

## **A STUDY INVESTIGATING CHALLENGES IN THE INTERFACE BETWEEN PRODUCT DEVELOPMENT AND MANUFACTURING IN THE DEVELOPMENT OF SOFTWARE-INTENSIVE AUTOMOTIVE SYSTEMS**

J. PERNSTÅL

*Volvo Car Corporation,  
SE-405 31 Göteborg, Sweden  
[jpernstal@volvocars.com](mailto:jpernstal@volvocars.com)*

A. MAGAZINIUS

*Department of Computer Science and Engineering,  
Chalmers University of Technology,  
SE-412 96 Göteborg, Sweden  
[ana.magazinius@chalmers.se](mailto:ana.magazinius@chalmers.se)*

T. GORSCHKE

*Blekinge Institute of Technology,  
Karlskrona, SE-372 25  
[tony.gorschek@bth.se](mailto:tony.gorschek@bth.se)*

Received (Day Month Year)

Revised (Day Month Year)

Accepted (Day Month Year)

The automotive industry is facing a tremendous growth in the engineering of software-intensive systems, giving rise to various challenges. To prevent problems related to the fit of new software technologies in vehicles and the manufacturing processes, a well functioning interaction between the functions for product development and manufacturing is crucial. This is complicated by the fact that the changeable nature of software development causes unprecedented needs for collaboration and coordination between these two functions. This paper reports on a process assessment that focuses on the interface between the functions for product development and manufacturing in the development and design of software-intensive automotive systems. The main purpose of the study was to identify the key issues for improvement in the area assessed. The assessment was performed at two Swedish automotive companies where data were collected from documents and in interviews with practitioners. Nine key improvement issues were established ranging from challenges in requirements engineering to the need for knowledge transfer between manufacturing and product development. In addition, to increase the understandability of the results and map possible avenues for solution and future research, the paper provides an extensive analysis of each improvement issue in relation to the state-of-the-art.

*Keywords:* Process Assessment; Integrated Product Development; Manufacturing Engineering; Automotive Software Development; Process Improvement; Software Intensive Product Development.

### **1. Introduction**

Trends show that the number of electrical functions in vehicles is increasing radically, as is the vehicle complexity [34]. Premium vehicles today contain not less than 70 Electronic Control Units (ECU) running on one gigabyte of software that enables and controls various features and electrical systems [56]. Further, the entry of software into the automotive domain has probably just started and the total value of electronic and electrical (E/E) systems in automobiles is expected to rise from the current 25% to 40% in 2015, where software will grow exponentially relative to E/E hardware [10][41].

As software is rapidly increasing in vehicles, automakers must manage it from the development of new software based technologies further down the chain all the way to the factory floor. However, automotive product development systems and manufacturing systems have traditionally been adapted for the design and manufacturing hardware components such as body and chassis parts. These parts are relatively stable, especially in the final stages of development. In contrast, software is characterized by its being highly responsive to change with short and frequent iterations through all development phases.

Meeting customers' demands for a broader range of products with various features and tighter scheduling of new product/model launches also forces automakers to manage an increased number of products and systems that must be developed in increasingly shorter time. Product development (PD) in the automotive industry in the old sequential fashion – the "over the wall approach" – no longer works and is being replaced by other PD methods such as concurrent engineering (CE), integrated product development (IPD) and lean product development (LPD) aiming at improving PD activities across functions/disciplines [41, 50]. In particular, integration between the functions of PD and manufacturing has been recognized in research as an enabler for developing new products faster than competitors, reducing production cost and increasing product quality. For example, Daetz [17] shows that 75% of the production costs are determined in the development of products.

The experience and documentation of the development of new vehicles at one of the case companies studied in this paper show that most of the faults related to vehicle software are discovered and corrected in late development phases. Today's cost for adapting the vehicle software to the manufacturing processes can be estimated to be between 500KEUR to 2MEUR depending on the degree of change in a new vehicle. Considering the rapid increase in vehicle software, this cost will rise to between 2 MEUR and 8 MEUR in 2015 [10][41]. In parallel with this, late changes ("fixes") of products and manufacturing processes impair manufacturing performance at the worst possible time, when the market demand is peaking after the launch of a new product.

Drawing upon an earlier empirical study by Pernstål et al. [52], this paper reports a process assessment that focused on the interface between PD and Manufacturing in the design and development of software-intensive automotive systems at two Swedish automotive companies: Volvo Car Corporation (VCC) and Volvo Truck Corporation (VTC). The main objective of the study was to identify key improvement issues in the targeted area and provide an extensive analysis of the results by viewing each of the issues against the contributing state-of-the-art. To capture the key improvement issues,

the assessment was primarily guided by the steps in iFLAP (improvement Framework utilizing Light weight Assessment and improvement Planning) as described in [54]. The results reported in this paper can be used by practitioners in the automotive domain to get an idea of the challenges that lay ahead and the characterization of the state-of-the-art will give a baseline in relation to the maturity of possible solutions. From a research perspective the results indicate the direction of future cross-functional research encompassing the emerging need for integration of PD and manufacturing in development of software-intensive automotive systems.

It is notable that the scope of this study is broad ranging over the boundaries of the software engineering (SE), systems engineering and manufacturing engineering (ME) disciplines, despite its special focus on challenges related to the PD/manufacturing interface in the development of software-intensive automotive systems. Thus, in this study, PD is primarily defined as the design and development of complete vehicles focusing on software-intensive automotive systems and manufacturing is defined as the processes of assembling vehicles and in particular processes affected by vehicle software. To distinguish between issues that are generic in PD and manufacturing and issues that are specifically related to the development of software-intensive products/systems, the remainder of the paper uses the terms software development or development of software-intensive systems/products when issues specific to software related issues are discussed. The paper is organized as follows. Section 2.1 reviews pertinent literature covering cross-functional interaction in product development in particular between the functions of PD and manufacturing. Sections 2.2-2.4 give an overview of automotive software and the challenges it brings in the manufacturing of vehicles, and Section 2.5 describes the case companies. The research methodology and research questions are given in Section 3. Section 4 presents the results and Section 5 analyzes and discusses them in relation to the state-of-the-art. Finally, Section 6 gives the conclusions.

## **2. Background and Related Work**

### **2.1. Related work**

This study focuses on the interfaces between PD and manufacturing in the development and design of software-intensive automotive systems. To define the study's theoretical boundaries, this section reviews literature addressing IPD and related research on Japanese lean approaches. Previous research investigating the PD/Manufacturing interface is reviewed in particular.

Managing activities across functional groups is one of the main characteristics of IPD, which is a significant feature in today's new product development projects, e.g., [33]. In particular, integration between the functions of PD and manufacturing has been recognized in research as an enabler of developing new products faster than competitors, reducing production costs and increasing product quality. However, Carlsson [14] concludes that although there is an agreement that integration and cooperation between

the function of PD and manufacturing are crucial in product development, there are no satisfactory solutions owing to a number of barriers such as information (poor knowledge about the operations of other functions) and late manufacturing involvement. Moreover, Carlsson [13] shows that there are large differences between the functions of PD and manufacturing for various key variables. For example, the degree of structure is high in manufacturing while it is low in PD, manufacturing has shorter time horizon than PD and in contrast to well defined manufacturing tasks the PD tasks are often abstract. Carlsson [13] thus suggests that the large difference between the functions motivates differentiation since it is believed their operations can be more easily optimized when they are treated separately. On the other hand this aggravates the ambition to simultaneously achieve integration.

There are few empirical studies focusing on the PD/manufacturing interface [68]. One example is Vandevelde and Van Dierdonk [68] who claim that formalization and empathy on the part of PD towards manufacturing 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, Lakemond et al. [44] emphasize the need for formalization and empathy and observed that product manufacturability, early manufacturing involvement, dedicated resources for manufacturing involvement, active involvement and continuous communication are critical factors in the PD/manufacturing interface.

Besides the literature mentioned above, there are several studies on Lean Product Development (LPD) at Japanese automakers, e.g., [75], that have had a strong influence on approaches that have been developed to reinforce integration efforts between PD and manufacturing in product development. Various models for LPD have been elaborated, such as the Lean Product Development System (LPDS) developed by Morgan and Liker [48] and the Learning First Product Development System (LFPD) conceptualized by Kennedy [42].

Several studies on Japanese practices can be explicitly associated with IPD. For instance, to accelerate product development, Wheelwright and Clark [74] emphasize the necessity of understanding how problem solving is carried out across disciplines and functions in the product development process. They claim that the degree of integration is dependent on communication between upstream and downstream functions, e.g., PD and manufacturing, and scheduling of development activities. This is illustrated by four modes: (1) serial mode, (2) early start in the dark, (3) early involvement and (4) integrating problem solving. Set-Based Concurrent Engineering (SBCE) [63] is another practice in LPD that elevates integration across functions.

Empirical research addressing the PD and manufacturing interface in product development is limited [68] and inquiries specifically focusing on the area of interest here have not been found. All the previously mentioned works provide critical factors and recommendations related to cross-functional work from the point of view of various aspects and perspectives, e.g., [44][74]. These studies do not consider specific contextual needs, however, and are probably not beneficial in every industrial setting owing to such

things as the business, culture, and discipline and company size. For instance, the philosophic view of lean production (LP) implies that it may be suitable only in Japanese industry owing to the historical and cultural heritage [76]. This line of reasoning is also highlighted by Gorschek et al. [29] who point out the importance of understanding and adopting specific organizational needs in the search for solutions to problems that have been identified. On the other hand, to enrich knowledge bound to a specific context, eliminating the possibility of reinventing the wheel and alleviating the risk of only focusing on experiences inherent to the organizations that can result in the use of obsolete practices [19], it is necessary to have sufficient insight into the state-of-the-art.

## **2.2. Automotive software**

The amount of software in vehicles has increased dramatically in a short time. However, automotive software deviates from many other types of software (e.g., PC and telecommunication) as it has explicit demands on such aspects as reliability, real-time behavior and resource consumption [12][15].

Today most innovations in the automotive domain are driven by electronics, where software is rapidly becoming a crucial and dominant component. For example, Grimm [34] estimates that more than 80% of all future innovations in the automotive domain will be in electronic components and 90% thereof will be software.

Legislation and the customer demand for new features such as reduced gas consumption and increased performance and safety (e.g., hybrids and by-wire controls) pressure automakers to adopt new technologies that are realized by an increasing amount of software. Moreover, software-enabled features make possible product differentiation and mass customization which are important competitive capabilities among automakers. For instance, a current premium car implements about 270 features with which a user interacts, e.g., central locking. These are in turn composed of ~2500 interdependent software functions, e.g., locking, unlocking and remote. To accommodate these features, premium vehicles now contain not less than 70 ECUs that run on up to one gigabyte of software and communicate over various bus systems such as controller area network (CAN), media-oriented systems transports (MOST), FlexRay and local interconnect network (LIN) [12][56]. The costs of vehicles get more influenced by development costs of vehicle software. McKinsey [41] expects the total value of electronics in automobiles to rise from the current 25% to 40% in 2015 and Berger [10] predicts an exponential growth of software relative electronic and electrical (E/E) hardware over the next five years (300% vs. 50 %). Assuming a linear relationship between value of software and size would mean going from today's one gigabyte of code (hundreds of millions of lines of code (LOC)) [56] to four or more gigabytes in 2015 (billions of LOC).

As most future automotive innovations will be realized with software, this will affect the vehicle development processes that have traditionally been adapted for the development of mechanical parts. Mechanical parts are relatively stable over time, and the characteristics of the development processes are commonly rigorous and plan-driven. In

contrast, vehicle software can be changed at much shorter intervals during the development phase and also during production [56]. Although partnerships such as Automotive Open System Architecture [4] aim to facilitate reuse by standardizing E/E systems, it is estimated that currently 90 % of the software must be changed from one generation of vehicles to the next, while only 10% of other systems, e.g., exterior (doors) and interior (seats), are changed [12].

