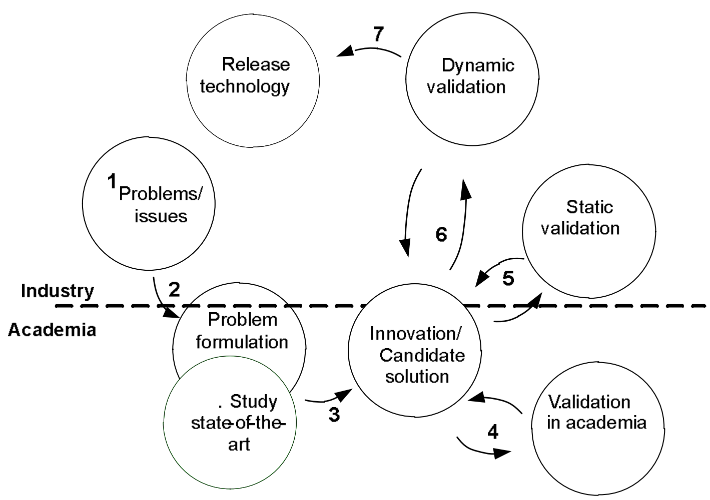

Figure 1 Research process overview adopted from [91].

The following briefly describes each step and how they have been followed in this thesis.

Step 1, overall research directions and questions are outlined based on initial problems and issues identified in industry. Step 1 involved a process assessment (PA), investigating the inter-departmental interaction between PD and MAN in development of software-intensive systems in real industrial settings at VCC and VTC. Chapter 2 presents the results in terms of nine key improvement issues and analyzes them in relation to state of the art, setting the baseline for the improvement planning (IP) in Step 2.

Step 2, planning improvements through formulating problems and a research agenda based on core organizational needs, and studying current knowledge and state of the art in related research fields and domains. The IP included three main studies: (1) prioritizing, dependency mapping, and packaging of the improvement issues and (2) root cause analysis (RCA) of the results of the packaging, and (3) postmortem analysis and modeling of communication problems between PD and MAN. Chapter 3 presents and evaluates the methodology for packaging the improvement issues, and reports the results. Chapter 4 describes and evaluates the RCA method used, and reports the results of applying it to the packaging. For the postmortem modeling and analysis of communication problems between PD and MAN, a novel model was developed. The model and its industrial application are reported in Chapter 5.
Studying state of the art consisted of two parts. The first part is reported in Chapter 2 where the key improvement issues identified are analyzed in relation to relevant state of the art. The second part includes a systematic mapping study (see Section 1.3.3.4), investigating to what extent LPD and its principle and practices have been used and evaluated in large-scale software development. This study is reported in Chapter 6.

Step 3, formulating candidate solutions in cooperation with industry and by using contributing state-of-the-art. Based on the core challenges identified and relevant state of the art, a framework called BRASS (Balancing Requirements and Solution Space and its basic constituents were developed and formulated in close cooperation with VCC. BRASS is presented in Chapter 7.

Step 4, for assuring that the solution(s) addresses and resolves the identified problems, initial validation of the candidate solutions are carried out in academia through practical tests in a lab environment. This step is not included in the thesis.

Step 5, performing static validation involves two-way communication between the practitioners and researchers where the candidate solutions are discussed and analyzed in regular meetings, seminars and workshops. The main part of the static validation was performed in regular meetings and seminars with line and program managers, and engineers in PD and MAN at VCC where BRASS was designed and tailored based on the core needs and experiences of the company. The tailoring of BRASS and its industrial application at VCC is reported in Chapter 7.

Step 6, dynamic validation entails validation of the solution in pilot projects representing real situations with minimized risks. BRASS was evaluated through a dynamic validation at VCC. For this three real cases (balancing issues) in an ongoing new car development project, mirroring VCCs specific problems in communicating and balancing requirements and solutions across the departments of PD and MAN, were used. The results of the validation are reported in Chapter 7.

Step 7, based on the results from the previous static and dynamic validations, the implementation ready solutions are released and deployed in Step 7. VCC has decided to conduct further validations of BRASS in order to provide extensive decision support for moving to a full-scale implementation of BRASS.

1.3.3 Research Methods Utilized

This thesis primarily adopts an empirical research approach. There are a number of research methods for conducting empirical research where the most common ones used in the SE discipline are case studies, surveys, and experiments [96]. In information systems research, action research has also been used [93]. In this thesis, case studies, surveys and action research were used. In addition, relevant literature were reviewed through an systematic mapping study [97, 98].

1.3.3.1 Survey

In general, surveys are carried out by selecting a sample which is representative from the population to be studied and the results are analyzed through statistical methods from which descriptive and explanatory conclusions are derived [99].

Sampling is an important aspect in surveys as it is closely linked to the generalizability. Overall, it can be based on a probability or non-probability sampling [100]. In probability sampling the likelihood of selecting each respondent is known, which enables statistical inference about the population from the responses of the sample. There are various probability sampling techniques such as simple random sampling (subjects are selected from a list of the population at random) and stratified random sampling (the population is divided into a number of groups or strata with a known distribution from which random sampling is applied). In contrast to a probability sample, the probability of selecting subjects is unknown in non-probability sampling. Consequently, it is difficult to know the accuracy of a sample estimate owing to the lack of proper statistical grounds. Although non-probability sampling depends on subjective judgments and restricts the possibility to generalize results to a larger population, it can be appropriate in certain circumstances. For example, when the surveyor selects a sample because it is composed of especially interesting cases or is convenient.

In surveys both flexible and fixed design can be used [100]. A flexible design allows an iterative research process where details of the study design emerge during data collection and analysis. In a fixed design the initial study design is followed throughout the research process. Data are collected in a survey instruments through, for example, self-administrated questionnaires, interviews, and structured observations.

1.3.3.2 Case study

Case studies place the researcher directly in the context of the phenomena of interest, such that contextual factors are fully accounted for and added into the content of the research [100, 101]. The overall objective of case studies is to develop an in-depth understanding about a single case or a number of related cases. Typical features are: (1) selection of a single case or a small number of related cases of a situation, individual or group of interest, (2) the case is studied in its context (e.g., real software organizations and projects), and (3) both qualitative and quantitative data can be used, and collected via a range of methods such as interviews, questionnaires and document analysis [100].

The tactics in case studies for sampling cases are based on criteria other than representativeness, since random or stratified collection from an identified population is not feasible in case research [102]. Single or multiple cases can be selected. Choosing a single case can be motivated by the fact that a specific case captures typical features or is revelatory enabling investigations of phenomena that has been previously inaccessible. The logic underlying a multiple case study design is similar to that guiding multiple experiments and that each case should be selected based on either: (1) literal replication, or (2) theoretical replication [101]. While the first approach to selecting cases predicts similar results across them, the latter yields contrary results, but for predictable reasons.

The sample size in case studies is often small, making the results from case studies difficult to generalize. To strengthen the validity of the results, case studies typically combine data collection methods such as archives, interviews, questionnaires, and observations.

Case studies usually adopt a flexible design, implying that the research process is iterative where details of the study design emerge during data collection and analysis [100]. However, good planning by defining the case(s) to be studied and data collection strategy is crucial for success.

1.3.3.3 Action Research

Action research is one of several streams of collaborative research approaches that are rooted in the action research school which suggest a strategy for collaborative studying and simultaneously changing social systems [103]. Action research involves some kind of intervention (e.g., process improvement) that is realized in a real world setting and the effects are then observed and evaluated. The research process is often viewed as cyclic including four main steps [100]: (1) planning of the intervention, (2) intervention (action taking), (3) observing the effects of the intervention, and (4) reflection (lessons learned) and planning of further actions (repeating the cycle).

In action research the focus is on the change process where the collaboration between the researchers and practitioners and their degree of participation in the process are central [100]. Instead of primarily playing the roles of observers as in case studies [104], the researchers take an active part in the team affected by the intervention introduced [100]. Furthermore, in some action research efforts the concept of insider/outsider research is used [105]. While the insider perspective offers detailed know-how about typical practices in the organization, as well as enables access to the company and key persons, the outsider provide a complementary critical distance to the empirical material leveraging an unbiased reflection in action [106].

Likewise case studies, a wide range of data collection methods and both qualitative and quantitative data can be used in action research. Due to the collaborative nature of action research, researchers and practitioners may share the responsibility for gathering data, and making the analysis

1.3.3.4 Systematic Mapping Study

Systematic mapping is a secondary study method. It provides a structure of the type of research reports and results that have been published by classifying them and often gives a visual summary, the map, of its results. A systematic mapping study often requires less effort than a systematic literature review while providing a more coarse-grained overview [97, 98]. Previously, systematic mapping studies in the field of SE have been recommended mostly for research areas where there is a lack of relevant, high-quality primary studies [97].

1.3.4 Data Collection Methods Utilized

In this thesis, both quantitative and qualitative data have been collected as a combination of these two data types can enhance the understanding of the phenomena studied and strengthen the validity of the results [100, 101]. Quantitative data are measurable values on a nominal, ordinal, interval, or ratio scale representing physical or ranking values, which are often collected as measurements (e.g., defect rates), but also in questionnaires (opinions of people). To ensure the collection of accurate quantitative data and control of variables, rigorous specifications of the study design and piloting of instruments (e.g., questionnaires) prior to data collection are required (i.e. fixed design) [100]. In qualitative data, the empirical material is represented as words and pictures, and not numbers. Usually, data are collected through interviews, archival data, and observations. The following data collection methods have been used in this thesis.

