Impact of Embedded Software Design Decisions on the Product Life Cycle Process

Examensarbete, Innovation och produktrealisering

30 högskolepoäng, avancerad nivå

Produkt- och processutveckling

PREPARE

Alberto Palomeque

Presentationsdatum: 18 maj 2010
Uppdragsgivare: Volvo CE
Handledare Volvo CE: Joakim Fröberg
Handledare MDH: Stefan Cedergren, Markus Lindgren
Examinator: Magnus Wiktorsson
Abstract

Software design decisions were considered in this study, as the possibly principal factor for unplanned adjustments related to the embedded software handling, at production- and service processes. The study reveals an increase of requirement changes during the last phases in then software development projects execution, which forces late design decisions in order to fulfil the changed requirements. Consequently, the likelihood of risks for unexpected impacts on the subsequent processes will increase.

A research approach based on interviews and data from previous projects at Volvo CE was performed. The process methodology used at Volvo CE for software development was investigated from the project planning and control view and the project team member’s perspective.

A high amount of software-design decisions were encountered at the end of the software development process at Volvo CE, as a result of numerous requirement changes at the final phases of the projects execution. A gap was identified between how the process methodology specified the progression of activities for software development and the actual progression of the project activities in Volvo CE.

This study discusses problem areas in the software development process at Volvo CE from an embedded design decisions perspective. As future work, the study recommends three steps to find improvements to the process methodology: 1) Update the process based on standardized procedures for management of requirements changes, risk handling, and communication. 2) Further analysis and possible adaptations of the process model 3) Develop methods and/or tools for process quality assurance.

The management of the embedded software decisions appears to be a very complicated area, the conventional statements on the importance of the decisions in the earlier phases, at least, should be further discussed and investigated.
Acknowledgements

In writing this study report and during the year that this has proceeded, I was privileged to have the opportunity to meet and get the assistance from several people. My sincerely gratitude to my supervisors, Stefan Cedergren, Markus Lindgren and Joakim Fröberg at the MDH University. Their contributions in the structure of this study and their invaluable advices helped me to lead the work in the right direction.

I would also like to thank the managers and my colleagues at Volvo CE who unconditionally supported me in this work, special thanks to Jonas Disman and the project leaders that benevolently supplied the study with their experiences and essential information.

I would also like to acknowledge Mats Jackson, professor at the MDH University, who generated a great cooperation activity between the academy and the industry.

Finally, I send all my love and my thoughts to my family, Marisol, Camilla and Carlos Eduardo. Thank you for always supporting me in whatever I do.

Eskilstuna, November 2010
Alberto Palomeque
6. CONCLUSIONS .............................................................................................................32
   6.1. Requirement changes ..........................................................................................32
   6.2. Design Decisions ...............................................................................................32
   6.3. Risk Management ...............................................................................................33
   6.4. Communication Strategy ..................................................................................33

7. FUTURE WORK .........................................................................................................35
1. Introduction

1.1. Preface

This study primarily was intended to find out methods and tools in order to secure quality and project delivery time of the activities associated with release and introduction of embedded software components into the production- and service- processes at Volvo CE. Software design decisions, during the software development projects execution, were considered in this study as the principal factor for possible unplanned adjustments related to the embedded software handling at production- and service processes. However the results revealed that, prior to find methods and tools for quality assurance, complementary work must be performed aimed to accomplish the activities in the current projects execution model. To achieve effective improvements, Volvo CE first should identify improvements based on the current conditions. The work presented in this report describes a study of the present situation as a basis to continue the work in order to achieve the main target, i.e., find out methods and tools for quality assurance of the deliveries from the software development process to the subsequent production- and service- processes.