### **2.3. Automotive software development**

Product development in the automotive industry is characterized by a high complexity and engineering level involving many project members. The development activities are often associated with a high degree of interdependency where project tasks are performed in parallel, sequentially or in an overlapping way in different functions or disciplines in the project [40][48]. An automotive development system commonly consists of an overall business development process that evaluates new business proposals and updates the product cycle plan leading to the start of pre-program activities, e.g., advanced engineering (AE) and new vehicle programs. The automotive development process generally contains gates or checkpoints, where project management reviews and confirms that the gate criteria are met for the current gate, demonstrates preparation for the next gate and updates the project prediction for final delivery and associated risks. Certain decisions need to be made at each gate by gathering the necessary information between the gates.

In the development of software based automotive systems, automakers act mainly as Original Equipment Manufacturers (OEM) and the activities cohere more to systems engineering, than to the development of completely new software [12]. In general, the development activities are interdisciplinary and guided by the V-model [1], where functional requirements on an abstract level, so called vehicle attributes, are decomposed into functions, architectures, systems, and hardware and software solutions and are placed on the left half of the letter V [15]. The right half comprises testing and integration of components, systems and functions. In practice, however, the V-model and its iterative approach generally serve more as a methodological design guideline since the deliveries are governed by the overall gate system for developing the complete vehicle. Thus the characteristic of development of automotive software-intensive systems more resembles the approach of the Waterfall model [60]. Primarily based on the PD system at one of the case companies, Figure 1 maps the phases in the overall development of complete vehicles and their gates and the underlying stages in the development of software-intensive systems to a generic product development model developed by Peters et al. [53]. The following provides a brief description of these stages.

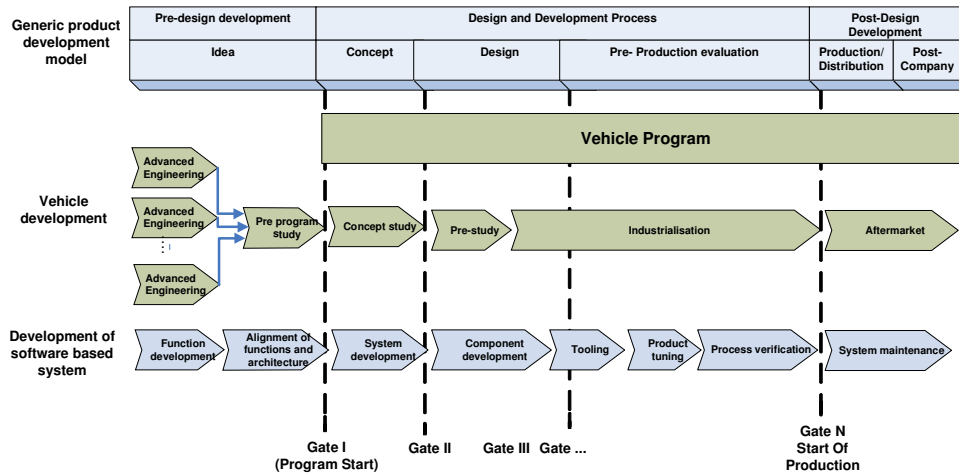


Fig. 1. Mapping of a generic model for product development (adapted from Peters et al. [53]), overall vehicle development and development of software based automotive systems.

In the pre-program development phase, business opportunities founded on customer demands and market surveys are incentives for new innovations and ideas (vehicle attributes) generating AE activities. In the development of software based functions, the AE activities typically consist of assessing the vehicle attributes and developing functions that accommodate the required attributes. The primary task in the subsequent pre-program study involves elaboration of an appropriate communication infrastructure and allocation of functions. The goal is to align functions that have been developed with underlying architectural constraints based on the complete vehicle design.

System development constitutes the first stage in vehicle program development. It comprises mapping and packaging hardware components such as sensors, actuators and ECUs that comply with vehicle design constraints. Further, the required data and message flow is defined by specifying and configuring signals, and prerequisites for designing software and hardware components are compiled and finalized in specifications such as software requirement specifications and design prerequisites for hardware. The next stage, component development, focuses on developing, integrating and verifying the hardware and software solutions based on the specifications elaborated in the prior system development stage. The primary task in the tooling stage is to achieve the required manufacturing capabilities at suppliers (e.g., quality and productivity) of hardware and software solutions. The subsequent product tuning stage aims to customize systems and the complete vehicle by, for example, fine adjustment and calibration of parameters and error diagnosis (diagnostics). Finally, the process verification stage

implies trial production of physically built vehicles in pilot plants and assembly plants. The objectives are to assure the manufacturing processes and logistic systems required to produce the product and that the production equipment is capable of maintaining the specifications required of the product. Manufacturing processes affected by software in vehicles are, for example, vehicle configurations (e.g., assembly plant software download) and vehicle verification (e.g., electrical tests).

#### **2.4. Manufacturing challenges in automotive software development**

To achieve a quality assured and cost efficient launch without schedule over-runs of new vehicle programs in the manufacturing process when production starts, it is essential to discover, analyze and deal with any product and manufacturing problems as early as possible [63][74]. Accordingly, the goal is that most of the software related issues in the development of vehicles such as faulty configurations, deficient functionality and any other problems related to configuration and verification of the vehicles in the manufacturing processes should be found and taken care of before pre-production evaluation starts or at least at the beginning of this phase. However, at the two case companies most software related issues start to be discovered in the pre-production evaluation phase peaking somewhere in the middle of this stage. Consequently, this causes unplanned design loop backs of the software in the final part of the project, and these are costly and jeopardize the launch of the vehicle.

The experience and project documentation at one of the case companies show that the average number of software releases for developing a new vehicle platform program (e.g., small or large) and a new model year program based on a mature platform are about 2000 and 500, respectively. Further, as a consequence of late discovery of software faults, most of the releases are carried out in late development phases, i.e. in design and pre-production evaluation, augmenting the costs for reengineering. The cost for each release in late project phases is roughly 5kEUR to 20kEUR.

Suppose that 10% of all software problems discovered are related to manufacturing problems and assume that the average cost of a software release is 10kEUR, which is probably higher owing to the late discovery of software faults, the cost of a new year model program for each platform would be about 500 KEUR ( $0.1 \cdot 500 \cdot 10\text{kEUR}$ ) to adapt the software to the manufacturing processes. Likewise, the cost of a new vehicle platform program would be 2 MEUR ( $0.1 \cdot 2000 \cdot 10\text{kEUR}$ ). With the assumption of a linear relationship between software size and the number of releases, the graphs in Figure 2 show how the exponential growth of automotive software influences the overall tendencies for rises in costs for adapting vehicle software and manufacturing parameters over time. For example, the graphs show that today's annual cost for new year models will rise from 500 KEUR to about 2 MEUR and the cost for a new platform will rise from 2 MEUR to about 8 MEUR in 2015. In addition to these external costs, there are underlying internal costs. To solve problems that arise, internal resources must be dedicated to such activities as data collection, analyzing, reporting and decision making.



To reduce these emerging costs for integrating the vehicle software and the manufacturing processes, there is a need to gain a better understanding of the mechanisms in the development process that have an influence on these time critical and resource consuming activities. Accordingly, the main objective of this study is to identify what the key improvement issues are regarding the integration of manufacturing and PD, especially in the design and development process for software-intensive automotive systems.

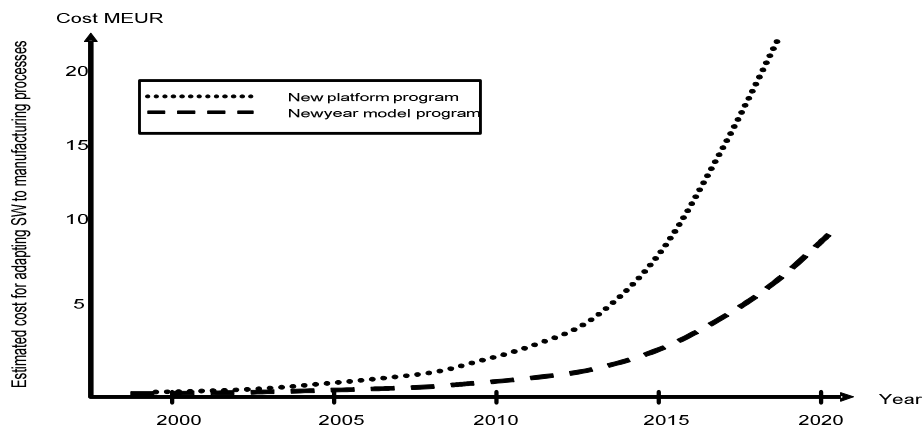


Fig. 2. Estimated rises in cost for adapting software to manufacturing processes in new platform and new year model programs.

### **2.5. Case Company Description**

This study was carried out at VCC and VTC. The companies have a long history of developing vehicles and their internal cultures are influenced by the Swedish cultural heritage. Although their organizations and range of products differ, they are characterized as matrix organizations where work is organized by functional (e.g., design and manufacturing engineers) and project (e.g., vehicle program) criteria. Further, the companies have operations in their manufacturing processes that are affected by vehicle software.

VCC is a premium car manufacturing company and has approx. 25,000 employees all over the world and produces roughly 450,000 cars per year (2011) [69]. 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) [70].

## **3. Methodology**

The rapid growth of software in vehicles and the increasing demands on efficient launches of new vehicle programs in production make it interesting to investigate the PD/manufacturing interface in the design and development of software-intensive

automotive systems. Hence, the broad objective of this study is to answer following general research question.

*RQ: What main challenges exist in the interface between the functions for PD and manufacturing in engineering, design and development of software-intensive automotive systems when the number and complexity of these systems will increase in future vehicles?*

Although the PD and manufacturing interface is also influenced by integration of other functions such as marketing and suppliers, these interfaces are not scrutinized in this work. Further, the question deals with the design and development processes involving concept study, pre-study and industrialization of vehicle programs. That is, pre- and post-design development is not considered in this study.

This general research question explores the area of interest from a holistic perspective. Since design processes involve, among other things, people, organizations and tools, which are associated with various scientific areas, e.g., social science and engineering, design research is highly multidisciplinary (see e.g., [11]) and the research requires a holistic approach [62]. To decompose this general research question into less comprehensive sub-questions without losing the holistic perspective, LPDS [48] and its three sub-systems: (1) People, (2) Process and (3) Tools and Technology, were used. Hence, the general research question is decomposed into three sub-questions as presented in Table 2.

Table 1. Sub-questions.

Research Question	Capacity
RQ1: What issues can be identified in relation to people in the PD/manufacturing interface in design and development of automotive software-intensive systems?	RQ1 concerns issues related to people and comprises the characteristics of the individuals in an organization. In this study, RQ1 primarily investigates the employees' knowledge and skills and the mutual understanding between the functions for manufacturing and PD with respect to the software development and manufacturing processes. Training programs, recruitment and ability to transfer knowledge are examples of factors that influence the mutual understanding.
RQ2: What issues can be identified in relation to processes in the PD/manufacturing interface in design and development of automotive software-intensive systems?	RQ2 deals with issues that can be referred to processes involving the sequence of steps that are required to bring a product from concept to start of production. In this study, RQ2 particularly investigates the integration of PD and manufacturing in the actual processes (both documented and informal) from design and development of new software technologies in vehicles to the finished product built by manufacturing, for example, requirements handling, software-intensive systems integration and testing and prototypes built.
RQ3: What issues can be identified in relation to tools and technology in the PD/manufacturing interface in design and development of automotive software-intensive systems?	RQ3 investigates issues related to tools and technologies of an organization employed to produce a desired outcome. This research question is explored by looking at tools and technology for such activities as requirements handling, model based development, prototyping and testing of software-intensive systems. Also "soft" tools, for example, problem solving, learning and standardization of best practices are included.