1.3.4.1 Interviews

Interviews are an important data collection method in case studies where the researcher is guided by an interview protocol [101]. The interview protocol embodies interview questions that cover the area of interest and have the capability of answering the research questions. The degree of structure of the interview questions can vary from closed via open-ended to open questions depending on the purpose of the study (e.g. explorative or explanatory). Open-ended questions support a semi-structured interview style which is advantageous in situations where the researcher aims at exploring an area of interest without drifting away from relevant topics during the interview [100]. Furthermore, semi-structured interviews are flexible, allowing the researcher to follow up answers and interpret expressions and the tone of the interviewee. On the other hand, interviews often generate an overwhelming amount of data to be analyzed (e.g., transcribed and coded), which is time consuming [107].

1.3.4.2 Self-Completion Questionnaires

Self-completion questionnaires are answered by the subjects themselves and can include closed and open questions. Using self-completion questionnaires implies a risk of that data quality is negatively affected by deviating interpretations owing to unclear and ambiguous wording, and that respondents may not necessarily report their beliefs and attitudes accurately [100]. To avoid different interpretation of the questions between respondents, closed questions are preferable to open-ended questions [100]. When measuring and evaluating subjects' opinions on something closed questions and different scales (e.g., Likert) representing levels of the subjects' opinions that are converted into numbers are widely used. Self-completion questionnaires can be administrated through, for example, email or online self-completion questionnaires.

1.3.4.3 Workshops

Workshops can be seen as group discussion where a number of subjects (e.g., representing different organizations, groups roles in a company), openly discuss a theme(s) (e.g., solutions to resolve problems or feedback on research results). During the workshop, the subjects are usually guided by a moderator, who also plays an important role in identifying differences of opinions and exploring why the subjects hold the view they do. The moderator should prevent single subjects or groups from dominating the workshop, encourage everyone to be actively involved and speak freely, and steer the workshops, but without bias the results. Workshops allow collecting data from several different people at one occasion. On the other hand, it can be difficult and resource consuming to assure that required people are available the same time.

1.3.4.4 Archival data

Archival data are a collection of written documents, such as books, project information, and documentation, describing processes and instructions, but also non-written documents such as films and drawings. Archival data can be both qualitative (e.g., written documents) and quantitative (e.g., project records). In contrast to data collection such as in interviews and observations, which are usually directly linked to the purpose of an inquiry, archival data are produced for other purposes and is an unobtrusive measure (e.g., the number of times books are borrowed from a library as an index of their popularity) [100].

1.3.5 Data Analysis Methods Utilized

In this thesis, descriptive statistics, correlation analysis and inferential statistics (hypothesis testing) have been used for analyzing quantitative data. The analysis of qualitative data was mainly influenced by the principles of grounded theory [108], and has been followed as practically as possible.

1.3.5.1 Qualitative Data Analysis

The principles of grounded theory emphasize the importance of systematically scrutinizing empirical data with an aim to develop theories that are grounded in data collected. However, complementary advice was needed as it is not an easy task to carry out an analysis based only on the prescriptions for a genuine grounded theory. Yin [101] stresses the importance of having a general analytic strategy, which is the best preparation for conducting a case study as it facilitates the analysis. A general analytic strategy defines the priorities for what to analyze and why.

Miles and Huberman [109] suggest that the data analysis consist of three concurrent flows of activity: (1) data reduction (extracting and coding data), (2) data display (categorizing and showing data in matrices), and (3) conclusion drawing/verification (e.g., triangulating data sources). Triangulation is an important tool for confirming the validity and generalizability of the conclusions in empirical research. It can be used both in research based on qualitative and quantitative data and applied in various ways and combinations. Triangulation involves cross comparison between different kinds of data collection methods (e.g. interviews, observations and archival data) and different units/cases (e.g. external/internal organizations and projects). In addition, triangulation between involved researchers/observers and theories with deviating approaches to the empirical material can be utilized.

When analyzing multiple cases, Eisenhard [102] suggests that the data analysis should follow two major steps: (1) within-case analysis and (2) searching for cross-case patterns. The overall idea of this approach is to become familiar with each case before the researchers endeavor to obtain generalization across the cases.

1.3.5.2 Quantitative Data Analysis

The purpose of descriptive statistics is to aid data analysis by graphically presenting data and identifying abnormal data [96]. For this, different types of diagrams (e.g., histograms, pie-charts, and box-plots) showing, such as mean values, distributions, standard deviations and correlations, are used. In order to get the feeling of how data is distributed, descriptive statistics may be used prior to hypothesis testing.

Inferential statistical methods are mathematical procedures for statistical hypothesis testing, dealing with inferences about populations based on the behavior of samples. That is, a null hypothesis is rejected depending on how likely it is that results based on a sample or samples are the same results that would have been obtained for the entire population. The statistical methods can be classified in parametric tests and non-parametric tests [96].

Parametric tests (e.g., t-test and ANOVA) are usually more powerful and generally to be preferred. However, to be valid, parametric tests make three assumptions of the data being analyzed: normal distribution of data, interval or ratio data, and randomization of sampling. If these assumptions cannot be met, non-parametric tests (e.g., Mann-Whitneys and Kruskall-Wallis test) can be used.

1.3.6 Utilized Research Method Overview

The challenges and research questions in this thesis have been addressed by using a mix of the above described methods. A summary of the research methods, data collection and analysis methods used in each chapter is shown in Table 3. Steps in the research process, research questions, and challenges are also linked to each chapter in Table 3.

In Chapter 2, RQ1, C1, and C2 are addressed. The main purpose of this study was to identify and gain a better understanding of the challenges between PD and MAN in development of large-scale software-intensive systems. Besides, the study demonstrates how process improvement initiatives addressing inter-departmental interaction can be conducted. A multiple case study was conducted where qualitative data were collected, using semi-structured interviews and archival data. The analysis of data was based on qualitative methods and consisted of two main steps. First, each case was analyzed independently by two researchers and their results were discussed and consolidated. Second, the results were compared and triangulated across the cases.

The studies presented in Chapters 3, 4 and 5 use case study and survey as research methods in order to address RQ1, RQ2, C1, C2, and C3. The primarily aims of these studies were to define the most critical problems and their causes, and evaluate the process improvement methods used, regarding their usefulness and applicability in industry. Both quantitative and qualitative data were collected through interviews, workshops, self-completion questionnaires and archival data. Descriptive statistics were mainly used for analyzing data, but also the significance of differences between groups of subjects was tested through inferential statistics (Chapter 3). In addition, in-depth analyses of qualitative data were conducted in order to identify root causes in Chapter 4, and enrich the postmortem analysis in Chapter 5.

Chapter 6 addresses RQ3, C4 and C5, and presents a literature review of the state of the art in large-scale software-intensive systems development, building on lean principles and practices. The main purpose was to clarify to what extent lean has been used in such development. A systematic mapping study was conducted, which is a variant of a survey. Relevant studies were located by systematically searching in scientific databases and selecting primary studies based on a number of exclusion/inclusion criteria. The primary studies were quantitatively classified into a number of predefined representative properties and analyzed through descriptive statistics. For a deeper analysis, data was enriched by extracting qualitative data (e.g., descriptions of the study context and the contributions).

In Chapter 7, RQ4, C4 and C5 are addressed, including a presentation of BRASS and a detailed description of how BRASS can be tailored in practice and a validation by applying BRASS to three real cases in industry. Action research was conducted as one of the researchers was actively involved during the study. For validating BRASS, quantitative data were collected using self-completion questionnaires and semi-structured interviews were used for collecting qualitative data, enriching the analysis. To analyze data, descriptive statistics and qualitative methods were used.

Table 3 Overview of links between chapters, research process steps, research questions, challenges and research methods.

Chapter	Step	Research question	Challenge	Research method	Data collection method	Data analysis method
2	1	RQ1	C1, C2	Case study	Interviews and archival data	Qualitative data analysis
3	2	RQ1, RQ2	C1, C2	Case study and survey	Self-completion questionnaires and workshops	Quantitative data analysis
4	2	RQ1, RQ2	C1, C2, C3	Case study	Self-completion questionnaires and workshops	Quantitative and qualitative data analysis
5	2	RQ1, RQ2	C2, C3	Case study	Interviews and archival data	Quantitative and qualitative data analysis
6	2	RQ3	C4, C5	Systematic mapping study	Archival data	Quantitative and qualitative data analysis
7	3, 5, 6	RQ4	C4, C5	Action research	Interviews and self-completion questionnaires	Quantitative and qualitative data analysis

1.3.7 Validity Evaluation

The quality of the results of empirical research is commonly discussed in terms of validity and reliability. Validity is concerned with the findings capability to describe the reality with a good fit [100]. Reliability concerns the possibility to repeat the study and arrive at the same conclusions [101] and is also considered in conclusion validity [96]. Based on the description in [96, 100, 101], the quality of the research presented in this thesis can be summarized according to four validity perspectives.

Construct validity. This concerns establishing correct operational measures for the concepts being studied, for example, the study instrumentation should measure what is intended. This threat was limited by performing, external reviews and piloting of interview guides, self-completion questionnaires, and data extraction forms. The threats to construct validity can also be expressed as respondent biases and researcher biases. In Chapters 2-5, and 7 these threats were primarily mitigated by utilizing three main strategies as described in [100]: 1) prolonged involvement, 2) triangulation and 3) peer debriefing. To guard against built-in bias of the selection of studies in Chapter 6, the consistency was evaluated by calculating the Fleiss' Kappa value [110].

Internal validity. This concerns establishing causal relationships, whereby certain conditions are shown to lead to other conditions, as distinguished from spurious relationships. When answering the questions posed in interviews and questionnaires, the subjects may feel unwilling to express their real opinions. To limit this threat, the subjects were guaranteed anonymity and that sensitive information would neither be published nor possible to trace to individuals. Deviating interpretations of questions owing to unclear and ambiguous wording may also affect accuracy of the subjects' answers. This was mitigated by adding complementary explanations text and clarifying instructions through examples.