During the last decades, Embedded Software Development has grown up to be a critical part of the development and production processes within construction equipment. Demands for high performance, and smarter and more flexible functionality has delimited the mechanical solutions and forced the introduction of new methods based on electronics controlled by advanced embedded software applications. Leen and Heffernan (2002) argued that the mechanical and hydraulic applications safely and efficiently are being replaced by developing electronic systems. The expansion from a traditional mechanical- to a multifaceted- development process implies increased complexity in subsequent processes. The production- and service processes in construction equipment are affected by the increase use of electronics and software functionality (Axelsson, 2009). Grimm (2003) stated that competitive challenges are presented by a dramatic growth in system complexity, and are of major relevance for the growing number of software-based automotive innovations. Thus, the complexity that the software solutions imply, results in a complex design process as well, e.g., tardy software design decisions might require changes to the final solutions but it will not necessarily signify a restart of the projects. Instead, a changed final solution may have essential impacts in the subsequent processes.

Volvo CE develops electronic control systems for around 150 different machine models such as wheel loaders, haulers, excavators, and road machinery. The electronic system constitutes an increasingly important part of a modern construction machine’s functionality and the business around it. In order to meet the demands on business, safety, and development time, Volvo CE adapts the development method to a more product line oriented approach based wholly on an electronic system platform. This includes work on development process, architecture, tools, and system modelling.

1.2. The Research Objectives

The research work in this project is intended to study the interface between the software development process and the subsequent processes at Volvo CE. It is expected to deliver recommendation and best practices in order to prevent undesired impacts on production and aftermarket by identifying a possible improved design decision process for developing and delivering of embedded software to the production and service teams. Additionally, a second part of
this work was to identify where in the software development process software design decisions should be handled.

The objectives of this study were founded in cases that have been experienced at Volvo CE and also motivated on primarily literature studies, e.g., as Smith and Morrow (1999) expressed the importance to focus on design decisions during the development process: “Because of the tremendous impact of design decisions, any attempt to improve decision making will have great benefit for those firms who are successful”. The work has been primarily structured to identify design- and project-related decisions with high risk of unpredicted impacts in software development-, production-, and service-processes.

The study started in April 2009 and is scheduled to proceed until middle of March 2010.

1.3. Scope

The title Impact of Embedded Software Design Decisions on the Product Life Cycle Process denotes an in-depth investigation and research in the Embedded Software areas. The time available, for the study, forced a gradual delimitation of the scope and the academic depth of the work. Consequently the original targets and assignment were reshaped from time to time. The scope was defined originally to cover three basic processes at Volvo CE:

- Development
- Production
- Service

All of the basic processes above were delimited to handling of embedded software components. The work was divided in two phases: The first phase aimed to form the basis of the further activities and possible revision of the study targets and scope. In the second phase the study was focused in defining a theoretical framework and summarizing results, conclusion and recommendations. One of the results from the first phase ended up in a reduced scope. The new revised scope covered merely the development- and production processes related to handling of embedded software components.

1.4. A main project with four sub projects

This project entitled Impact of Embedded Software Design Decisions on the Product Life Cycle Process is part of a main project initiated within the electronic department in Volvo CE, under the name Platform of State of the Art Development Methods for Electronic Systems, in cooperation with Mälardalen University (MDH).

Platform of State of the Art Development Methods for Electronic Systems is the name of a main project consisting of four sub projects, each focusing on one pre-study. The sub projects have some interaction but can be completed and run separately if needed. The sub project; Impact of Embedded Software Design Decisions on the Product Life Cycle Process is defined as the Sub Project 3 (SP3).

The purpose of this main project is to produce guidance to the electronics development department on how to adapt practices, methods and tools, in order to meet challenges in shortened development time, development precision, and quality. Currently, the Volvo CE electronic development organization is restructured to focus on a developing machine electronic system based on a common electronic platform of electronic base services, hardware, and applications. This project shall aid in investigating the introduction of methods, tools, and architecture. In order to fulfil the purpose of the
main project; research work will be made to study the topics and produce input on how to accommodate the necessary change. The current electronic system and development methods will be studied in order to propose changes. This project and its sub projects, aim at producing results within the areas of requirements and system modelling, use of software development tools, system architecture, and development process.

1.5. Outline

The report is structured as follows:

- **Section 1: Introduction** introduces the present work. Additionally, a description of the interdependences between this study and mayor project is briefly presented at the end of this section.

- **Section 2: Research Method** presents a discussion about research methods- and methodology-, and a description of the research process utilized in this study.