To answer the research questions, the methodology used in this inquiry is primarily based on iFLAP, originally developed by Gorschek and Wohlin [26][27] and further validated and refined by Pettersson et al. [54]. IFLAP is a light-weight framework that relies on an inductive bottom-up approach developed to guide practitioners in process improvement initiatives involving process assessment and improvement planning. Further, the underlying research strategy used in iFLAP is case research [77]. In this work, it was found appropriate to adopt a bottom-up, or inductive, approach mainly owing to three factors.

First, iFLAP and its predecessors have taken a bottom-up approach such as Basili's Quality Improvement Paradigm (QIP) [5], which has been proven to be scalable and useful in industry, especially in the automotive domain [26][27][54]. The second factor concerns the studied organizations' limitations in allocating the necessary resources for improvement efforts. Commencing top-down SPI initiatives such as CMMI [16] or Automotive SPICE [3] implies that a large amount of resources must be committed [21][78]. In comparison, the resource usage (including company staff and assessors) for the assessment presented in this paper was ~400 person-hours. Third, top-down SPI initiatives do not involve validation of improvement issues identified through e.g., triangulation of data points, and do not include techniques that identify what should be improved first, e.g., prioritizing and dependency mapping, based on specific organizational needs. Further, earlier work assessing the area of interest has not been found and the area investigated is interdisciplinary, involving different engineering disciplines such as software, system and manufacturing engineering (ME). This increases the complexity of identifying prescriptive and standardized best practices that are suitable across the functions of PD and manufacturing and the engineering disciplines concerned. The iFLAP process consists of three main consecutive steps: (1) Selection—includes selection of relevant cases such as organizations, projects and roles for the assessment, (2) Assessment—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—involves prioritization and dependency mapping of triangulated improvement issues that generate packages of improvement issues and outline the agenda for what should be improved first. The study presented in this paper reports and analyzes the results of the assessment in the area of interest, and thus only steps one and two were carried out. The steps in the iFLAP process are shown in Figure 3, and the execution of them is described in the following.

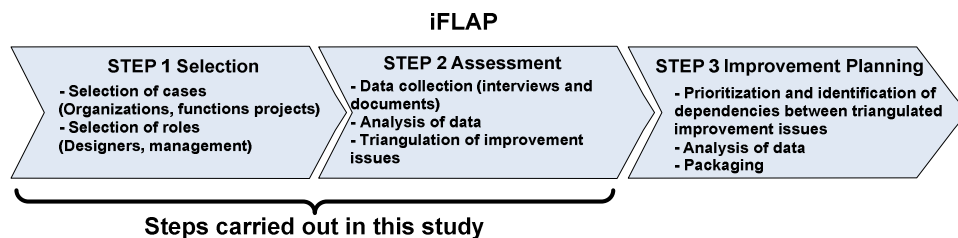


Fig. 3. IFLAP adopted from Pettersson et al. [58].

### 3.1. Step 1—Selection of cases and roles

The tactics for selecting cases were based on criteria other than representativeness, since random or stratified collection from an identified population is not feasible in case research [20][77]. Instead, the inquiry strived for literal replication primarily by using an inductive approach as described by Yin [77]. Each case was thus selected on the basis of contextual similarities rather than diversities. Moreover, resources available for the project were considered by utilizing convenience sampling of the cases.

The objective in the selection of participants was to pinpoint and cover all the roles according to their involvement, or importance, in the interaction between manufacturing and PD at the companies. This was achieved at VCC by examining the roles in the software release process together with representatives of the organization studied. The selection of interviewees at VTC was made primarily on the basis of expert judgment by representatives working in the organization. It was preferred that the number of participants representing each function would be fairly balanced. A balanced distribution was difficult to achieve at VCC, however, since the availability of staff in some roles was limited. This imbalance was alleviated by selecting three participants at PD with experience of ME of E/E systems. Table 3 shows the distribution of the selected subjects at the companies. Further, the average time at the company and the experience from development of software-intensive automotive systems among the participants were 14 and eight years, respectively.

Table 2. Distribution of selected stakeholders.

Characteristics of the subjects	Case Company	
	Volvo Truck Corporation (8)	Volvo Car Corporation (12)
Organizational affiliation	PD (4), Manufacturing (4)	PD (10), Manufacturing (2)
Roles	Design Engineers (2), Program Manager (1), Configuration Manager (1), Manufacturing engineers (3), Line Manager (1)	Design Engineers (4), Line Manager (1), Program Manager (4), Configuration Manager (1), Manufacturing engineers (2)
Time at the company	Less than 1 year (0) 1-5 years (2) 6-10 years (3) >10 years (3)	Less than 1 year (0) 1-5 years (2) 6-10 years (3) >10 years (7)
Experience of automotive software development	Less than 1 year (0) 1-5 years (2) 6-10 years (3) >10 years (3)	Less than 1 year (0) 1-5 years (3) 6-10 years (3) >10 years (6)
Gender	Male (7), Female (1)	Male (10), Female (2)

### 3.2. Step 2—Assessment

Five major data sources types were used in the assessment. They are data from documentation at VCC and interviews with participants representing PD and manufacturing at VTC and VCC. The primary data source was in-depth semi-structured

interviews with the selected stakeholders, since this qualitative method has the potential to provide rich and highly illuminating material [59]. An interview guide was developed and used during the interviews, and the included questions were asked to all of the interviewees regardless of their role. To obtain a top-down view of the area of interest, the interview guide followed a funnel analogy with general questions at beginning of the guide (e.g., questions about requirements engineering and testing in automotive software development) and more specific ones at the end (questions addressing the interaction between PD and manufacturing). In addition to the interviews, pertinent documentation and archival records were used to augment and corroborate interview data and triangulate data sources. However, complementing documents from VTC were difficult to find and interpret due to a lack of research resources in these activities.

Twelve participants were interviewed on the VCC site over a period of one month (March 2007) and eight interviews were held at VTC over a period of two months (December 2007-January 2008). All the interviews (except two) were recorded and held in Swedish with two interviewers (except for two of them owing to a lack of resources), where one was responsible for the interview process and the other took extensive notes. The interview time varied between 55 and 120 min with a mean time of 90 min.

The data analysis was based on the process described in Eisenhard [20], involving 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. This was found appropriate mainly because the within-case analysis of the first case studied could be used in the second case. In addition, the analysis consists of a third step where the issues were refined and adapted for the subsequent improvement planning (Step 3 in Figure 3). Figure 4 shows the analysis steps.

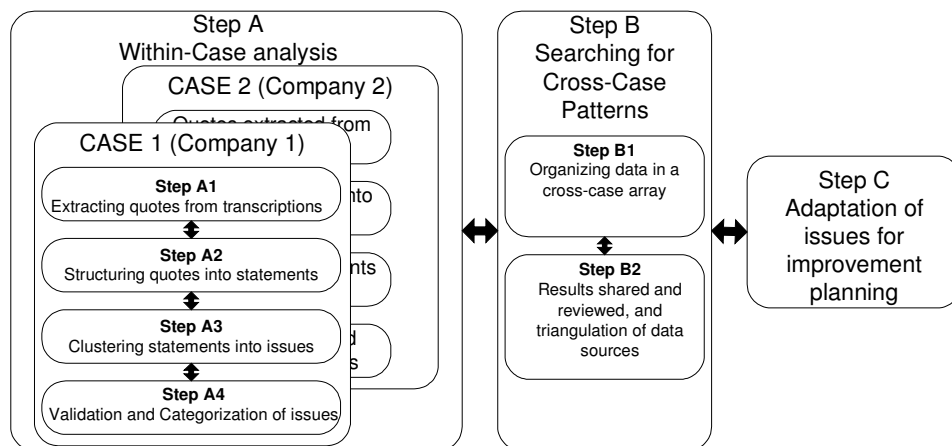


Fig.4. Overview of analysis steps.

The within-case analysis (Step A) was influenced by the principles of grounded theory [25], i.e. theories grounded in data, since this inquiry has an explorative approach. 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 [77] 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 and can be based on three strategies: (1) relying on theoretical propositions, (2) rival explanations and (3) case descriptions. The first strategy was chosen for the analysis by relying on the theoretical proposition for this study, which claims that there exist challenges in the interface between manufacturing and PD during design and development of software-intensive automotive systems. The data analysis technique was based on the approach described by Miles and Huberman [47] with three concurrent flows of activity: data reduction, data display and conclusion drawing/verification.

The recorded interviews were transcribed before the analysis and varied between six and ten pages in length. The transcription was done by one of the researchers since it was not possible to allocate extra resources for this time-consuming work. Tapes, transcripts and notes were stored in a case study database [77]. The analysis of interview data was divided into four main sub-steps in the within-case analysis, where all of them were iterated several times as shown in Figure 4.

Step A1 (extracting quotes from transcriptions) included filtering of data by extracting quotes from passages in the raw data that could in some way be factors that influence the area to be assessed. Example quote: "In project x, we did not understand how the production requirements would affect us when we wanted to configure our system x in the factory".

In Step A2 (structuring quotes into statements), the remaining information was scrutinized, gathered in a list and sorted into statements with an identification number and a description. To be able to trace the statements, the list also contained references to passages in the raw data. Further, quotes containing detailed company specific terminology were summarized and reformulated so that the statements could be described in one or two sentences. Example statement: Unclear production requirements make it difficult to configure software-intensive systems

Step A3 (clustering statements into issues) comprised aggregation of statements that were displayed as issues at a higher level of abstraction. Example issue: There are difficulties in incorporating production requirements and requirements for software-intensive systems. To enhance the validity of the results, phases one to three were carried out separately and in parallel by two researchers.

In Step A4 (validation and categorization of issues), the two researchers reviewed and discussed their results together with two other researchers and validated the issues through triangulation of the data sources used in each case. Moreover, the issues were sorted into categories on the upper level that were primarily based on the sub-processes to the product and manufacturing development process at one of the organizations studied,

which was found appropriate for both organizations. Categories on lower levels emerged from the analysis of the data. For example, the issue exemplified above was categorized in Manufacturing Development Activities\Requirements Engineering.

The document analysis was guided by the results extracted from the interviews and covered archival records with project information and documentation describing processes and instructions primarily related to the development of automotive software based systems. For each of the issues identified, relevant documentation was examined to check whether it indeed confirmed the issue. If the documentation contained what was included in an issue, it was considered to support it. Similarly, if the documentation contradicted what was included, it was considered not to support the issue.

The cross-case search for patterns (Step B) consisted of two main sub-steps and aimed to illuminate interrelated issues that were both supportive and contradictory in the two cases. In Step B1, the two researchers extracted relevant data from the within-case analysis, i.e. documents and statements made in the interviews about the area of interest, and organized these data in a cross-case array as suggested in Voss [71]. The remaining data were merged into issues with cross references to tagged passages in interviews and pertinent documents. Since this study has a flexible design, as discussed in Robson [59], research procedures and questions that emerged were described in a case study protocol based on the recommendations of Yin [77] and attached to each issue together with the initial purpose and research questions of the study. In Step B2, the researchers discussed and analyzed their results together with two other researchers and representatives of the companies. To increase the validity, the steps in issue derivation were reviewed and data were analyzed across the cases through triangulation of all five data sources. A threshold of at least two supporting data sources was judged appropriate for considering an improvement issue to be confirmed.