External validity. This concerns establishing the domain to which a study's findings can be generalized. In this thesis, only two companies and their industrial settings have been studied in Chapters 2-4, and Chapters 5 and 7 include one of these companies. Furthermore, the selection of companies and subjects is based on non-probability quota sampling limiting the possibility to generalize the results. To enhance the possibilities for the readers to judge whether the results can be used in other settings, detailed descriptions of the contexts studied are provided. Furthermore, the overall organizational structure and the development processes used at the case companies are representative for automotive companies. This should strengthen the possibility to generalize the results to industrial sectors with similar characteristics, or at least to the automotive domain.

Conclusion validity. To enhance the possibility of repeating the research conducted in this thesis, the research processes used, such as sampling, data collection and analysis, have been thoroughly described, and detailed records of relevant data have been established (e.g., interview tapes, transcripts, and archival data). However, achieving repeatability is difficult as the industrial setting (e.g., development processes and organizational structure) at the case companies studied continuously change over time. The relatively few cases and small sample sizes used yield a low power of the statistical test from which conclusions can be drawn. To alleviate this threat, the number of cases and the sample sizes have been expanded as much as possible in relation to available resources. Furthermore, statistical analysis techniques meeting the required assumption of the data sets analyzed have been used.

1.4 Results

In this section, Section 1.4.1 provides a brief overview of each chapter, including main results and contributions. In Section 1.4.2 an author contribution statement is given and Section 1.4.3 draws together the obtained results in order to provide answers to the research questions asked in this thesis.

1.4.1 Chapter Summary

1.4.1.1 Chapter 2

Chapter 2 reports on an empirical study designed for process assessment (PA) of the inter-departmental interaction between PD and MAN in development large-scale software-intensive automotive systems at VCC and VTC. The main objectives of Chapter 2 were to discover a broad set of challenges (improvement issues) in the area assessed and provide an extensive analysis of the issues by viewing each of them against the contributing state of the art. The results of Chapter 2 set the baseline for answering RQ1.

The methodology used for the PA was primarily based on iFLAP (improvement Framework utilizing Light weight Assessment and improvement Planning) mainly as both companies had limitations in allocating resources for the SPI initiative and iFLAP provides a cost effective alternative to larger assessment frameworks. IFLAP was further tailored (e.g., selection of roles and subjects, and development of data collection and analysis methods) to fit the industrial setting assessed and meet the needs of VCC and VTC. To capture the knowledge and experiences of practitioners, 20 professionals representing different roles in PD and MAN at VCC and VTC were interviewed. In addition, documentation about development processes and organization were studied.

The main results are reported in terms of nine improvement issues that were identified using iFLAP. The cross-case analysis showed that eight of these issues were supported by both VCC and VTC indicating that they had similar challenges. The issues covered a broad range of aspects from challenges in requirements engineering (RE) to the need for knowledge transfer between MAN and PD. The improvement issues were further classified into three categories: people, process, and tools and technologies. A complete list and detailed descriptions of the issues are provided in Chapter 2.

The main results of the analysis of the improvement issues in relation to the state of the art showed that previous studies provide little practical guidance for resolving the problems identified in the area investigated.

The main contributions of Chapter 2 to the challenges presented in Section 1.2.2 are:

- Presents a detailed description of how PA can be performed in the context of large-scale software-intensive systems development focusing on inter-departmental and multidisciplinary interaction where the organizations assessed have limited resources for SPI effort.
- The resources needed for assessing the two companies (including company staff and assessors) was about 400 person-hours.

This primarily addresses:

C1—SPI initiation threshold (attaining effective and efficient use of the resources required to conduct SPI initiatives is challenging*)*.
C2—Tailorability of SPI methods (aligning SPI initiatives with company's strategies, goals, size and setting is a major challenge which requires tailoring of the SPI methods).

1.4.1.2 Chapter 3

Following the results of the PA, Chapter 3 presents the results of the subsequent improvement planning (IP) step. Chapter 3 had two main objectives. First, to establish a realistic plan of the development and implementation of improvements determined from priorities and dependencies between improvement issues. This prioritization also takes risk, cost of implementation, and TTROI into account. Second, to evaluate the industrial applicability of the method used for improvement planning in the context of inter-departmental interaction in large-scale software development.

In Chapter 3, a tailored IP method was developed and used, building on prioritizing, dependency mapping and packaging of improvement issues as used in iFLAP. Data were collected in both a web-based survey and a workshop. Cumulative voting and Analytical Hierarchical Process were used for prioritizing the improvement issues and their dependences were assigned in a table. In total 41 professionals at VCC and VTC assigned priorities and dependencies to the improvements issues. For evaluating the method itself, feedback was collected from participating professionals in a questionnaire and a subsequent workshop.

The results of the packaging showed that three issues were deemed most important to deal with first: (1) RE, (2) Early manufacturing involvement, and (3) Roles and responsibilities. Of these, the issue of RE was central due to dependences identified. An overall result related to the IP method evaluation showed that it is useful and has the capability of identifying the most critical issues and sorting them into feasible packages. However, even though the issues had been successfully packaged into a delimited number of high-priority issues, they were on a high-level and needed to be broken down into clearly defined problems and root causes.

The main contribution of Chapter 3 to the challenges presented in Section 1.2.2 are:

- Presents a detailed description of how IP can be performed in the context of large-scale software-intensive systems development focusing on inter-departmental and multidisciplinary interaction where the organizations assessed have limited resources for SPI.
- Using the tailored IP method required 196 person-hours in total (including company staff and assessors).

This primarily addresses:

C1—SPI initiation threshold (attaining effective and efficient use of the resources required to conduct SPI initiatives is challenging*)*.
C2—Tailorability of SPI methods (aligning SPI initiatives with company's strategies, goals, size and setting is a major challenge which requires tailoring of the SPI methods).

1.4.1.3 Chapter 4

The high–priority improvement issues identified in Chapter 3 detailed the main challenges on a high level. They could to some extent be considered as the symptoms of an undesired behavior rather than the actual root causes. However, both traditional and lightweight frameworks give little further guidance on how to decompose the issues into targeted and defined sub-problems, and uncover their causes. The main purposes of Chapter 4 is to introduce and evaluate a root cause analysis (RCA) method for a systematic causal analysis of identified improvements issues and to report the results of applying the method to the results of the packaging in Chapter 3.

For the RCA, a method called Flex-RCA was developed. Flex-RCA, has four main steps: (1) problem and goal definition, (2) selecting problems for root cause generation, (3) root cause generation, and (4) analysis. Flex-RCA builds primarily on the lean Six Sigma approach and uses data based on organizational knowledge and experience. The data were collected in a workshop series with 13 professionals representing different roles in PD and MAN at VCC and VTC. An evaluation of the method itself (Flex-RCA) was also performed by collecting feedback on the method through questionnaires and in review meetings.

The results showed that the most prevalent root causes revolved around inadequate methods for communicating and handshaking requirements between PD and MAN, but also a need for clarifying the breakdown process of requirements for deeper analysis and detailing of requirements specifications.

Overall, the lessons learned from using Flex-RCA showed that it had the desired effect of both producing a broad base of causes on a high level and underlying root causes with adequate depth. Furthermore, the evaluation of Flex-RCA indicated that it was effective and useful.

The main contribution of Chapter 4 to the challenges presented in Section 1.2.2 are:

- Gives a detailed description and exemplification of Flex-RCA, clarifying the practical use of the method and helping practitioners in different industries to tailor Flex-RCA for systematically carrying out root cause identification as a stand-alone activity or in combination with different SPI frameworks.
- Flex-RCA gives detailed guidance on identification and definition of goals and targeted problems pertaining to the process area being analyzed, and identification of underlying root causes.
- Using Flex-RCA required 209 person-hours in total (including company staff and assessors).

This primarily addresses:

C1—SPI initiation threshold (attaining effective and efficient use of the resources required to conduct SPI initiatives is challenging*)*.
C2—Tailorability of SPI methods (aligning SPI initiatives with company's strategies, goals, size and setting is a major challenge which requires tailoring of the SPI methods).
C3—Goals and measurement (a critical challenge is to clearly define and formulate problems and expected goals, and to measure the effects of the SPI initiative in order to determine whether it was successful)

1.4.1.4 Chapter 5

Following the results of Chapter 4, the communication between PD and MAN was further analyzed in Chapter 5. The main objectives of Chapter 5 are to present a model covering central communication aspects that can be used for approximating organizational communication problems, to exemplify how to tailor and apply the model, and to give feedback on the model's usefulness. Furthermore, the results of the modeling and analysis of the communication problems between PD and MAN are reported.

In Chapter 5, a novel model called software communication redundancy effectiveness model (SCREAM) for in-depth postmortem analysis of deficient organizational communication is presented. It consists of four main elements representing central aspects in communication: *sender*, *receiver*, *communication*, and *specification*. The model focus on communication of problem-solution artifacts (e.g., requirements) associated to development projects). Each element in SCREAM has two attributes used to describe different communication problems in an organization. Different patterns of communication can be modeled by chaining the elements and the coding of their attributes. For collecting additional information, a set of properties can be attached to the model and enrich the analysis and understanding. The overall goal of SCREAM is to be conceptually simple and practically useful and tailorable to suit different industry contexts. SCREAM also characterizes organizational communication problems in a structured and descriptive way in order to reveal effects and causes on which efforts for developing improvements can be motivated and based.

SCREAM evolved in close cooperation with VCC. As part of the evolution, SCREAM was applied to 16 real communication events, representing different types of inter-departmental communication failures between PD and MAN at VCC.