- **Section 3: Frame of Reference** introduces the concepts and concerned areas related to the present work. As part of the frame of reference, a short description of the project directive is presented in this section.

- **Section 4: Data Collection & Analysis** provides information about methods, handling, and analysis of data.

- **Section 5: Threats to validity** discusses the validity threats of the study

- **Section 6: Conclusions** summarizes and discusses the conclusions of the present work

- **Section 7: Future Work** presents suggestions regarding future work
2. Research Method

2.1. Research Approach Theory

The methodology adopted to conduct a research, is currently named as Research Approach. How objective or subjective the methodology is, classifies the approach in different perspectives. Fagerström (2009) described three epistemological perspectives having different starting points for how scientific knowledge can be presented and justified:

- **Positivistic**
- **Hermeneutic**
- **Systematic**

These epistemological perspectives are also called paradigms. A paradigm is presented by Fagerström (2009) as: *Accepted practices applied in the scientific community*. The three paradigms are described as follows:

- **Positivistic perspective.** Scientific rationality, knowledge is empirically evaluated. Explanations can be seen as a cause - effect (1-1). Objective measurements represent the guideline. A rationalistic approach is dualistic, individual and reality can be interpreted isolated (Fagerström, 2009)

- **Hermeneutic perspective.** Man and world are related to each other by the personal experience of the world. Hermeneutics can be translated to the interpretation doctrine. Interpretation must be done in relation to its context. In an interpretive perspective, the individual and the reality are not separated (Fagerström, 2009)

- **Systematic perspective.** Systems theory is an answer to positivism, to be able to handle more complex problems. Systems theory refers to a system where a group of components (natural or artifacts) are interacting through relations. Explanation and understanding is reached by mapping of the final connections, which means looking for an effect by finding a purpose-oriented motivation (Fagerström, 2009)

2.1.1. Methodological approach

How the methodology of a research approach, is applied and elaborates new knowledge, is presented and discussed by Arbnor and Bjerke (1997) as three methodological views:

- **The analytical view.** The one and only purpose of the analytical view is to explain reality, that reality is filled with facts and independent of individual perceivers

- **The systems view.** In the systems view, the purpose is either to explain or to understand. The reality looks complete of facts, parts are not isolated from each other, and the parts are quite structured as relative wholes, called systems

- **The actors view.** The actors view is interested in understanding the complete aspects and parts, explanations are not of interest. Wholes and parts are ambiguous and are constantly reinterpreted. In the actors view, reality is treated as a social construction

2.1.2. Research Methods

In social science the research are categorized in two main groups, Johnsson (2008) describe these approaches as follows:
- Qualitative Research. Generates hypothesis based on descriptions or interpretations
- Quantitative Research explains and predicts, confirms and validates, or applies to test theory

However, an important clarification about method, research and data must be made, qualitative or quantitative methods do not exist. It is the data, which is qualitative or quantitative (Fagerström, 2009). Moreover, research methods are also defined as Deductive- and Inductive- approaches, Trochim (2006) describes these approaches as follows:

- **Deductive approach.** Informally called “top-down” approach, the reasoning proceeds from the more general to the more specific in order to confirm original theories
- **Inductive approach.** Informally called "bottom up" approach, the reasoning proceeds in the opposite way, from specific observations to generalizations and theories

2.1.3. Planning the study

The work in this study was aligned to follow a systematic perspective from the research approach point of view. In the same way, a systematic view was most appropriate to be considered as research methodology, Figure 1.

**Figure 1:** The system perspective and the system view were considered more suitable for this study

The discussion regarding how to follow a research method as either qualitative or quantitative resulted in a complicated issue. It became more suitable to discuss in terms of deductive or inductive approach; thereafter, the available data could be handled as qualitative or quantitative. The study was planned to start with explorative interviews and thereafter, by observations conducts to build up final assumptions, this method is presented above as inductive approach. Figure 2 visualize the structure of the inductive approach according to Trochim (2006).