In the final analysis step (Step C), the results of the cross-case search for patterns (Step B) reported in [52] were adjusted for the improvement planning. The number of issues was deemed to be too high to efficiently carry out the improvement planning, however, and the abstraction levels between them deviated too much. In consequence, two researchers with experience of SPI were engaged in a series of meetings. This was an iterative process where different perspectives and categories were used to tweak the issues by splitting and merging them in various ways. For example, the main categories were replaced by the three generic categories of People, Process and Technologies and tools, and the issues themselves constituted the sub-category (e.g., requirements engineering) which in turn resulted in the decomposition of the general research question into the three sub-questions presented in Table 1. To ensure the quality of the final list of confirmed issues, it was reviewed by industry representatives who agreed upon the list after some minor changes (e.g., formulations and explanations).

#### 4. Results

The process assessment resulted in a set of nine confirmed improvement issues which are listed in Table 4. Each issue is tagged with an ID and a name and is explained in an overall description and examples pertaining to the issue. Table 3 gives an overview of how each issue is supported by the triangulated data sources that are given (the number of supporting participants is put in brackets). Overall, Table 3 shows that a majority of the nine issues (six of nine) are supported by all the data sources and eight of nine are supported by at least one data source in each case. At one of the companies, manufacturing does not support two of the nine issues (I4 *Quality Control in Component Development* and I8 *Adoption of New Technologies*). However, considering that three of the participants from PD at this company had experience of ME of E/E systems, all of the issues are supported by manufacturing. Despite the lack of evidence across the cases in issue I8 *Adoption of New Technologies*, it was viewed as a confirmed improvement issue owing to the support of all three data sources at one of the companies, when the experience of ME of E/E systems among three of the interviewees at PD was considered. The single case support of this issue may be explained by a recent launch of a new vehicle program at this company.

Table 3. Triangulation of improvement issues.

Issue ID	Case 1			Case 2		Supportive data sources
	Interview study		Document study	Interview study		
	PD	Manufacturing		PD	Manufacturing	
I1: RE	X(10)	X(2)	X	X(4)	X(3)	5
I2: EMInv	X(10)	X(2)	X	X(3)	X(4)	5
I3: V&V - ManProc	X(8)	X(2)	X	X(3)	X(4)	5
I4: QC - CompDev	X(10)		X	X(3)	X(2)	4
I5: DevMan&PDSys	X(8)	X(1)	X	X(4)	X(4)	5
I6: KnowDev	X(10)	X(2)		X(4)	X(4)	4
I7: ResAlloc	X(4)	X(1)	X	X(3)	X(2)	5
I8: AdoptNewTech	X(5)		X			2
I9: Roles&Resp	X(10)	X(2)	X	X(4)	X(4)	5



Table 4. Confirmed improvement issues.

ID	Name and Description	Examples
I1:RE	<u>Requirement Engineering</u> : There are problems in eliciting, specifying and communicating requirements. Requirements elicitation and specification involves discovering, collecting and documenting the needs of all stakeholders ( e.g. manufacturing and customer service) and design constraints. Communication concerns requirements analysis and negotiation, intra-company as well as extra-company.	Ex1: Communication of manufacturing requirements are mainly informal ("something you hear about") and specified based on experience ("something you learn the hard way") which makes it difficult to interpret their impact on the design of SW and HW solutions. Ex2: Unclear requirements in early phases makes it hard to understand, interpret and implement what the customer need and want. Ex3: There are difficulties in writing and communicating requirements so the sub-contractors know what we want and can deliver desired quality.
I2:EMInv	<u>Early Manufacturing Involvement</u> : To effectively secure product manufacturability there is a need of early manufacturing involvement in product development that critically assesses manufacturing processes in relation to desired product design.	Ex1: There is a need for implementing an "eye of a needle" early in the development process where configuration and test methods for SW and HW solutions in the manufacturing process were specifically forced to be handled. Ex2: There are risks for not giving priority in early product development to configuration of SW and HW solutions in the manufacturing process, for example, assembly plant software download and calibration. Ex3: There are problems in mapping the manufacturing organization to the product development organizations which leads to difficulties in obtaining multidisciplinary project teams in early product development.
I3:V&V - ManProc	<u>Verification and Validation of Manufacturing Processes</u> : This issue concerns securing manufacturing readiness regarding , for example, tools and methods for testing and configuring SW and HW solutions in the manufacturing process.	Ex1: The implementation status of SW and HW solutions in vehicles and tools/methods in manufacturing are not sufficient for adequate verification and validation of manufacturing processes. Ex2: Potential risk for late changes since test and configuration methods in the manufacturing processes are developed after decided design and implementation of SW and HW solutions. Ex3: Difficult to verify and validate manufacturing processes of all variants that SW and HW solutions enable.
I4:OC - CompDev	<u>Quality Control in Component Development</u> : This issue concerns the control systems for securing that development of components and sub-systems based on SW and HW solutions meet specifications and obtain desired quality.	Ex1: Methods and tools for requirements reviews, code inspections and verifications of SW and HW integration and incremental systems tests are insufficient. Ex2: Inability to inspect and gain a clear insight into the sub-contractors' processes for verification and validation of components/systems based on SW and HW solutions give difficulties in controlling that they obtain desired quality of deliveries. Ex3: There are difficulties in verifying and validating all variants that SW and HW solutions enable
I5:DevMan& PDSys	<u>Development of Manufacturing and Product Development Systems</u> : To be able to effectively manage the rapid growth of SW and HW solutions in vehicles, there is a need to take this evolution into consideration in the development of systems for manufacturing and product development.	Ex1: To reduce HW variants and enable rapid handling of quality issues, there is a need for increased ability of manufacturing processes and tools to configure vehicles more efficiently. Ex2: Developed methods of working and tools in product development tend to be suited for mechanics and not to be adapted for development of software and electronics. Ex3: To reduce complexity and facilitate the adaption between SW and HW solutions and manufacturing processes, there is need for achieving communality in the plants.
I6:KnowDev	<u>Knowledge Development</u> : This issue concerns the capability of creating, capturing, sharing and reusing knowledge throughout the business.	Ex1: Problems tend to reoccur in vehicle programs caused by insufficient sharing and reusing of knowledge across programs, functions and disciplines. Ex2: Lack of understanding for development of SW and HW solutions outside the units working with SW, for example, (e.g. Manufacturing and other disciplines than electrical). Ex3: Too little knowledge about the manufacturing processes in the PD organization leads to difficulties in introducing new SW and HW solutions in the production.
I7:ResAlloc	<u>Resource Allocation</u> : Necessary resources are not allocated to early product development phases.	Ex1: Most of the resources are used to clean up and finalize programs in late phases ("fire fighting") which snatches resources from programs in early phases. Ex2: Lack of resources in early product development makes it difficult to accomplish necessary early product development work such as review of requirements and analyses of manufacturability (e.g. DFMA). Ex3: Insufficient early product development work propagates wishful engineering which is characterized as testing for quality in late phases and unplanned and costly design loopbacks.
I8:Adopt NewTech	<u>Adoption of New Technologies</u> : This issue concerns the capabilities for adopting of new technologies such as transfer of new technologies from technology development to product development.	Ex1: New technologies based on SW and HW solutions are not verified when they are taken into vehicle programs which causes late changes and costly loopbacks. Ex2: Some vehicle programs involve simultaneous development of immature technologies in vehicles and manufacturing processes which increases the complexity and the risk for goal failure. Ex3: There are problems in predicting and adapting the organization (e.g. competence, structure) at the same pace as new technologies based on HW and SW solutions are introduced.
I9:Roles& Resp	<u>Roles and Responsibilities</u> : Unclear definitions and allocations of roles and responsibilities and difficulties in understanding who is accountable for what	Ex1: Processes and instructions for quality assured release of software are not always followed. Ex2: Engineers have difficulties in making out how and which decisions made by line managers and project managers are affecting their tasks. Ex3: Unclear allocation of responsibilities between manufacturing and PD regarding adaption of SW and HW solutions and manufacturing processes.

## 5. Analysis and Discussion

To enrich the understanding of the studied context, this section addresses the research questions by analyzing and discussing the improvement issues in relation to the state-of-the-art. Table 5 gives an overview of how each issue is linked to the research questions and the section in which the analysis and discussion are provided. In spite of the relationships between the issues and research questions, they were found to be somewhat intertwined, Table 5 discloses the linkages that were conceived as most coherent. Accordingly, some of the issues (I1, I3, I4 and I5) can be linked to both RQ2 and RQ3

which are presented in Section 5.4. This section concludes with a validity evaluation in Section 5.5 and a summary in Section 5.6.

Table 5. Overview of links between issues, research questions, and in which section the analysis and discussion are provided.

Issue ID	Research question	Section
I1: RE	RQ2 and RQ3 (processes and tools&technologies)	5.4
I2: EMInv	RQ2 (processes).	5.2
I3: V&V-ManProc	RQ2 and RQ3 (processes and tools&technologies)	5.4
I4: QC-CompDev	RQ2 and RQ3 (processes and tools&technologies)	5.4
I5: DevMan &PDSys	RQ2 and RQ3 (processes and tools&technologies)	5.4
I6: Know Dev	RQ1 (people)	5.1
I7: Res Alloc	RQ1 (people)	5.1
I8: Adopt NewTech	RQ3 (tools&technologies)	5.3
I9: Roles&Resp	RQ1 (people)	5.1

### 5.1. RQ1: Issues related to people

**Issue 6 Knowledge Development:** Participants at both companies acknowledged that the level of software competence varies in the manufacturing organization. The overall level of software development know-how influences the possibility of obtaining the right understanding of, and attitude towards, quality-assured handling of software related faults in the plants. In addition, most of the interviewees at both companies highlighted the importance of understanding and having insight into the prerequisites of manufacturing vehicles. One of the participants said, "A software designer without knowledge about production of vehicles can be more damaging than useful".

According to Hansen et al. [39] there are two main strategies for knowledge management: codification and personalization. Codification involves systematization and storage of explicit (codified) knowledge, and makes it assessable and easy to use by anyone at the company. In the personalization strategy, the flow of tacit knowledge is supported by facilitating contacts between units, functions and individuals. Nonaka [49] found that organizational knowledge is created through a continuous dialogue between tacit and explicit knowledge.

There are a number of approaches that are related to the codification strategy. For example, lean companies emphasize the value of building knowledge in PD by turning new knowledge from projects into knowledge standards that can be reused in subsequent projects [42]. At Toyota this is commonly accomplished by establishing know-how databases evolved from checklists and A3s [42][48]. Similarly, in software development, reusing life cycle experience, processes and products for software development is often referred to as having an Experience Factory (EF) [6]. However, implementing and maintaining these kinds of efforts imply a major undertaking in an organization, as this requires dedicating a considerable amount of resources to activities such as maintaining the quality and accuracy of project data and training [6].

In work associated more with the personalization strategy, Vandeveld and Van Dierdonk [68] conclude that empathy on the part of PD towards manufacturing is one of the contributors to a smooth start of production. To obtain a mutual understanding of each other's work (e.g., design and manufacturing engineers), it is important to have experience of different functions via, for example, job rotation [13][14][48].

An organization's ability to transfer knowledge is also an important aspect related to knowledge management. Morgan and Liker [48], in LPDS model principle 9, point out the importance of developing, diffusing and maintaining tacit knowledge in the organization by creating learning networks that facilitate knowledge transfer from, for example, PD to manufacturing.