The results of applying SCREAM to the 16 events, suggested that lack of shared understanding of the matter being communicated is prevalent. Furthermore, in many of the analyzed events a more detailed specification and more and/or better communication in order to collectively agree on that specification, would be needed. A deeper analysis of the results also showed a grand total estimated improvement potential of 11 224.6KUS$ (pertaining to the 16 events). An example of a specific problem identified through the use of SCREAM at VCC, having a major impact on the costs, concerned inadequate and too imprecise communication and balancing of requirements and solutions over the full car development cycle.

The results of using and applying SCREAM showed that it provides a structured and systematic way for collecting, modeling and classifying data, and analyzing dysfunctional organizational communication patterns. SCREAM also allowed deeper analysis of individual communication aspects, which helped in revealing underlying causes and effects of communication failures.

The main contribution of Chapter 5 to the challenges presented in Section 1.2.2 are:

- Presents SCREAM, which is a model that can be tailored for modeling and analyzing different organizational communication problems in different industries and organizations.
- Using SCREAM, made it possible to structure the communication problems and to clarify what is most critical to improve.
- A set of properties for collecting data that can be used for measuring the effects of deficient organizational communication are provided and exemplified.

This primarily addresses:

C2—Tailorability of SPI methods (aligning SPI initiatives with company's strategies, goals, size and setting is a major challenge which requires tailoring of the SPI methods).

C3—Goals and measurement (a critical challenge is to clearly define and formulate problems and expected goals, and to measure the effects of the SPI initiative in order to determine whether it was successful)

1.4.1.5 Chapter 6

Lean approaches have had a strong influence on many industries, in particular the automotive, and there have been many proponents for lean in software development. The main purpose of Chapter 6 is to identify and classify state of the art in large-scale software development influenced LPD principle and practices, and use this established knowledge to support the creation of solutions to the core challenges addressed in this thesis and the specific problems identified at VCC.

For locating relevant state of the art a systematic mapping study was conducted. To provide researchers with an overview of the status of the area and any research gaps, the selected studies were classified into a number of representative facets (e.g., research type, topic in SE and coverage of LPD principles) and visually summarized. To assist practitioners when seeking to adopt new “best” lean practices, and give researchers information about the quality of the studies reported, the degree of relevance and rigor for each study was also assessed and gauged.

The findings of the systematic mapping study showed that there are very few studies reporting on state of the art utilizing LPD principles and practices and clearly addressing large-scale software development. Furthermore, the methodological quality and strength of evidence were low as a majority of these studies lacked information about study design, context, and empirical validation. Most of the identified results focused on eliminating waste and creating flow in the software development process, but in general there was a lack of results for other LPD principles and practices.

The main contribution of Chapter 6 to the challenges presented in Section 1.2.2 are:

- Reports on an extensive review of available state of the art based on LPD practices and principles applied to large-scale software development.

This primarily addresses:

C4—Coordinating and communicating enlarged requirements and solution space (the substantial growth of software increases the interdependences between requirements and development tasks and their uncertainty, leading to major challenges in coordinating and communicating a much larger requirements and solution space within and across departments).
C5—Achieving effective requirements and solution specifications (for achieving improved coordination communication within and across departments, a critical challenge is to efficiently manage specifications of requirements and their solutions while it is vital to ensure the quality of them).

1.4.1.6 Chapter 7

Chapter 7 presents the BRASS (Balancing Requirements and Solution Space) framework as a solution to the core challenges, and the specific needs at VCC identified in Chapters 2-5. The main purpose of Chapter 7 is to thoroughly describe and exemplify how to tailor BRASS in practice and to test its applicability through initial validation in industry.

Inspired by the model of goal-oriented requirements communication and the handshaking technique developed and applied by Fricker et al. [89], and the lean practice Set Based Concurrent Engineering (SBCE) [4], BRASS was developed in close collaboration with VCC. BRASS is a lightweight RE framework and its overall goal is to improve requirements coordination and communication and to be generic enough to be tailorable to a wide variety of organizational needs. BRASS consists of four generic dimensions, *communication*, *content*, *connections*, and *actors*, which must be adapted in order to fit an organization's needs and prerequisites. To validate and demonstrate how to use BRASS, it was tailored and applied in three real cases (balancing issues) in an ongoing new car development project at VCC. In addition, the framework itself was evaluated by collecting feedback from ten professional on its practical usefulness and potential using a questionnaire and follow-up interviews.

Overall, the results of the validation showed that BRASS can be tailored and applied in industry, and the feedback indicated that BRASS is useful and an efficient way of communicating and balancing requirements and solutions across PD and MAN. However, there are also concerns, such as effectively identifying the needs of communicating and balancing requirements, and solutions, requiring necessary resources and competences in early phases and attaining acceptance and adoption of BRASS throughout the company.

Overall, however, the results were promising. This has led to that VCC has decided to perform further validations of BRASS in a large-scale pilot. This in order to identify what works and what needs to be changed, giving decision-support for refining BRASS before it can be implemented full-scale.

The main contribution of Chapter 7 to the challenges presented in Section 1.2.2 are:

- Presents and validates the BRASS framework designed to improve requirements coordination and communication and be tailorable to satisfy the needs of different industries and companies.

C4—Coordinating and communicating enlarged requirements and solution space (the substantial growth of software increases the interdependences between requirements and development tasks and their uncertainty, leading to major challenges in coordinating and communicating a much larger requirements and solution space within and across departments).
C5—Achieving effective requirements and solution specifications (for achieving improved coordination communication within and across departments, a critical

challenge is to efficiently manage specifications of requirements and their solutions while it is vital to ensure the quality of them).

1.4.2 Contribution Statement

Joakim Pernstål is the main author of each chapter (i.e. paper) in this thesis. With the assistance and guidance of Professor Tony Gorschek and Professor Robert Feldt. Joakim Pernstål was responsible for the planning and design of the studies presented in Chapters 2, 3, 4 and 7 assisted by the coauthors. For the studies presented in Chapters 5 and 6, Professor Tony Gorschek and Professor Robert Feldt were responsible for the planning and design of the studies with high participation of Joakim Pernstål. Joakim Pernstål was responsible for the data collection and analysis in all the studies presented in this thesis with assistance from the coauthors. The author's contribution to this thesis is summarized in Table 4.

Table 4 The author's contribution to this thesis

Activity	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7
Planning and design	Responsible	Responsible	Responsible	High participation	High participation	Responsible
Data collection	Responsible	Responsible	Responsible	Responsible	Responsible	Responsible
Data analysis	Responsible	Responsible	Responsible	Responsible	Responsible	Responsible
Writing	Responsible	Responsible	Responsible	Responsible	Responsible	Responsible

Responsible—driver of the main part of the activity.
High participation—actively involved and driver of parts of the activity.

1.4.3 Thesis Research Questions Revisited

RQ1: What areas are most important software engineering challenges in the interface between PD and MAN in the automotive domain?

To answer RQ1 four empirical studies were conducted and their results are reported in Chapters 2-5. Chapter 2 reports the results of the PA in terms of nine improvement issues. The results of the packaging of the issues in Chapter 3 show that RE, early manufacturing involvement, and roles and responsibilities were deemed as most critical issues. In Chapter 4, the results of applying RCA to the packaged improvement issues show that insufficient communication and handshaking of requirements between PD and MAN is a predominant root cause. The postmortem analysis and modeling of communication failures between PD and MAN resulted in that much of the problems can be traced to lack of bidirectional communication and insufficient quality of specifications, leading to lack of shared understanding of the matters being communicated, and in particular requirements (see Chapter 5).

The answer to RQ1 is that effective requirements communication and coordination, and sufficient quality of specifications are the core challenges.

RQ2: How can the challenges identified through a process improvement initiative be prioritized and analyzed to reflect the core needs of an organization?

RQ2 is addressed in Chapters 2-5 where different methods for assessing and planning improvements are presented and evaluated in industry. In Chapters 2 and 3, the tailored SPI methods based on iFLAP show that it has the capability of assessing and identifying improvement issues, and packaging the ones that are most critical and must be dealt with first. The results of the evaluation indicate that the methods were useful, giving valuable decision support for the planning of the improvement. Using and evaluating Flex-RCA show that it was effective and useful, and had the desired effect of both producing a broad base of causes on a high level and underlying root causes with adequate depth (see Chapter 4). In Chapter 5, the results of using and applying SCREAM show that the model helped structure and conduct systematic data collection and analysis of dysfunctional communication patterns.

The answer to RQ2 is that lightweight inductive PA and IP can produce packages of improvement establishing an implementation order based on organizational core needs. However, the packages are often on a high level indicating symptoms, and thus there is a need of supplementing guidance to identify delimited and targeted problems and find their causes and effects. For this, Flex-RCA and SCREAM are potential solutions.

RQ3: Can current lean practices alleviate the core challenges (RQ1) identified?

To answer RQ3, a systematic mapping study was conducted in Chapter 6. The findings show that there are very few high-quality studies reporting on state of the art utilizing LPD principles and practices being applied in large-scale software development in industry. State of the art in research (non-empirically validated work) is also scarce.

The answer to RQ3 is that the current state of the art, offering specific advice to industry professionals pursuing improvements in large-scale software development, by applying lean principles and practices, is scarce. In particularly, when it comes to the core challenges identified in RQ1.

RQ4: How can the core challenges identified (RQ1) be efficiently and effectively resolved?

To answer RQ4, Chapter 7 presents and validates the BRASS framework for improving requirements coordination and communication over the full development cycle. Overall, the results suggest that BRASS can be tailored and applied in industry, and feedback from practitioners indicates that BRASS is useful and an efficient way of communicating and balancing requirements and solutions across PD and MAN. The promising results have led to that VCC has decided to perform further validations of BRASS in a large-scale pilot.