The adapted network model presented in Figure 5 can be used to analyze the capability of the organizations studied to create learning networks. The model was derived from organizational charts and interviews at the organizations assessed, and it discloses a clustering of manufacturing actors on the one side and PD, purchase, suppliers and product planning on the other side in the operational area (here called the PD cluster). It can be seen that the network model indicates a low-density network between the PD and manufacturing clusters where the ME unit has a high degree of between-ness centrality. Meyer and Rowan [46] suggest that, as density increases, shared norms and behavior tend to diffuse across the network. Hence it may be possible to anticipate that the issues discussed here can be related to the low density. However, to implement appropriate improvements that facilitate the diffusion of knowledge between the organizations, further investigations must be carried out of how knowledge is transferred through informal processes and communication channels on deeper structural levels.

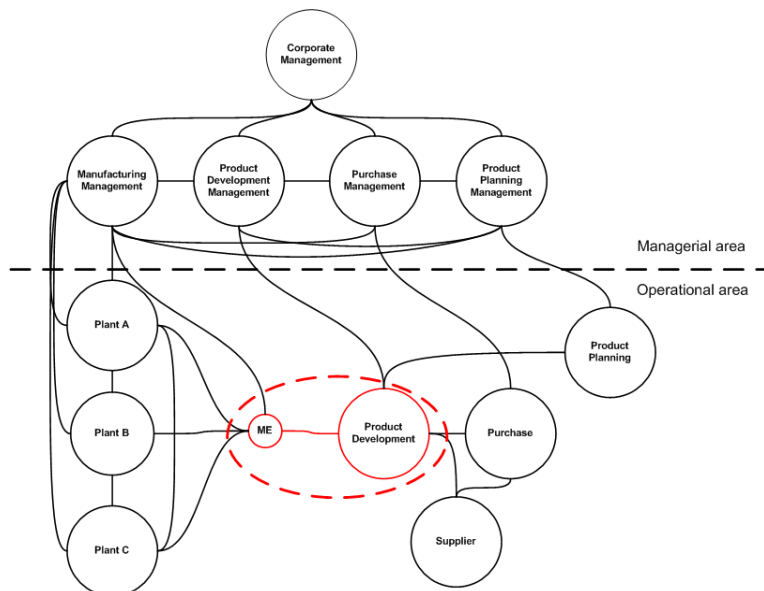


Fig.5. Adapted network model.

There seem to be two recommendations that can be related to the codification and personalization strategies. The first recommendation concerns the organization's ability to capture, express and store experience from projects and make use of it in subsequent projects. Second, it is recommended that mutual understanding of the work done by design engineers and manufacturing engineers should be given attention by considering the conditions for the recruitment and training of engineers.

**Issue 7 Resource Allocation:** The interviewees emphasized the importance of having well-functioning project teams with stakeholders who represent disciplines involved in product development. However, respondents from both companies experienced that most of the resources are spent in late phases of the projects. Some of the interviewees mentioned that 70 to 80% of the project activities are allocated to the industrialization phase. In particular, respondents at both companies experienced difficulties in allocating manufacturing resources in early phases of the projects due to a lack of resources. The importance of dedicating resources to involve manufacturing in product development is one of the critical factors identified by Lakemond et al. [44].

The adapted network model shown in Figure 5 can be used to analyze the conditions for allocating necessary manufacturing resources in early development phases. One observation in the operational area is the single tie between the clusters of PD and manufacturing passing through the ME unit. According to Wellman [73], this means that the ME unit possesses a so-called broker role. Wellman argues that brokers' marginal position may aggravate their capabilities to access necessary resources, thus impeding the ME unit's possibility to bridge the manufacturing and PD clusters. In fact, organizational data and the responses of the interviewees indicated a difference in proportions between the PD organization and ME unit, which is illustrated in Figure 5 (the size of the bubbles roughly indicates the number of employees).

The observed imbalance in the differences in size between the PD organization and the ME unit has to do with the formal organization. Thus, most of the interviewees highlighted the importance of interpersonal relationships between manufacturing (in particular ME) and PD that form an indispensable informal network system. To facilitate appropriate informal and formal participation of ME resources in projects, some of the respondents even said that the ME unit should instead be an integrated part of the PD organization.

Leveling out the imbalance between ME and PD by recruiting more personnel is most likely not feasible. Instead, it is essential to encourage the building of informal networks facilitating interpersonal relationships and knowledge sharing by reducing the barriers between PD and manufacturing (e.g., geographical and organizational dispersion and information flow). Clear roles and responsibilities in the PD/manufacturing interface also have an impact on how allocated resources can be utilized more efficiently; see below.

**Issue 9 Roles and Responsibilities:** The respondents brought up uncertainties regarding roles and responsibilities within and across functions in a project team. Specifically, the

respondents mentioned that the responsibilities between manufacturing and PD that ensure a fit between the design of the software-intensive systems and the manufacturing processes are unclear. Moreover, the respondents described that well-elaborated and established processes were not always followed and, in some cases, work was based more on how colleagues conducted it. As one of the respondents expressed it, "The processes are OK, it's the individuals in the processes that create problems through, for example, miscomprehensions or not understanding what is expected of them".

Vandevelde and Van Dierdonk [68] point out that formalization, for instance, in the form of clear roles and responsibilities in the PD/manufacturing interface, is a key factor for smooth production start-up. Although formalization can impede innovative environments in the design phase, they argue that, when the design is introduced in production, it is beneficial to formalize the processes. Further, Lakemond et al. [44] similarly conclude that, by primarily focusing on technology transfer, five of the six most important factors for integration of PD and manufacturing are seen to be related to transfer management involving clearly defined roles and responsibilities.

Another comment given by the respondents was the importance of making clear the decision structure between the project and line organization for software related issues. Both companies are based on a matrix organization involving a line organization responsible for the development of technologies and processes and a project organization that drives new vehicle and platform projects. However, the matrix organization implies conflicts between two competing structural principles and violates, unity-of-command principle of administrative management theorists' (Fayol [23]), which specifies that no organizational participant should receive orders from more than one superior. Moreover, the high degree of uncertainty and changeability associated with the development of software makes it even more difficult to take in and implement decisions related to software.

## **5.2. RQ2: Process related issues**

**Issue 2 Early Manufacturing Involvement:** Participants at both companies brought up the importance of incorporating manufacturing aspects as early as possible in the development of software-intensive systems – not only because of the necessity of securing that the product becomes manufacturable, but also to elicit manufacturing knowledge and experience in early stages that can be beneficial to customers.

The companies' ability to reduce time-to-market has become one of their most important capabilities. To accelerate the PD process, a number of authors in the literature, e.g., [40][63][74], propose a more concurrent and multidisciplinary approach to the development of products that maximizes the use of enabling technologies such as rapid prototyping and cross-functional techniques, among them quality function deployment (QFD), failure mode and effect analyses (FMEA), Design for Manufacture and Assembly (DFMA) etc.

The study done by Sobek et al. [63] discovered that Toyota considers sets of possible designs in the early development phases, which gradually converge to one final design. They termed this approach SBCE. Condensed from Sobek et al. [63], Figure 6 depicts the characteristics of SBCE with its parts A to D simplified by only two functions: design engineering and ME.

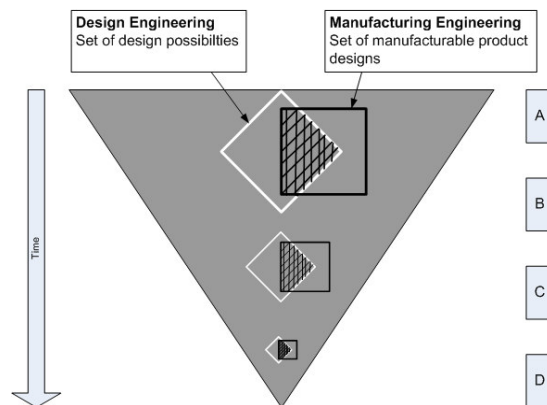


Fig.6. SBCE condensed from Sobek et al. [63].

Similarly, the Requirements Abstraction Model (RAM) [28][30]—see also Section 5.4—embodies early rejection of software requirements by comparison to company strategies, and tries to reach the best trade-offs in terms of what to include in the product/solution by consolidating different stakeholders' views, negotiations between stakeholders involving evaluation, and trade-offs between requirements and alternative solutions. Another activity related to this in software development is usually to give priority to requirements where the ultimate goal is to deliver the right product to the market at the right time [57]. There are several techniques that can be used to assign priorities to software requirements, e.g., the Analytical Hierarchy Process (AHP) [61], Cumulative Voting (CV) [9] and the Planning Game [7].

To resolve manufacturability issues in the early phases of PD, both companies endeavor to obtain early manufacturing involvement. However, instead of jointly solving the problem on the basis of early information from PD, manufacturing engineers can have a tendency to wait for implementation-ready solutions, which forces them to push manufacturing prerequisites through in late phases. In this vein, one respondent from PD said, "Sometimes the demands from manufacturing are perceived as an unpleasant surprise since manufacturing has not yet actively discussed them in the project". Another comment from PD was that "production sometimes tends to only report the problems and believe that PD will just solve them". In contrast, some of the respondents from manufacturing perceived that they have to "hunt for" information from PD. Referring to Wheelwright and Clark [74], the companies seem to be struggling somewhere between mode two (early involvement) and three (integrated problem solving) depending on, for example, the complexity of the product and the capability of understanding the impact on

the manufacturing processes in early phases of the development of software-intensive systems.

### **5.3. RQ3: Issues related to Tools and Technology**

**Issue 8 Adoption of new technologies:** Some of the interviewees experienced that new technologies introduced in vehicle projects are not mature and have to be developed simultaneously in new vehicle programs. Respondents also stressed that new and complex changes in the products simultaneously drive changes in the manufacturing systems. For instance, a change in the E/E architecture and the communication interfaces involved forces the manufacturing systems to adapt to, for example, new techniques and tools for configuring and testing vehicles.

Lakemond et al. [44] address technology transfer between the technology development/PD interface and the PD/manufacturing interface. To manage both interfaces, Lakemond et al. [44] formulate a tentative interface management model that includes six contextual factors. Each of these factors can be assessed by several statements generating a ranking of the project that indicates the degree of difficulty relative to other projects. The model outlines the risks in a certain project and provides management with a set of recommendations that guards against the risks.

Technology-push entails that innovation is driven by science, which in turn drives technology and application while market-pull poses the opposite, that user demand is the primary factor and that markets, users and applications are, or should be, the key drivers of innovation [31][58]. Reflecting on the PD/manufacturing interface in terms of technology management from a technology-push and market-pull perspective, it seems that PD demands primarily drive technology development of manufacturing systems and processes affected by software-intensive systems. As one of the manufacturing engineers put it, "We generally only upgrade our manufacturing systems when PD comes to us with a new product/system". Apparently, there is an imbalance between technology-push and market-pull in product and manufacturing innovations in the area investigated here.

### **5.4. RQ2 and RQ3: Issues related to processes and tools&technology**

**Issue 1 Requirements Engineering:** In the literature, requirements engineering (RE) constitutes a central and important part of the product development process [64][66]. The RE process usually involves four main elements [64]: (1) elicitation and specification, (2) analysis and negotiation, (3) validation, and (4) management.

A concern brought up by the interviewees at both companies was the process for requirements elicitation and specification. For example, ambiguous product requirements from pre-studies in vehicle projects cause difficulties in interpreting and implementing correct functionalities when they are realized. To improve the analysis and negotiation of requirements between various stakeholders, i.e. answering the question "do we have the right requirements" [43], the participants emphasized a need to specify the requirements

on an adequate abstraction level and to communicate them in an understandable and consistent way.

RE in automotive development is complex because it involves several stakeholders, e.g., design, PD, suppliers, manufacturing and after-market, and is highly multidisciplinary, for example including mechanical, electrical and software engineering. Automakers usually adopt a formal approach where the breakdown of product requirements follows a top-down process starting with overall business and user requirements (e.g., product strategies and legal and customer demands on safety and the environment) derived from the business development process, and ending in component requirements on hardware and software solutions, for example, via complete vehicle attributes and function/systems requirements. The process is seldom as straightforward as that, however, since there are often already settled requirements on underlying levels that must be captured and considered. For instance, while requirements in complete vehicle attributes such as fuel consumption may affect requirements on lower levels, there are underlying requirements on lower levels derived from such things as E/E architectural or manufacturing constraints, which have an impact on requirements on higher levels. In addition, requirements are frequently changed and updated on various levels of abstraction throughout the course of product development. Meeting the high speed of the evolution of new and improved innovations realized with automotive software and the changeable nature of software makes the handling of requirements even more complex.

Several methods have been created to manage requirements that have a rather formal approach. These methods assume that the elicitation and specification process of requirements start with goals that are subsequently on lower levels of abstraction [66]. Further, they do not commonly encourage the role of requirements as drivers of value and associated product solutions. Gorschek and Wohlin [28] addressed the issue that requirements can have the role of driving development and that they continuously arrive on different levels of abstraction by designing the RAM framework.

Issues relating to RE in the manufacturing and PD interface were mentioned at both companies. Specifically mentioned were difficulties in constraining manufacturing requirements to be understandable and convertible to measurable parameters for developers of software-intensive systems. This finding corresponds to one of the observations reported in [2] that considers the incorporation of manufacturing requirements, where it is claimed that production requirements are experienced-based rather than being specifications of purposes and goals, and that they describe expected results.

Morgan and Liker [48] reflect upon this in LPDS principle 4, by describing the use of engineering checklists at Toyota. Basically, the checklists are a knowledge data base that records experiences of design practices that have evolved from prior projects considering, e.g., performance requirements, production requirements and current design standards. Further, since many of the requirements do not embrace detailed specifications of parameters, the checklists primarily provide guidelines that allow the engineers to use their creativity and improve existing solutions in the spirit of kaizen [24]. Although



checklists are widespread in other companies, Morgan and Liker [48] argue that they are useless at many companies owing to poor discipline in updating and deploying them effectively. In contrast, LPD emphasizes the importance of useful checklists since they accumulate new knowledge and experiences that can be effectively used to achieve feasible solutions and improve existing design standards [42]. Further, elaborated and well-known design standards elucidate the engineering framework, which enables a shorter ramp-up time at program start.

In the same vein, Fricker et al. [22] emphasize the importance of extensive communication and negotiation over focusing on achieving perfection in requirement specifications through sometimes over-elaborated specification practices in RE. By adopting this line of reasoning, Fricker et al. [22] successfully used a technique called handshaking with implementation proposals in the industry that leverage knowledge transfer, understanding and integrated problem solving in the development of new products.

The checklists and principles of the RAM framework can be used to bridge cross-functional requirements such as product and manufacturing constraints. In the elicitation and specification of production requirements, they can serve as input to the repository of all requirements by conveying changed or new prerequisites of production. Further, to communicate requirements across functions on an appropriate level of abstraction, the checklist can constitute an extract of all requirements by selecting them on a suitable abstraction level in the requirements repository of the organization.

Another comment made by the participants had to do with a lack of appropriate tools for managing manufacturing requirements. Some of the interviewees also expressed an overall desire to improve tools that would facilitate the RE activities. Weber and Weisbrod [72] emphasize the necessity of developing adaptable tools that support engineers in their daily tasks, since there is otherwise a significant risk that users would reject the tools as well as the corresponding process improvements. Morgan and Liker [48] support this view in principle 11 and provide five sub-principles in the selection of tools by considering the other two LPDS sub-systems in the sociotechnical model—process and people. For instance, the sub-principles highlight the necessity of adapting new tools to existing systems and the fact that investments in tools are only motivated by a reduction in the staff is often counterproductive.

***Issue 3 Verification and Validation of Manufacturing Processes & Issue 4 Quality Control in Component Development:*** One of the major activities in the development of vehicles incorporates the building of prototypes in the form of digital mock-ups, physical vehicles and tools for simulation of associated manufacturing processes and tooling. This phase aims to eliminate later engineering changes, i.e. "fixes", since it is the last stage in the PD process that allows any rework of products or manufacturing processes. Both the quality of the test objects and their components/systems and the readiness of the methods and tools in manufacturing influence the verification and validation of manufacturing

processes and prototypes. Issues 3 and 4 are therefore gathered together as they are intertwined.

Morgan and Liker [48] highlight the importance of the prototype and tooling phase in LPDS principle 7. The work in this stage of the PD process is particularly influenced and guided by the lean practice of *genchi gebutsu* engineering, which constitutes one of the four core principles in the Toyota Way [45]. *Genchi gebutsu* is translated into English as "go and see for yourself to thoroughly understand the situation". The essence of this practice is that engineers responsible for a specific situation are required to observe the real scene in which it is going on and to analyze it by revealing possible problems and their causes based on their knowledge and experience. Likewise, in lean approaches to software development, Poppendieck and Poppendieck [55] emphasize regular prototyping since it creates a need for cross-functional communication and provides early feedback.

In practice, however, the influence of software-intensive systems on manufacturing processes is primarily interpreted and verified in trial production on physically built vehicles in the late stages of PD, implying a high cost and a limited number of iterations before the start of production, as described in Section 2.4. Moreover, the respondents said that the conditions for proper verification of associated manufacturing processes are not always fulfilled. This often leads to late changes ("fixes") of products and manufacturing processes that jeopardize the manufacturing performance at the worst possible time, when the market demand is peaking after the launch of a new product.

Respondents at both companies acknowledge that insufficient verification and validation of the compliance of components and systems with requirements is one of the reasons that the prerequisites of proper testing in prototype building are not fulfilled. Causes mentioned were a lack of tools and methods for verification and validation, e.g., code inspection and systems tests, and an inability to inspect and gain clear insight into sub-contractors' processes for controlling the quality of their deliveries. Moreover, to fulfill the scheduling of vehicle programs, another cause is that the time reserved for testing activities dedicated for verification of components and systems is sometimes shortened. This is in line with the study by Torkar and Mankefors [67] who found that, according to 60% of the developers, verification and validation were the first things to be neglected when there was a shortage of time in a project. There is consequently a greater risk of unplanned design loop-backs in late phases.

Sommerville and Sawyer [65] advocate that the design of test cases for verification and validation of requirements and test plans should be elaborated during initial requirement specification. Similarly, to enhance the quality of requirements enabling efficient testing, Graham [32] underlines the importance of early tester involvement in RE. Sommerville [64] suggests two main methods for verification of software-intensive systems: static verification and dynamic verification. Static verification involves inspections where the conformance of the static system with the specifications is checked while, in dynamic testing, the system is executed with test data and its operational behavior is observed. To

obtain rigorous and efficient testing, both approaches should be used in conjunction with each other [64].

However, obtaining efficient prototyping to enable the verification process of software-intensive systems and associated manufacturing processes is a challenging task owing to two broad factors. First, the vehicles become more complex since the functions that are realized by software are distributed over several ECUs communicating on different bus systems. In addition, as mentioned by some of the respondents, the software-intensive systems enable an increase of variants. A premium car typically has about 80 electronic fittings that can be ordered depending on the country etc. [12]. Simple yes/no decisions for each function yield a possible maximum of roughly 280 variants to be ordered and produced for a car. Further, in contrast to mechanical parts, which in early stages of development can be visualized in applications such as CAD/CATIA or as physical prototypes, software-intensive systems are intangible and are often described in written requirements specifications implying a higher abstraction level to be interpreted and understood.

The second factor considers the manufacturing processes, tools and applications that are affected by software embedded in vehicles. The characteristics of these manufacturing systems are often recognized as complex since they incorporate, for example, vehicle communication technologies, interaction with other IT systems, wireless data transfer and man-machine interactions. Moreover, the systems have to fulfill production demands such as user friendliness, efficiency, availability, reliability and maintainability. Being able to understand and foresee the impact on the production of products and systems with high complexity that are described in an abstract form and complex manufacturing systems requires deep knowledge and a great deal of experience among both manufacturing engineers and developers.

To accomplish earlier prototyping that underpins earlier process verification and shorter feedback loops at a reduced cost, the participants at both companies suggested integration of the manufacturing processes in model-based development. At present, there are a number of approaches and techniques used to model and validate software-intensive systems in the automotive domain, e.g., Unified Modeling Language (UML) [50], Matlab/Simulink and Hardware In the Loop (HIL). In comparison to mechanical systems, however, model-based development and testing of software-intensive systems in the automotive industry are in their infancy. For example, Broy et al. [12] point out that, because of the lack of a formalized modeling language, modeling is only applied to certain steps in PD, yielding an inefficient exploitation of its possibilities. For instance, tools supporting auto-code generation are only exploited to 10% of what they could deliver [38]. Another issue considers the lack of appropriate digital tools to support a holistic approach to model-based development by incorporating different kinds of aspects such as auto-code generation and verification of function/system requirements. Insufficient integration possibilities such as linking engineering data to models and compatibility between different tools are examples of issues that also need to be resolved. Finally, Broy et al. [12] also highlight the necessity of cost-efficient use of modeling by

reducing the probable increased work load of maintaining and building the models. This aspect was also brought up by some of the interviewees.

The above discussion indicates that it is not yet feasible to integrate manufacturing processes in model-based development since there are still issues that have to be resolved concerning the modeling of the software-intensive systems themselves. Nevertheless, despite the challenges of model-based development, it enables earlier and more frequent prototyping loops, which facilitates cross-functional lean engineering practices such as *genchi gebutsu*. To develop models that can be used by the manufacturing function, it is recommended that the five sub-principles provided in Morgan and Liker [48], principle 11, be considered in the selection of tools. Moreover, manufacturing engineers should be involved in the further development of modeling tools and working procedures since it is preferred that this development should be an evolving process rather than a big bang. In practice, it may start with an elaboration of work procedures comprising analyses based on observations of computerized visualizations of the software-intensive systems on an appropriate abstraction level (e.g., the functional level), and by using cross-functional techniques such as FMEA together with checklists and requirements containing manufacturing prerequisites.

Like today's modeling of mechanical parts, the goal of the development might be to enable export of digital models from the systems used for model-based development to the manufacturing systems, or vice versa, where models and manufacturing processes can be simulated and virtually verified. Furthermore, the systems for digital development of software-intensive systems and manufacturing processes should be seamlessly integrated, which would facilitate cross-functional access to the necessary data and applications.

***Issue 5 Development of Manufacturing and Product Development Systems:*** A review of the literature acknowledges that an overall view among researchers, e.g., Womack et al. [75] and Liker [45], is that competitive automakers have started to adopt lean principles (e.g., continuous improvement, standardization, visualization etc.) in the development of manufacturing and PD systems [45][48]. However, there are significant differences between today's PD and manufacturing operations. While manufacturing operations are characterized by sequential process chains that are performed several times in a very similar or almost identical way, PD operations are rather like a non-linear process where tasks such as design, test and redesign are performed in cycles and are highly interconnected across functions on different hierarchical levels. In addition, Carlsson [13] for example presents differences between the functions of PD and manufacturing with respect to such things as degree of structure, time horizon and level of abstraction. While there is shelf after shelf of literature on LP, e.g., [24][45][75], the same cannot be said for the literature that focuses on the application of lean thinking to PD. However, LPDS [48] and LFPD [42] are examples in the literature of work that aims to guide companies in adopting lean operations in PD.