RQ4 can be partly answered through the initial validation of BRASS at VCC, indicating that it can be tailored and be useful in further improving requirements communication between PD and MAN. To fully answer RQ4, further investigations of using BRASS in larger scale and in other industrial settings are needed to validate its tailorability and scalability.

1.5 Discussion

This section discusses the answers to each research question and future work.

1.5.1 Research Questions

RQ1: What areas are most important software engineering challenges in the interface between PD and MAN in the automotive domain?
Attaining good RE, and in particularly, effective and efficient coordination, communication, and specification of requirements, and transformation of them into balanced solutions across the departments of PD and MAN was found core challenges (C4 and C5) in the studies presented in Chapters 2-5. Looking at literature and previous studies reporting on related state of the art, this finding is partly supported. For example, [1, 9, 11] point out RE as a major challenge in the automotive domain due to the advancement of automotive software, and the importance of requirements coordination and communication across organizational boundaries has been especially emphasized in large-scale software-intensive systems development [4, 5, 10]. Furthermore, in general product development, the need of a well-functioning interaction between PD and MAN has been stressed to prevent misfits between product design and manufacturing processes [3, 7, 8, 17, 69, 71, 72]. However, related state of the art viewing the challenges from both the software development and the PD and MAN interface perspectives are very limited. The core challenges (C4 and C5) answering RQ1 can therefore provide a starting point for improvement work in industry and as well as new research opportunities in the research setting focused on in this thesis.

Central to any communication is to achieve a shared understanding of the matter of concern [112]. However, deeper analysis of the communication between PD and MAN in Chapter 5 showed that there is commonly a lack of shared understanding of matters being communicated mainly as the exchanged information, specifically requirements and solutions, are specified insufficiently. One reason found for this is that particularly manufacturing requirements is often experienced-based and tacit rather than being captured in detailed specifications of purposes and goals, which is in line with [14]. In addition, it is well-known that much information is tacit and never written down in software development [4]. Another reason found is that constraining manufacturing requirements are generic and on the complete vehicle level, and not sufficiently broken down on lower design levels, making it difficult for developers of a single system or function to convert the constraints to measurable and understandable parameters. Difficulties in specifying and communicating precise and understandable requirements on an appropriate level of abstraction are also well-known problems in software development [11, 113]. Only improving the quality of the requirements specifications would probably have a minor effect, since perfect requirements specifications are impossible to achieve, especially when it comes to large complex software-intensive systems [11, 78]. It will also increase specification costs and unnecessarily constraining the solution space [9, 89]. Furthermore, involved people and roles have a tendency to assume that upfront produced artifacts convey all the information needed for downstream development work, inhibiting continuous exchange of information through-out the product development cycle, and in particular, across organizational units [4].

Closely related to achieving a shared understanding is sufficient knowledge development, which was also found as one of the key improvement issues identified and was given the third highest priority (see Chapters 2 and 3). One strategy for building knowledge in an organization is systematization and storage of explicit (codified) knowledge gained from concluded projects, and making it accessible and easy to use by anyone at the company. In lean automotive companies, this is commonly accomplished by establishing know-how databases evolved from checklists [8][65], and similarly this is often referred to as having an Experience Factory (EF) [114] in software development. Examples of benefits are, improved quality of produced development artifacts, more effective risk management throughout the development process, and reduced risk of propagating the same mistakes across projects [115]. However, this strategy is often resource consuming [114] and the need of communication is probably not reduced by increasing the documentation [5]. Therefore, in order to build organizational knowledge other strategies, such as creation of learning networks that encourage and facilitate informal transfer of tacit knowledge throughout the company must also be adopted. This is central in LPD [8, 65] and LSD [83], but also agile software development methods, such as Scrum[84], highly rely on the organization's capability of mediating tacit knowledge [88].

The above discussion suggests that seeking solutions to the core challenges identified promoting communication of good-enough specifications [89] over focusing on achieving perfection in requirement specifications seems beneficial. However, the analysis in Chapter 5 also showed that in some cases neither of PD or MAN had understood the matter being communicated. Thus, even if measures are taken that directly can improve the communication itself (e.g., adding communication mediums and processes), the shared understanding will not increase, as requisite knowledge is not available within the two departments. This means that the communication problems are most likely stemming from other factors than insufficient communication and sharing of knowledge. Differences between PD and MAN in terms of personality, cultural and organizational barriers (see Section 1.2.4.1), and lack of required education and training of staff and inadequate information systems [116] can be examples of such factors.

RQ2: How can the challenges identified through a process improvement initiative be prioritized and analyzed to reflect the core needs of an organization?

Using an inductive lightweight framework (iFLAP see Section 1.2.3.2) showed that it could be tailored for PA and IP in the context of inter-departmental interactions in large-scale development of software-intensive systems. However, in order to identify the core challenges of an organization in this context, there is a need to extend such lightweight frameworks beyond the packaging. For this Flex-RCA and SCREAM were developed and used in this thesis (see Chapters 4 and 5). The overall results of applying Flex-RCA and SCREAM showed that they have the capability of producing adequate and relevant outputs, and are simple and generic enough to be tailorable and scalable for industrial use while still providing detailed guidance on how to be used in practice. However, creating models and methods that are easy to comprehend, scalable and adaptable to specific industrial needs and problems without omitting critical aspects and sacrificing accuracy is challenging [117, 118]. For example, a limitation of SCREAM is that it can only be used to model and assess communication events that involve two actors. However, including more actors in SCREAM will dramatically increase the complexity. Consequently, it was decided to include this aspect of communication in the set of properties for collecting additional information. To further investigate the communication events between other actors, instantiations of SCREAM can be used.

During the process improvement initiative, the methods used and the results obtained were evaluated in questionnaires and review meetings by professionals at VCC and VTC and researchers. The total resource usage (including company staff and assessors) including for PA, IP and RCA using iFLAP and Flex-RCA was 805 person–hours (see Section 1.4.1). This can, for example, be compared to CMMI appraisal costs [58, 59]. Overall, the results of the evaluations showed that there is a good confidence in the methods used and an agreement on the resulting core challenges and specific problems identified. This indicates that the methods used are effective, but it is more difficult to say whether they are efficient. This is partly due to difficulties in collecting trustworthy data on the opinions of effort used (i.e. how can the participants judge this when the research team cannot) and partly because comparisons to earlier improvement is hard as the accuracy of the evaluation results can be questioned and detailed descriptions of process changes and contextual settings are often lacking [52]. Nevertheless, showing and discussing the required process improvement effort in this thesis give researchers and practitioners an indication that can be helpful when estimating required effort and judging the efficiency of their method.

Usually, a positive side-effect for cross-functional improvement works, like the one performed in this thesis, is that the barriers between the departments are slowly but steadily torn down while the empathy and understanding of each other's work are increased [119]. This could also be seen during the research presented in this thesis. The representation of different functions and roles provided an opportunity to build networks between PD and MAN, enabling sharing of knowledge between them. Furthermore, involving different companies provided an opportunity to benchmark against other companies (VCC and VTC). However, the cross-functional work can only initiate the building of the networks but do not have the capability of nurturing them and to assure that the obtained empathy and understanding of each other's work among the participants are disseminated throughout their organizations. Thus, the long-term effect is highly dependent on the participants' ability and willingness to maintain their relationships and share acquired knowledge.

RQ3 Can current lean practices alleviate the core challenges (RQ1) identified?
The quality assessment of the studies selected in the systematic mapping study showed that there were a relative few relevant and high-quality studies. However, this assessment gives only an overall picture in the field being reviewed by approximating the potential industrial relevance and its progress, rather than providing precise and detailed criteria for an exact classification of each individual study [117]. There are other factors influencing the evaluation of the feasibility than those based on the actual use in industry, making it difficult to distinguish feasible state of the art from unfeasible ones. For example, the likelihood of introducing subjective bias when assessing the value of research is high in both what to assess (e.g., what constitutes value and quality), and reviewers' competence to assess it [120]. The time perspective needs also to be considered as it takes in the order of 15-20 years before state of the art has matured so it can be implemented in industry [121]. Thus, even though most of the studies were deemed as being low-quality studies and could not be used for solving the core challenges and specific problems identified in the research presented in this thesis, this does not mean that the state of the art reported are unfeasible for other industries and organizations.

A majority of the studies reported on the use of lean in general terms without specifying the particular practice adopted. It could be observed that value stream mapping and visual management are the only specific types of lean practices reported in the studies addressing large-scale software development. Furthermore, these studies have been published recently (2006-2010), which is much in line with Wang et al. [73], who saw a trend of adopting more and more concrete lean practices. Furthermore, they found that mature agile software organizations tend to shift from agile methods to lean approaches. For example, to move away from time-boxed agile processes (e.g., Scrum sprints) to more flow–based lean processes, such as Kanban software development. Moreover, the unique focus on the whole in lean supports the expanding industry need of scaling agile software development [122, 123]. However, there is currently no ground for such a movement in traditional industries developing large-scale software-intensive systems, such as the automotive one. This, because these industries have only started to adopt agile methods and they are not yet widespread [88]. From a systems engineering perspective with a primarily focus on hardware development, on the other hand, a trend in these industries is to adopt LPD principles and practices for increasing the efficiency in their product development processes. A main driver for this is to utilize the full benefits of their implementations of lean manufacturing (LM) practices [8].

Given these trends, one implication for future research is that there is a need for more rigorous studies on the topic of how the advancement of LPD in systems engineering and the observed movement from agile to lean in software development can be combined and tailored to suit specific situations and organizational needs. This includes a need to explicitly map what must be added to LPD and how its base principles must be refined and extended in order to support lean-oriented industrial sectors where the share of software in the products is rapidly growing.