Several of the respondents at both companies acknowledge that the manufacturing and PD systems have not been adapted at the same rapid pace as the growth of software-

intensive systems. One of the future key challenges in software engineering is that development of software must be responsive to rapid changes without compromising system quality [64]. This approach corresponds to one of the key values in agile software development [8]. Due to the flexibility of software and relatively low cost of change, unexpected system problems are often solved by enhancing software capabilities without increasing hardware cost. On the other hand, the high reliability and performance requirements on embedded real-time systems such as vehicle electronics and associated manufacturing systems imply tightly linked modules, which are sensitive to changes.

Research covering organizations' agility, i.e. how organizations manage and respond to continuous, rapid and unpredictable changes in their business environment, is discussed in [18][36][37] for example. Response ability and knowledge management are the two main capabilities that are required to practice agility [18]. Response ability is achieved through change proficiency (ability to proactively and reactively respond) and flexible relationships that are reusable, reconfigurable and scalable. In turn, knowledge management requires collaborative learning (support for collaborative networks and events) and knowledge portfolio management (management of core competences) (see also Section 5.1). To achieve organizational agility, Haeckel [36][37] suggests an adaptive enterprise approach based on four principles dealing with: organization-specific governance mechanisms, personal accountability, learning processes (sense-interpret-decide-act) and modular processes.

The participants at both companies said that differences between production units in terms of available processes and assembly sequences have an impact on managing software-intensive systems in production. Further, adapting the design to these variabilities in manufacturing generally leads to excessive work and a larger number of variants, aggravating the standardization of components and systems. The respondents also described that well-elaborated and established processes were not always followed and, in some cases, work was based more on how colleagues conducted it. Standardization is important since it reduces variation, which enables increased flexibility and predictable outcomes [42][45][48]. The importance of process discipline, implying that employees follow established and standardized working procedures, is well known in lean companies. For example, Morgan and Liker [48] claim that stable and standardized processes are prerequisites for continuous improvements (kaizen) in PD and manufacturing. However, Gunnesson [35] argues that while an agile mindset is associated with organizations' ability to simultaneously sense and respond to unpredictable changes in their environment and in being productive, the efficient use of resources focusing on the elimination of waste tends to predominate in lean operations. Consequently, to accomplish the required response to rapid changes caused by vehicle software without reducing productivity and system quality, it is necessary to pursue standardization of processes and design and to achieve process discipline among employees without suppressing the organization's agile capabilities.

### 5.5. *Validity Evaluation*

Four tests to establish the quality of case studies can be discussed in terms of construct, external and internal validity and reliability [77].

**Construct validity.** This concerns establishing correct operational measures for the concepts under study. The threats to construct validity can be expressed as respondent biases and researcher biases, which were mitigated here by primarily utilizing three different strategies discussed in Robson [59]: (1) prolonged involvement, (2) triangulation, and (3) peer debriefing.

Prolonged involvement means learning the culture and building trust. Two of the researchers worked for eight years with electrical development and electrical ME at one of the companies. This strategy was used to guard against respondent bias.

Triangulation involves the use of multiple sources that enhance the rigor of the research. The embedded multiple design [77] used in this study enables the possibility to triangulate different sources of evidence. Thus interview data from key informants at both companies representing the functions for PD and manufacturing and pertinent documents were triangulated. Observer triangulation was also utilized, since two researchers conducted the interviews and had the possibility to discuss and analyze the outcome of the interviews.

Peer debriefing means that analyses and conclusions are shared and reviewed by other researchers. This was done by conducting the analysis with two researchers who independently drew their conclusions on the basis of the data collected and shared them, where any disagreements were resolved through extensive analysis and discussion. Moreover, discussion groups were gathered in which analyses and conclusions were discussed with research colleagues, experienced SPI researchers, and members of the studied organizations who were familiar with the study. For example, issues were merged and split, and examples were added, clarifying the meaning of the issues.

In addition, to maintain a chain of evidence and facilitate a cross-case search for patterns, a well-structured case study database was established for each case on which a cross-case array was based. Four pre-interviews were also held to evaluate the correctness of the interview guide that was prepared, and to analyze the bias of one of the researcher's close relations to colleagues.

**Internal validity.** This concerns establishing causal relationships, whereby certain conditions are shown to lead to other conditions, as distinguished from spurious relationships. For example, the interviewees may not express their real opinions because they feel restricted by the recording of what they say on paper, and this poses a threat to the internal validity. This threat can be limited by guaranteeing participants anonymity in interviews. Further, the pre-interviews were used to analyze the impact of using a tape recorder. The impression was that the interviewees spoke freely and were not disturbed by the recording device. This study is primarily explorative, however, and internal validity is mainly a concern in explanatory case studies.

**External validity.** This concerns establishing the domain to which a study's findings can be generalized. The multiple case study design enhances the possibility to generalize the

results to other automotive companies. The cases selected represent two of the leading vehicle manufacturers in Sweden, where one has a leading role in the development of software-intensive systems at FMC, which is one of the world's major car manufacturers. The other company is one of the world's largest producers of heavy vehicles. This should strengthen the possibility to generalize some of the findings to the automotive domain. However, while the multiple case study presented strengthens the possibility to generalize the results, similar studies at other automotive companies may result in other findings.

**Reliability.** The objective is to be sure that, when a later investigator follows the same procedures as those used by an earlier investigator and conducts the same case study again, the later investigator should obtain the same findings and arrive at the same conclusions. Reliability can be enhanced by the use of two tactics: (1) a detailed case study protocol containing various activities carried out in connection with the study and (2) a structured case study database with all relevant data such as interview tapes, transcripts, documents etc. These tactics were used in the present case study research to increase the reliability of the results. Further, the research process is thoroughly described in Section 3 and additional information is available at the project website ([http://www.cse.chalmers.se/~pernstal/publications/2011Pernstal\\_PA\\_MAN\\_PD/pernstal\\_2011\\_process\\_assessment\\_man\\_PD.html](http://www.cse.chalmers.se/~pernstal/publications/2011Pernstal_PA_MAN_PD/pernstal_2011_process_assessment_man_PD.html)), and in [51][52].

## **5.6. Summary**

Table 6 summarizes the results of the analysis and discussion of the key improvement issues identified in relation to the research questions and the state-of-the-art. Looking at the table, an overall conclusion is that the holistic nature of the study yielded issues covering a broad range of aspects. Further, related state-of-the-art either address the issues from the software development or the PD/manufacturing interface perspective, but not in combination with each other. This indicates that the focused area in this study is relatively unexplored and immature.

Table 6. Summary of the results of the analysis and discussion.

Research question	Issue ID	Result	Related state-of-the-art
RQ1	I6: Know Dev	*There is a need to improve the organization's capability of capturing, sharing and reusing acquired knowledge from vehicle projects. *Mutual understanding of the work done by design engineers and manufacturing engineers should be given attention.	LPDS [48], LFPD [42], Experience Factory [6], Empathy [68]
	I7 Res Alloc.	*There is an imbalance in the differences in size between the PD organization and the ME unit which can be managed by clarifying roles and responsibilities and encouraging informal networks in the PD/manufacturing interface.	Formalization [68] Transfer Management [44], Network Analysis[73]
	I9 Roles &Resp	*The volatile nature of software coupled with the use of matrix organization makes it crucial that roles and responsibilities are clear in order to ensure a fit between the designs of the software-intensive systems and the manufacturing processes.	Formalization [68] Transfer Management [44]
RQ2	I2: EMInv	*Manufacturing engineers must be more actively involved in early phases by enabling and fostering active search for early information and start of the engineering of operations in their own domain.	SBCE [63], Integrated problem solving [74], Requirement Abstraction Model [28]
RQ3	I8: Adopt NewTech	*To reduce the risk of adopting new technologies in the manufacturing processes, their maturity level, when introduced in vehicle programs should be considered. *To encourage manufacturing innovations, it is necessary to achieve a two-way technology transfer between PD and manufacturing in early phases.	Interface Management Model [44] Market-pull and technology-push [31], [58]
RQ2 and RQ3	I1: RE	*Informal ad hoc procedures dominate the elicitation, communication and negotiation of manufacturing and product requirements, which calls for establishing and carefully nurturing procedures that drive the RE across PD and manufacturing in a more formal way. *There is a need of introducing better and adaptable tools that support daily RE activities across manufacturing and PD.	Requirement Abstraction Model [28], LPDS [48] Handshaking with implementation proposals [22]
	I3: V&V-ManProc and I4: QC-CompDev	*The verification and validation of software-intensive systems and manufacturing processes must be performed earlier and with higher precision and quality, and it is recommended to improve the tools and procedures for both static verification (e.g., inspections, earlier tester involvement in RE) and dynamic verification (e.g., modeling of prototypes and manufacturing processes).	LPDS [48], LFPD Test then—design [42], Static&Dynamic test [64]
	I5: DevMan &PDSys	The rapid growth of vehicle software implies that the manufacturing and PD systems need to be more responsive to change, which can be achieved through enhancing the proactive and reactive response ability, and improving knowledge management and standardizing processes and tools.	LPDS [48], LFPD [42], Agile organization [18], Adaptive organization [36], [37]



## **6. Conclusions**

This paper details experiences from current practice in, and in relation to, the interface between PD and manufacturing, focusing on the design and development of software-intensive systems in the automotive industry. The main purpose of the study was to identify key improvement issues in the area and analyze them in relation to state-of-the-art. Using iFLAP, nine confirmed improvement issues were identified in the two Swedish automotive companies (VCC and VTC) investigated.

It can be concluded that the companies have similar challenges since the triangulation yielded that eight out of nine improvement issues were supported by at least one data source type across the companies. This indicates that there may be a possibility to generalize the results to the automotive domain. Further, the holistic nature of the study yielded issues covering a broad range of aspects, which were classified into three categories: people, process, and tools and technology.

Looking at the people category, there seems to be a need to improve the capturing, sharing and reuse of knowledge and experience gained in previous projects, and to enhance the mutual understanding of each other's work and establish clearly defined roles and responsibilities in the PD/manufacturing interface. In the category of processes, both companies recognize the importance of early manufacturing involvement, but manufacturing engineers tend to act reactively by postponing the engineering of operations in their own domain until concepts and design of the software-intensive systems have been decided. In the categories of tools, technologies and processes, a major challenge identified could be traced to the requirements engineering. In particular, difficulties in specifying and mediating interpretable manufacturing requirements, and a lack of formalized procedures and suitable and useful tools that could drive and support the RE activities across the functions for manufacturing and PD were identified.

The analysis of the improvement issues in relation to the state-of-the-art resulted in a number of recommendations. However, previous key studies provide little practical guidance for resolving the problems identified in the area investigated here. That is why it is difficult to offer any specific advice to practitioners based solely on state-of-the-art in research.

This paper provides a number of findings that indicate the characteristics of the improvement issues and recommendations based on state-of-the-art that can aid further improvement efforts. Further, the characterization of the domain and companies gives insight into the automotive domain and its growing software focus. Thus the results presented here can be useful for researchers as new research areas are identified, as well as the need to apply already established solutions in a complex domain where development and manufacturing both are involved in the development of the software-intensive systems, namely vehicles.

## Acknowledgments

The authors would like to express their gratitude to all those somehow involved in this study and especially the interviewees at VCC and VTC.

## References

- [1] ABG, V-Model: Development Standard for IT-Systems of the Federal Republic of Germany, Lifecycle Process Model, 1997, <http://www.v-modell.iabg.de>.
- [2] L. Almfelt, 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).
- [3] Automotive SIG, The SPICE User Group, Automotive SPICE Process Assessment Model v2.5 and Process Reference Model v4.5, 2010, <http://www.automotivespice.com>.
- [4] AUTOSAR consortium, 2009, <http://www.autosar.org>.
- [5] V.R. Basili, *Quantitative Evaluation of Software Methodology*, (University of Maryland, College Park, MD, 1985).
- [6] V. R. Basili, G. Caldiera, and H. D. Rombach, The experience factory, in *Encyclopedia of Software Engineering*, ed. J. J. Marciniak, (John Wiley & Sons, 1994), pp. 469–476.
- [7] K. Beck, *Extreme Programming explained: embrace change*, (Addison-Wesley, Reading, MA, 2000).
- [8] K. Beck, A Cockburn, R. Jeffries, and J. Highsmith, J. Agile Manifesto, 2001, <http://www.agilemanifesto.org>.
- [9] P. Berander and A. Andrews, Requirements Prioritization, in *Engineering and Managing Software Requirements*, eds. A. Aurum and C. Wohlin, (Springer, 2005), pp. 69-94.
- [10] R. Berger, Reuse of Software in Distributed Embedded Automotive Systems, Audi 2004, in *Embedded Automotive Symp.*, Peugeot, June 23, 2004.
- [11] L.M. Blessing, A. Chakrabati and K.M. Wallace, An Overview of Descriptive Studies in Relation to a General Design Research Methodology, in *Designers: The Key to Successful Product Development*, eds. E. Frankenberg, P. Badke-Schaub, and H. Birkhofer, (Springer, 1998), pp. 42-56.
- [12] M. Broy, I.H. Kruger, A. Pretschner and C. Salzmann, Engineering Automotive Software, in *Proc. IEEE* **95**(2) (2007).
- [13] M. Carlsson, *Integration mellan konstruktion och produktion*, Licentiatuppsats, (Institutionen för Industriell Organisation, Chalmers tekniska Högskola, Göteborg, 1988).
- [14] M. Carlsson, Aspects of the Integration of Technical Functions for efficient Product Development, *R&D Management* **21**(1) (1991) 55-66.
- [15] F. Charfi and F. Sellami, Overview on Dependable Embedded Systems in Modern Automotive, in *Proc. IEEE Int. Conf. on Industrial Technology*, 2004.
- [16] 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/>.
- [17] D. Daetz, D, The effect of Product Design on Product Quality and Product Cost, *Quality Progress* **20**(6) (1987).
- [18] R. Dove, *Response Ability – The Language, Structure, and Culture of the Agile Enterprise*, (Wiley, New York, 2001).
- [19] T. Dybå, Improvisation in small software organizations, *IEEE Softw.* **17** (2000) 82-87.
- [20] K.M. Eisenhardt, Building theory from case study research, *Academy of Management Review* **14**(4) (1989) 532-550.
- [21] K. El Emam and L. Briand, Cost Benefits of Software Process Improvement, Fraunhofer—Institute for Experimental Software Engineering, ISERN-97-12, 1997.

- [22] S. Fricker, T. Gorschek, C. Byman and A. Schmidle, Handshaking with Implementation Proposals: Negotiating Requirements Understanding, *IEEE Softw.* **27**(2) (2010) 72-80.
- [23] H. Fayol, *General and Industrial Management*, (Pitman, London, 1949).
- [24] T. Fujimoto, *The Evolution of a Manufacturing System at Toyota*, (Oxford University Press, 1999).
- [25] B.G. Glaser and A.L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*, (Aldine Publishing, 1967).
- [26] T. Gorschek and C. Wohlin, Identification of Improvement Issues Using a Lightweight Triangulation Approach, in *the European Software Process Improvement Conf.*, Graz, Austria, 2003.
- [27] T. Gorschek, and C. Wohlin, Packaging Software Process Improvement Issues - A Method and a Case Study, *Software: Practice & Experience* **4** (2004) 1311-1344.
- [28] T. Gorschek, C. Wohlin, C. Requirements Abstraction Model, *Requirements Engineering journal* **11** (2006) 79-101.
- [29] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, A Model for Technology Transfer in Practice, *IEEE Softw.* **23**(6) (2006) 88-95.
- [30] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, Industry Evaluation of the Requirements Abstraction Model, *Requirements Engineering Journal* **12**(3) (2007) 163-190.
- [31] T. Gorschek, S. Fricker, K. Palm, and S. Kunsman, A Lightweight Innovation Process for Software-Intensive Product Development, *IEEE Softw.* **27** (1) (2010) 37-45.
- [32] D. Graham, Requirements and testing: seven missing-link myths, *IEEE Softw.* **19**(5) (2002) 15-17.
- [33] A.Griffin, PDMA research on new product development practices, *J. Product Innovation Management* **14**(6) (1997) 429-458.
- [34] K.Grimm, Software Technology in an Automotive Company – Major Challenges, in *Proc. 25th Int. Conf. on Software Engineering (ICSE'03)*, Portland, Oregon, USA, 2003.
- [35] A. Gunnesson, *Transitioning to Agility, Creating the 21st Century Enterprise*, (Addison-Wesley, Reading, MA, 1997).
- [36] S.H. Haeckel, Adaptive Enterprise Design: The Sense-and-Respond Model, *Planning Review* **23**(3) (1995) 6-42.
- [37] S.H. Haeckel, *Adaptive Enterprise: Creating and Leading Sense-and-Respond Organizations*, (Harvard Business School Press, Boston, MA, 1999).
- [38] Hansen, Software-defined features. The Hansen Report on Automotive Electronics, *A Business and Technology Newsletter* **18**(4) (2005).
- [39] M.T. Hansen, N. Nohria, and T. Tierney, What is your strategy for managing knowledge?, *Harvard Business Review* **77**(2) (1999) 106-116.
- [40] J. Hartley and J. Mortimer, *Simultaneous engineering: The management guide to successful implementation* (2nd ed.), (Industrial newsletter Ltd, Dunstable, 1991).
- [41] D. Hoch, W. Huhn, U. Naher, and A. Zielke, The race to master automotive systems development, McKinsey Company, Automotive and assembly sector business technology office, 2006.
- [42] M.N. Kennedy, K. Harmon and E. Minnock, Ready, Set, Dominate –implementing Toyota’s set based Learning for Developing Products ... and nobody can catch you, (Oaklea Press, Richmond, VA, 2008).
- [43] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques*, (John Wiley, New York, 1998).
- [44] 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) (2007) 317-330.

- [45] J.K. Liker, *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*, (McGraw-Hill, New York, 2004).
- [46] J.W. Meyer and B. Rowan, Institutional organizations: Formal structures as myth and ceremony, *American Journal of Sociology* **80** (1977) 340-363.
- [47] M.B. Miles and A.M. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook* (2nd ed.), (Sage Publications, Thousand Oaks, Calif., 1994).
- [48] J.M. Morgan and J.K. Liker, *The Toyota Product Development System: Integrating People, Process, and Technology*, (Productivity Press, New York, 2006).
- [49] I. Nonaka, A dynamic theory of organizational knowledge creation, *Organization Science* **5** (1994).
- [50] Object Management Group OMG, OMG UML 2.3, Infrastructure, 2010, <http://www.omg.org/spec/UML/2.3/Infrastructure/PDF/>.
- [51] J. Pernstål, A. Magazinovic, and P. Öhman, A case study of the interaction between development and manufacturing organizations with focus on software engineering in the automotive industry, in *Proc. 7th Conf. on Software Engineering Research and Practice in Sweden* (SERPS'07), 2007.
- [52] J. Pernstål, A. Magazinovic, and P. Öhman, P, A Multiple Case Study Investigating the Interaction between Manufacturing and Development Organizations in Automotive Software Engineering, in *2nd Int. Symp. on Empirical Software Engineering and Measurement* (ESEM), 2008.
- [53] 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) (1999) 172-179.
- [54] F. Pettersson, M. Ivarsson, T. Gorschek and P. Öhman, A practitioner's guide to light weight software process assessment and improvement planning, *The Journal of Systems and Software* **81**(6) (2008) 972-995.
- [55] M. Poppendieck and T. Poppendieck, *Lean software development: an agile toolkit*, (Addison-Wesley, Boston, 2003).
- [56] A. Pretschner, M. Broy, I.H. Kruger and T. Stauner, Software Engineering for Automotive Systems: A Roadmap, *Future of Software Engineering* (2007) 55-71.
- [57] B. Regnell, M. Host, J. Natt och Dag, and T. Hjelm, An Industrial Case Study on Distributed Prioritisation in Market-Driven Requirements Engineering for Packaged Software, *Requirements Engineering* **6**(1) (2001) 51-62.
- [58] E. Robers, Managing Invention and Innovation, *Research Technology Management* **50**(1) (2007) 35-54.
- [59] C. Robson, *Real World Research: A Resource for Social Scientists and Practitioners-Researchers* (2nd ed.), (Blackwell, 2002).
- [60] W.W. Royce, Managing the Development: Concepts and Techniques, in *Proc. Wescon*, 1970.
- [61] T.L. Saaty and L.G. Vargas, and L.G. Models, *Methods, Concepts & Applications of the Analytic Hierarchy Process*, (Kluwer Academic Publishers, Boston, MA, 2001).
- [62] J.W. Schreggenberger, The Further Development of Design Methodologies, An Overview of Descriptive Studies in Relation to a General Design Research Methodology, in *Designers: The Key to Successful Product Development*, eds. E. Frankenberg, P. Badke-Schaub and H. Birkhofer, (Springer, 1998), pp. 57-67.
- [63] D.K. Sobek, A.C. Ward, and J.K. Liker, Toyota's Principles of Set-Based Concurrent Engineering, *Sloan Management Review* **40**(2) (1999).
- [64] I. Sommerville, *Software Engineering* (Eighth ed.), (Addison-Wesley Publishers Ltd., 2007).
- [65] I. Sommerville and P Sawyer, *Requirements Engineering: A Good Practice Guide*, (John Wiley & Sons, Chichester, 1999).
- [66] R. Stevens, P Brook, K. Jackson, and S. Arnold, *Systems engineering—coping with complexity*, (Prentice-Hall, London, 1998).

- [67] R. Torkar and S. Mankefors, A survey on testing and reuse, in *Proc. IEEE Int. Conf. on Software: Science Technology and Engineering (SwSTE'03)*, Herzlia, Israel, 2003, pp. 164–173.
- [68] A. Vandavelde and R. Van Dierdonk, Managing the design-manufacturing interface, *Int. J. of Operations & Production Management* **23**(11) (2003) 1326-1348.
- [69] Volvo Car Corporation, (2010), <https://www.media.volvocars.com/global/enhanced/en-gb/Media/Preview.aspx?mediaid=41678>.
- [70] 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](http://www.volvotrucks.com/trucks/global/en-gb/newsmedia/quick_facts_about_volvo_trucks/pages/quick_facts_about_volvo_trucks.aspx).
- [71] C. Voss, N. Tsirikitis, and M. Frohlich, Case research in operation management, *International Journal of Operations & Production Management* **22** (2) (2002) 195-219.
- [72] M. Weber and J. Weisbrod, Requirements engineering in automotive development: experiences and challenges, *IEEE Softw.* **20** (2003) 16–24.
- [73] B. Wellman, Network Analysis: Some Basic Principles, *Social theory* **1** (1983).
- [74] S.C. Wheelwright and K.B. Clark, Accelerating the design-build-test cycle for effective product development, *International Marketing Review* **11**(1) (1994) 32-46.
- [75] J.P. Womack, D.T. Jones, and D. Roos, *The Machine that Changed the World*, (Rawson Associates, New York, 1990).
- [76] J. Womack and D. Jones, *Lean Thinking: Banish Waste and Create Wealth in your Corporation* Revised and updated edition, (Simon & Schuster Ltd., London, UK, 2003).
- [77] R. Yin, *Case study research: design and methods* 3rd edn, (SAGE Publications Inc., Thousand Oaks, 2003).
- [78] S. Zahran, *Software Process Improvement: Practical Guidelines for Business Success*, (Addison-Wesley, Reading, MA, 1998).