RQ4 How can the core challenges identified (RQ1) be efficiently and effectively resolved?

The results of using and validating BRASS indicate that it is useful and an effective way of communicating and balancing requirements and solutions across organizational boundaries, and thus can address C4. However, the findings also suggest a challenging part of BRASS is to effectively identify the needs of balancing requirements and solutions between PD and MAN. To tackle this, two extreme approaches were discussed. The first one implies that all requirements in a car project are reviewed and any identified dependences between MAN and PD requirements indicate a balancing need. However, a car project contain about 100 000 requirements along with other development artifacts (e.g., specifications, models and standards), which makes this approach enormously resource consuming and not feasible. Another problem is that the specifications of the software-intensive systems are often incomplete in early phases, which is in line with [11, 78]. The other extreme approach merely relies on experiences from concluded project—i.e. the balancing needs are based on identified and documented needs and experiences of earlier projects (see below). The effort is much lesser, but a big disadvantage is the obvious risk of omitting new balancing needs. Thus, the results suggest to compromise between these two extremes by, for example, relying on experience along with singling out new features in the projects and reviewing related development artifacts.

The validation of BRASS indicated that it has the capability of improving the quality of specifications (C5) with regard to precision and support for decision making. The findings show that the BRASS documentation (see Chapter 7) helps to get an overview of issues that needs to be balanced and to give information about the status of each issue. Furthermore, gathering and structuring requirements and solutions in the BRASS documents can clarify both the boundaries for the solutions and the consequences of realizing the requirements. The results also suggest that the BRASS documents can supplement requirements specifications in the requirements management systems by providing additional information that enriches analysis and understanding of requirements. Furthermore, the documentation can clarify decisions taken during a project, and enable learning from concluded projects and transferring and refining solutions across projects.

As much manufacturing knowledge is tacit [14] and information and knowledge exchanged in software development are often not written down [4], the BRASS documents can be used for surfacing and capturing such knowledge in order to build know-how and competence. Thus, BRASS may have the potential of also address the high-prioritized improvement issue of knowledge development identified in Chapters 2 and 3, which has also been argued in earlier work to be a critical success factor in automotive and software development [8, 65, 114]. In addition, for effective use and administrations of the BRASS documents, they should be stored and archived in one structured repository. Traceability and consistency between the documents produced by BRASS and related formal documentation was also found important, but it was preferred to not integrate the documentation with the formal information systems (e.g., requirements management systems) in order to keep the effort low for handling the documentation. VCC has therefore established a shared website where the BRASS documentation is stored with trace-links to documents in the formal document management systems.

1.5.2 Future Work

Based on the answers to the research questions, four main directions of future work are suggested in this thesis.

First, the answer to RQ1 is only based on the results of investigating two automotive companies. Even though the results (C4 and C5) are partly supported in earlier work there is a need of further studies on large-scale software-intensive systems development with a focus on inter-departmental interaction. Similar investigations in other companies and industrial settings would most likely result in other core challenges and specific organizational needs than those addressed in this thesis.

The second direction concerns RQ2. The results show that it was necessary to extend lightweight frameworks for process improvement with two methods, namely Flex-RCA and SCREAM enabling deeper analysis beyond the packaging. To enhance the generalizability and exploit the applicability of these methods further studies on using them in industry is encouraged. Furthermore, in future work, the required effort for conducting process assessment and improvement in Chapters 2-4 can be compared to other methods in order to evaluate efficiency.

The third direction is related to RQ3. The results of the systematic mapping study showed that rigorous research on lean principles and practices as used in industries developing large-scale software-intensive systems is very limited. Moreover, there seems to be two intersecting lean trends as discussed above. Traditional hardware-focused industrial sectors where software is becoming a substantial components have started to implement parts of LPD while mature agile software development organization tend to shift from agile methods to lean approaches. Consequently, there is need of more rigorous research on lean based applications to large-scale software development. Furthermore, it should be investigated which combinations of lean and agile practices are best suited for such development, taking the overarching adoption of LPD into account.

Finally, to provide a more complete answer to RQ4 more research is needed. A concrete research opportunity is to study the further validation of BRASS in the large-scale pilot at VCC. This, in order to obtain a better understanding of how all parts of BRASS can be tailored, and to evaluate its feasibility and scalability over a full product development cycle. In particular, the applicability of the validation step of balanced and agreed requirements and solutions, as described in Chapter 7, must be further investigated. Furthermore, improvements and extensions of BRASS are needed including pertinent techniques for identifying balancing needs, and tools and metrics for measuring the effects of BRASS. In addition, to strengthen the generalizability of

BRASS, there is also a need of extended research on its tailorability and applicability in different industries and settings.

1.6 Conclusions

The research presented in this thesis is primarily based on industrial needs and challenges. The purpose was to help practitioners improve their practices, and thus solve industrial problems. This thesis focuses on two main parts. First, methods and models for process assessment and improvement were developed, but also detailed analysis of the identified challenges as a precursor to devising candidate solutions. Second, focus was put on developing a framework to address the specific challenges associated with the inter-departmental interaction between the departments of PD and MAN in large-scale development of software-intensive automotive systems.

An important contribution of this thesis is the methods developed and used for process assessment and improvement—which were lightweight, tailorable and used triangulation for increased accuracy. However, it could be concluded that even though lightweight methods go a step further than most process improvement methods there is need of extending them with supplemented guidance on how to break down identified problems areas.

For this reason, two novel approaches were developed. Flex-RCA, was developed to identify more targeted problems and their root causes, and SCREAM was developed for postmortem analysis and modeling of organizational communication problems. Overall both Flex-RCA and SCREAM produced useful results. However, they are highly dependent on the experiences and knowledge inherent the organization, and thus it is crucial to identify and ensure accessibility to key personnel when doing assessments. The key persons should also represent different functions, groups, and roles that are concerned. Furthermore, developing too precise and fine-grained methods including too many details sacrifices scalability and thus industrial applicability.

As a result of the assessment it could be observed that with the high complexity of automotive development follows many challenges. The importance of good requirements engineering, and a need of effective interaction between a multitude of different engineering disciplines and departments was considered central. A critical challenge is also to effectively manage the large amount of artifacts and information. The lack of bidirectional communication combined with insufficient specifications (of e.g., requirements) presented significant problems at the partner company. Most of the communication is performed in late development phases and much of the exchanged information is tacit. This was aggravated by unclear roles and responsibilities and handover points which resulted in communication gaps.

Any possible solution to the challenges had to have the characteristics of being able to stimulate and enable communication in the early phases of development, capture tacit knowledge, and be trailorable to the companies' needs. For this reason, this thesis presents a lightweight RE framework called Balancing Requirements and Solution Space (BRASS). BRASS combines goal oriented requirements communication with lean based concurrent engineering, and promotes communication over achieving perfection in requirement specification. Based on initial validation, BRASS can be tailored and applied in industry and the practitioners perceived the use of BRASS as useful and an efficient way of communicating and balancing requirements and solutions. The main reason for this is the adaptable and simple nature of BRASS since organizational needs change over time, and it is well known that methods being too rigid are less efficient in development work than methods that can be adapted to the context of development instance in question.

More research is needed to understand, characterize, and create scalable, effective and efficient solutions to the many problems present in large-scale software-intensive product development. The potential solutions presented in this thesis might be a good starting point for some areas—but the work continues, as process assessment and improvement is a continuous endeavor.

References

[1] M. Broy, I.H. Kruger, A. Pretschner, C. Salzmann; “Engineering Automotive Software”, *Proceedings of the IEEE*, 95(2), pp. 356–373, 2007.

[2] K. Venkatesh Prasad, M. Broy, I. Krueger, “Scanning Advances in Aerospace & Automobile Software Technology,” *Proceedings of the IEEE*, 98: 510-514, 2010.

[3] J. Nihtilä,” R&D–Production integration in the early phases of new product development projects,” *Journal of Engineering and Technology Management*, 16: 55-81, 1999.

[4] R.E. Kraut, and L.A. Streeter, “Coordination in software development,” *Communications of the ACM*, 38(3):69-81, 1995.

[5] B. Curtis, H. Krasner, and N. Iscoe, “A field study of the software de-sign process for large systems,” *Communications of the ACM*, 31(11), 1988.

[6] D.K. Sobek, A.C. Ward, and J.K. Liker, “Toyota's Principles of Set-Based Concurrent Engineering”, *Sloan Management Review*, 40(2), 1999.

[7] S.C. Wheelwright and K.B. Clark, “Accelerating the design-build-test cycle for effective product development,” *International Marketing Review*, 11:32-46, 1994.

[8] J.M. Morgan and J.K. Liker, *The Toyota Product Development System: Integrating People, Process, and Technology*, New York: Productivity Press, 2006.

[9] T. Gülke, B. Rumpe, M. Jansén, J. Axmann, “High-level requirements management and Complexity costs in Automotive Development Projects: A Problem Statement,” in *Requirements Engineering: Foundation for Software Quality Engineering and Managing Soft-ware Requirements*, B. Regnell, D. Damian (eds), Springer, 2012

[10] S. Konrad and M. Gall, “Requirements engineering in the development of large-scale systems,” in *Proceedings of the 16th IEEE International Requirements Engineering Conference*, pp. 217–222, 2008.

[11] M. Weber and J. Weisbrod, “Requirements engineering in automotive development: Experiences and challenges,” *IEEE Softw.*, 20:16-24, 2003.

[12] C. Karlsson, P. Åhlström, "Assessing changes towards lean production, *International Journal of Operations & Production Management*," 16: 24-41, 1996.

[13] J.P. Womack, D.T. Jones and D. Roos, The machine that changed the world. New York: Rawson Associates, 1990.

[14] L. Almefelt, F. Berglund, P. Nilsson and J. Malmqvist, "Requirement management in practice: findings from an empirical study in the automotive industry," *Research in engineering design*, 17(3), 2006.

[15] V. Sumantran, "Accelerating product development in the automobile industry. *Int. Journal of Manufacturing Technology and Management*," 6(3/4), 2004.

[16] A. Peters, E. Rooney, J. Rogerson, R. McQuater, M. Spring, and B. Dale, "New product design and development: A generic model," *The TQM Magazine* 11(3):172-179, 1999.

[17] A. Vandevelde and R. Van Dierdonk, "Managing the design-manufacturing interface*, Int. J. of Operations & Production Management*, 23(11):1326-1348, 2003.

[18] Volvo Car Corporation, (2011), https://www.media.volvocars.com/global/enhanced/en-gb/Media/Preview.aspx?mediaid=41678. Accessed Feb 2013.

[19] Volvo Truck Corporation, Quick facts about Volvo Trucks, 2010. http://www.volvotrucks.com/trucks/global/en-gb/newsmedia/quick_facts_about_volvo_trucks/pages/quick_facts_about_volvo_trucks.aspx . Accessed Feb. 2013

[20] ABG, "V-Model: Development Standard for IT-Systems of the Federal Republic of Germany, Lifecycle Process Model," available at http://www.v-modell.iabg.de, 1997.

[21] F. Charfi and F. Sellami, "Overview on Dependable Embedded Systems in Modern Automotive," *in Proc. of the 2004 IEEE Int. Conference on Industrial Technology*, 2004.

[22] J.P. Kuilboer, and N, Ashrafi, "Software Process and Product Improvement: An Empirical Assessment", *Information and Software Technology*, 42: 27-34, 2000.

[23] M.E. Fayad and M. Laitnen, "Process assessment considered wasteful", *Commun. ACM*, pp. 125-128, 1997.

[24] K.D. El Emam, D.J. Goldenson, J. McCurley, and J. Herbsleb, "Modeling the Likelihood of Software Process Improvement: An Exploratory Study", *Empirical Software Engineering*, 6: 207-229, 2001.

[25] V.R. Basili, F.E.McGarry, R. Pajerski, M.V. Zelkowitz, "Lessons Learned from 25 Years of Process Improvement: The Rise and Fall of the NASA Software Engineering Laboratory," in *Proceedings of the 24th International Conference on Software Engineering (ICSE02)*, ACM, Orlando, pp. 69-79, 2002.

[26] Y. Mashiko and V.R. Basili, "Using the GQM Paradigm to Investigate Influental Factors for Software Process Improvement," *The Journal of Software and Systems*, 36: 17-32, 1997.

[27] D.H. Kitson and S.M Masters, "An Analysis of SEI Software Process Assessment Results: 1987–1991", in *Proceedings of the 15th International, Conference on Software Engineering*, IEEE, Los Alamitos CA, pp. 68 – 77, 1993.

[28] CMMI. Capability Maturity Model Integration Version 1.3. Technical Report CMU/SEI-2010-TR-033. Software Engineering Institute (SEI), 2010. http://www.sei.cmu.edu/cmmi/. Accessed Feb.2013.

[29] ISO/IEC Std. 15504: Information Technology - Process Assessment,Part 1 to Part 5, International Organization for Standardization, 1998–2005.

[30] M.C. Paulk, B. Curtis, M.B. Chrissis, C.V. Weber, *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley, Reading MA, 1994.

[31] S. Zahran, *Software Process Improvement: Practical Guidelines for Business Success*, Reading, Addison-Wes, 1998.

[32] J.G. Brodman and D.L. Johnson, "What Small Businesses and Small Organizations Say About the CMM", in *International Conference on Software Engineering (ICSE-16)*, Sorrento, Italy, pp. 331-340, 1994.

[33] F. Pino, F. Garcia, M. Piattini, "Software process improvement in small and medium software enterprises: a systematic review", *Software Quality Journal* 16 (2): 237–261, 2008.

[34] J.A. Calvo-Manzano, G. Cuevas, T. San Feliu, A. De Amescua M. Pérez, "Experiences in the application of software process improvement in SMES", *Software Quality Journal*, 10 (3), 261–273, 2002.

[35] C.Gv. Wangenheim, S. Weber, J.C. Rossa Hauck, G Trentin,. (2006) " Experiences on establishing software processes in small companies". *Information and Software Technology*; 48 (9), 890–900, 2006.

[36] K. Kautz, H.W. Hansen, K. Thaysen, "Applying and Adjusting a Software Process Improvement Model in Practice: The Use of the IDEAL Model in a Small Software Enterprise", in *Proceedings of the 2000 International Conference on Software Engineering*, IEEE, Los Alamitos CA, pp. 626-633, 2000.

[37] L. Scott, R. Jeffery, L. Carvalho, J. D'Ambra, P. Rutherford, "Practical Software Process Improvement - the Impact Project", in *Proceedings of the Australian Software Engineering Conference*, IEEE, Los Alamitos CA, pp. 182-189, 2001.

[38] V.R. Basili, *Quantitative Evaluation of Software Methodology*, University of Maryland. College Park, MD, 1985.

[39] F. Pettersson, M. Ivarsson, T. Gorschek, P. Öhman, "A practitioner's guide to light weight software process assessment and improvement planning", *The Journal of Systems and Software* 81, 972-995, 2008.

[40] T. Gorschek, C. Wohlin, "Identification of Improvement Issues Using a Lightweight Triangulation Approach," In *The European Software Process Improvement Conference*, Graz. Austria, 2003

[41] T. Gorschek, C. Wohlin, "Packaging Software Process Improvement Issues - A Method and a Case Study", *Software: Practice & Experience*, 34, 1311-1344, 2004.

[42] T. Gorschek, P. Garre, S. Larsson, S. C. Wohlin, C. "A Model for Technology Transfer in Practice", *IEEE Softw.*, 23: 88-95, 2006.

[43] J. Sivie, and E. Forrester, "Enabling Technology Transition Using Six Sigma", in J. Bergey et al. (eds), *Results of SEI Independent Research and Development Projects and Report on Emerging Technologies and Technology Trends* (Chapter 5) (CMU/SEI-2004-TR-018). Software Engineering Institute, Carnegie Mellon University, Pittsburgh PA, 2004. http://www.sei.cmu.edu/library/abstracts/reports/04tr018.cfm. Accessed Feb. 2013.

[44] K. Beck, "Extreme Programming Explained: Embrace Chage", 2nd ed, Addison-Wesley, 2004.

[45] M.C. Paulk, "Agile METHODOLOGIES AND Process Discipline", *Crosstalk: The Journal of Defense Software Engineering*, 15:15-18, 2002.

[46] D.M Ahern, A. Clouse, R. Turner, *CMMI Distilled: A Practical Introduction to Integrated Process Improvement*, Addison-Wesley, Boston MA, 2003.

[47] A. Rainer, T. Hall, "A Quantitative and Qualitative Analysis of Factors Affecting Software Processes", *The Journal of Systems and Software*, 66(1):7-21, 2003.

[48] B. Kitchenham and S.L. Pfleeger, "Software Quality: the Elusive Target," *IEEE Softw.*, 13, p 12, 1996.

[49] S. Rifkin, "What Makes Measuring Software so Hard?," *IEEE Softw.*, 18, p. 41, 2001.

[50] S. Rifkin, "Is Process Improvement Irrelevant to Produce New Era Software?," *in Proceedings of the 7th International Conference on Software Quality*, pp. 13-16, 2002.

[51] B. Kitchenham, L. Pickard, and S. Pfleeger, "Case studies for method and tool evaluation," Software, IEEE, 12(4): 52–62, 1995.

[52] M. Unterkalmsteiner, T. Gorschek, M. Islam, C. Cheng, R. Permadi and R. Feldt, "Evaluation and Measurement of Software Process Improvement—A Systematic Literature Review," *IEEE Transactions on Software Engineering*, 38(2), 2012.

[53] K.B. Clark, T. Fujimoto, *Product Development Performance: Strategy, Organization and Management in the World Auto Industry*, Harvard Business School, Boston, 1991.

[54] J.D., Thompson, *Organizations in action*, McGraw-Hill, New York, US, 1967.

[55] J.R. Galbraith, "Designing Complex Organizations," Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1973.

[56] W.S. Humphrey, *Managing the Software Process*, Addison Wesley, Reading, Massachusetts, 1989.

[57] D. Jacobs, *Acceleraring process improvement using agile techniques*, Auerbach Publications, 2006.

[58] SCAMPI-ADT, "Standard CMMI Appraisal Method for Process Improvement (SCAMPI) Version1.1 (CMU/SEI-2001-Hb-001)", Carnegie Mellon SEI. Pittsburgh, PA, 2001.

[59] CMMI-PDT, "Appraisal Requirements for CMMISM, Version 1.1 (ARC, V1.1)," 2001.

[60] ISO/IEC 15288: 2008, "Systems engineering—system life cycle processes".

[61] ISO/IEC 12207:1997, "Software life cycle process."

[62] Automotive SIG, "The SPICE User Group, Automotive SPICE Process Assessment Model v2.5 and Process Reference Model v4.5," 2010. http://automotivespice.net/web/download.html. Accessed Feb. 2013.

[63] K.E. Emam, L. Briand, "Cost and Benefits od Software Process Improvement". Fraunhofer—Institute for Expereimental Software Engineeering, ISERN-97-12, 1997.

[64] D.R Goldenson, K.E. Emam, J Herbsleb, C. Deephouse, "Empirical Studies of Software Process Assessment Methods". In *Elements of Software Process Assessment and Improvement*: IEEE CS Press, 1999.

[65] M.N. Kennedy, K. Harmon, E. Minnock, *Ready, Set, Dominate –implementing Toyota's set based Learning for Developing Products … and nobody can catch you*, Oaklea Press, Richmond, VA, 2008.

[66] H. Mintzberg, *The Structure of Organizations*, Upper Saddle River, NJ, Prentice Hall, 1979.

[67] M.E. Conway, "How do committees invent?," *Datamation*, 14(4):28-31, 1968.

[68] D.L. Parnas, "On the criteria to be used in decomposing systems into modules," *Communications of the ACM*, 15(12), 1972.

[69] M. Carlsson, Integration mellan konstruktion och produktion, Licentiatuppsats, (Institutionen för Industriell Organisation, Chalmers tekniska Högskola, Göteborg, 1988.

[70] L. Trygg, "Engineering Design—Some Aspects of Product Development Efficiency. PhD thesis, Chalmers University of Technology," Göteborg, Sweden, 1991.

[71] P. Adler, "Interdepartmental interdependence and coordination: the case of the design/manufacturing interface*," Organization Science*, 6 (2):147-167, 1995.

[72] N. Lakemond, G. Johansson, T. Magnusson, and K. Safsten, "Interfaces between technology development, product development and production: critical factors and a conceptual model," *International Journal of Technology Intelligence and Planning* 3(4):317-330., 2007.

[73] X. Wang, K. Conboy, O. Cawley, ""Leagile" software development: An experience report analysis of the application of lean approaches in agile software development." *Journal of Systems and Software* 85:1287– 1299, 2012.

[74] IEEE Standard: 830-1998, "Recommended practice for software requirements specifications."

[75] N. Maiden, "What has requirements research ever done for us," *IEEE Softw*., 22(4):104–105, 2005.

[76] I. Jacobson, M. Christerson, P.Jonsson, and G. Övergaard*, Object-Oriented Software Engineering – A use Case Driven Approach*, Addison Wesley, 1992

[77] B. Potter, J. Sinclair, and D. Till, *An Introduction to Formal Specification and Z*, Prentice Hall, 1996.

[78] I. Sommerville*, Software Engineering* (Eighth ed.), Addison-Wesley Publishers Ltd., 2007.

[79] Object Management Group (OMG), "OMG UML Version 2.3, Infrastructure," 2010. http://www.omg.org/spec/UML/2.3/Infrastructure/PDF/. Accessed Feb. 2013

[80] T. Gorschek and A.M. Davis, "Requirements Engineering: In Search for the Dependent Variables, *Information and Software Technology* 50: 67-75, 2008.

[81] AUTOSAR, "Automotive Open System Architecture." Available: http://www.autosar.org/. Accessed: Feb. 2013.

[82] ISO 14229-4: 2012, "Road vehicles – Diagnostics systems –Diagnostic services." http://www.iso.org/iso/catalogue_detail.htm?csnumber=55285. Accessed Feb. 2013

[83] M. Poppendieck and T. Poppendieck, "Lean software development: an agile toolkit, " Addison-Wesley, Boston, 2003.

[84] K. Schwaber, and M. Beedle, M, "Agile Software Development with Scrum,", Prentice Hall, Upper Saddle River, 2001.

[85] K. Petersen, C. Wohlin," Software process improvement through the Lean Measurement (SPILEAM) method," *Journal of Systems and Software* 83: 1275-1287, 2010.

[86] B. Ramesh, L. Cao, R. Baskerville,"Agile requirements engineering practices and challenges: an empirical study." *Information Systems Journal* 20(5): 449-280, 2010.

[87] J. Savolainen, J Kuusela, A Vilavaara "Transition to agile development – rediscovery of important requirements engineering practices," in: Proceedings of the Requirements Engineering Conference, IEEE, pp. 289–294, 2010.

[88] T. Dybå and T.Dingsøyr, T. "Empirical Studies of Agile Software Development: a Systematic Review, " Journal of Information and Software Technology, vol. 50, 833-859, 2008.

[89] S. Fricker, T. Gorschek, C. Byman, and A. Schmidle, "Handshaking with Implementation Proposals: Negotiating Requirements Under-standing, " *IEEE Softw*., vol. 27, no. 2, 72-80, 2010.

[90] R.L. Glass, "The Software-research-crisis," IEEE Softw., 11:42-47, 1994.

[91] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, "A model for Technology Transfer in Practice," *IEEE Softw*., 23: 88-95, 2006.

[92] C. Wohlin, A. Aurum, L. Angelis, L. Phillips, Y. Dittrich, T. Gorschek, H. Grahn, K. Henningsson, S. Kågström, G. Low, P. Rovegård, P. Tomaszewski, C. van Toorn C., and J. Winter, "The Success Factors Powering Industry-Academia Collaboration", *IEEE Softw*., vol. 29(2): 67-73, 2012.

[93] L. Mathiassen, "Collaborative Practice Research. *Information, Technology & People*, 15(4): 321-345, 2002.

[94] L. T. Blessing and A. Chakrabarti. *DRM, a Design Research Methodology*. London: Springer, 2009. http: //link.springer.com/book/10.1007/978-1-84882-587-1.

[95] M. Ivarsson, "Experince Driven Software Process Assessment and Improvement," PhD Thesis, ISBN 978-91-7385-398-9, Chalmers University of Technology, 2010.

[96] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlson, B. Regnell, and A.Wesslén*, Experimentation in Software Engineering: An Introduction*, Kluwer Academic, Boston, MA, 2000.

[97] B. Kitchenham, B, S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software Engineering," Technical Report EBSE 2007-01, School of Computer Science and Mathematics, Keele University, 2007.

[98] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic Mapping Studies in Software Engineering, 12th International Conference on Evaluation and Assessment in Software Engineering (2008), 71-80, 2008.

[99] A. Bryman and E. Bell, *Business Research Methods*, Oxford University Press: Oxford, 2003

[100] C. Robson, *Real World Research: A Resource for Social Scientists and Practitioners-Researchers* (2nd ed.), Blackwell, 2002.

[101] R. Yin, *Case study research: design and methods* (3rd ed.), Sage Publications Inc, Thousand Oak, 2003.

[102] K. Eisenhardt, "Building theories from case study research," *Academy of Management Review*, 14(4):532–550, 1989.

[103] A.B. Shani, A. David and C. Wilson, "Collaborative Research: Alternative Roadmaps", in N. Adler, A.B Shani, A. Styhre (eds.), *Collaborative Research in Organizations: Foundations for Learning, Change and Theoretical Development*, SAGE Publications: London, 2004.

[104] P. Runesson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, 14, 131-164, 2009.

[105] J.M. Bartunek and M.R. Louis, "Insider/outsider Team Research," *Qualitative Research Methods* 40, Sage Publications, 1996.

[106] D. Schön, *The Reflective Practitioner: How Professionals Think in Action*, New York, NY: Basic Books, 1983.

[107] S. Kvale, *Den Kvalitativa forskningsintervjun*, Studentlitteratur Lund, 1997.

[108] B.G. Glaser and A.L. Strauss, *The Discovery of Grounded Theory:Strategies for Qualitative Research*, Aldine Publishing, 1967.

[109] M.B. Miles, A.M. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook* , 2nd ed., Thousand Oaks, Calif.: Sage, 1994.

[110] J. Fleiss, Measuring nominal scale agreement among many raters, Psychological Bulletin 76 (1971 378–382.

[111] S. Fricker, T. Gorschek, and M. Glinz, "Goal-Oriented Requirements Communication in New Product Development," in: *Proc. IEEE Second Int Workshop on Software Product Management*, pp. 27-34, 2008.

[112] C.E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, 27: 379-423 & 623-656, 1948.

[113] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, "Industry Evaluation of the Requirements Abstraction Model", *Requirements Engineering Journal*, vol. 12, no. 3, 163-190, 2007.

[114] V. R. Basili, G. Caldiera, and H. D. Rombach, "The experience factory, in Encyclopedia of Software Engineering," ed. J. J. Marciniak, John Wiley & Sons, pp 469–476, 1994.

[115] J.M. Verner and W.M. Evanco, "In-house software development: what project management practices lead to success?," *IEEE Software*, 22, p 86, 2005.

[116] L. Argote, "Organizational Learning: Creating, Retaining and Transferring Knowledge," Kluwer Academic Publishers, 1999.

[117] M. Ivarsson, T. Gorschek, "A Method for Evaluating Rigor and Industrial Relevance of Technology Evaluations," *Empirical Software Engineering*, 16(3), 365-395, 2011

[118] M. Ivarsson, T. Gorschek, "Technology Transfer Decision Support in Requirements Engineering Research: A Systematic Review of REj," Requirements Engineering journal 14: 155-175, 2009.

[119] S.C. Wheelwright and K.B. Clark, *Revolutionizing Product Development - Quantum Leaps in Speed, Efficiency, and Quality*, New York,The Free Press, 1992.

[120] T. Dybå, T Dingsøyr, G.K. Hanssen, G.K, "Applying Systematic Reviews to Diverse Study Types: An Experience Report" First International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 225-234, 2007.

[121] T. Dyba, B.A. Kitchenham, M. Jorgensen, "Evidence-based software engineering for practitioners," Softw. IEEE 22:58-65, 2005.

[122] K. Petersen, Is Lean Agile and Agile Lean? A Comparison Between Two Software Development Paradigms, in Modern Software Engineering Concepts and Practices: Advanced Approaches; Ali Dogru andVeli Bicer (Eds.), IGI Global, 2010.

[123] J. Coplien and G. Bjornwig, Lean architecture for agile software development. West Sussex, UK, John Wiley & Sons Ltd. 2010.