**Inductive Approach**

**Figure 2:** The inductive approach structure (Trochim, 2006)
The operative part of the study was planned to collect data. Collection of data can be performed using different methods:

- Surveys
- Interviews
- Case Studies
- Controlled Experiments
- Literature studies
- Literature reviews
- Observations
- Other

Two methods mostly were used for data collection in this study:

- Interviews
- Literature Studies
2.2. Research Process

This sub project is related to how people, working in the software development process areas, manage and adapts needed process activities in order to fulfil the process requirements. A system approach, based on interviews and supported with data from previous projects, was considered the best method to perform this work. An investigation on the process methodology used at Volvo CE, would complete the whole picture of the software development process from two views; the project planning and control view and the project team members perspective. Figure 3 illustrates how the study was conducted.

![Figure 3: The research process structure based on two phases](image)

PAP = Project Assurance Plan
2.2.1. The first phase

The research work started with a review of the literature within the interface of software development and production process. The review included several sources, i.e., books, dissertations and journal articles. Software Development Process and Decision Making are well investigated and described areas. However, the interface into the production process is less researched and therefore less explained. Based on the findings from the literature review, an exploratory interview study was needed in order to better understand the scope of the research content and also to develop a basic framework.

2.2.2. The Second phase

The second phase was intended to end up in recommendations and best practice for improving the embedded software development process from a project execution perspective. A theoretical framework was considered important to develop. Three factors were found fundamental to design the framework:

- Analysis of the Project Assurance Plan (PAP) document
- Review literature within software development processes
- Sum up and conclusion about the result from the PAP analysis and the literature review

The three factors were integrated with the results from the first phase, which resulted in the final theoretical framework.
3. Frame of Reference

3.1. Software Development

Regardless the support model for project execution within the software development process, the handling of design decisions during the project is hard to manage. Only about 25% of the software development projects are successfully completed according to the original requirements, in time and within the budget (Wallin, Ekdahl et al., 2002). A poor relation between decision making and software engineering will end up in inefficient and low qualitative software development (Karhunen, Jantti et al., 2005). Seen from that perspective, most of the software development projects have difficulties to manage the decision making and consequently the impacts will affect the subsequent processes. Keene and Keene (1992) stated the earliest design decisions as decisions that have a high predisposition to end up in significant impacts on the operational and support aspects of the software product. In general, decision making is suggested to be handled early in the projects; strategic decisions, design decisions, etc made early in the projects, are considered the most important in the lifecycle (Kiper and Feather, 2005). Thus, software design decisions, made in the software development process, have an interrelation with the production- and aftermarket processes, which will affect the interface between the three processes. Fröberg, Åkerholm et al. (2007) pointed out the importance of a well handled design decision management in terms of communication and preparation through the three processes; “The desired quality is therefore achieved if the project is executed as planned. Many of these strategies are, once chosen, not negotiable. If a component were to fail in complying to a decided diagnostic signalling scheme for instance, the world wide service organization may not be able to handle this component, which certainly would prevent the vehicle from being produced at all”. However, a significant part of the literature handle software development related to conventional software, while embedded software and its related areas appear less discussed (Shehabuddeen, Hunt et al., 2002). There are considerable differences between conventional- and embedded software development. The main differences are related to how the software interacts with hardware. While the embedded software interact with specialized hardware in which timing has a crucial role, the conventional software uses operating systems as the interaction domain (Lee, 2005). Another important aspect is that software development implies software production and software delivery. It means, for conventional software, the complete product process while, for embedded software, the delivery of a component to the final production-and after market-processes. That aspect is a fundamental difference that should be considered when the embedded software is discussed from a software design decisions point of view. Though such differences should be reflected in the design decision management, the reviewed literature does not present sufficient discussion material about the impacts of embedded software design decisions in production- and after market processes.

Embedded software parts are in construction equipment industry defined as any other mechanical part but managed by a virtual logistic process, which consequently demands to divide the production process, related to the embedded software parts, in two main sub processes; component production and machine production (final assembly). The component production, like other standard software production, is implied in the development process, which inevitably leads to consider an interface to the machine production. Software design decisions during the design phase in software development have impacts on production (machine production), service and use of the machine. Some of these decisions might create unexpected or revised requirements regarding e.g. assembly processes, new
production equipment, changes in the service tools, education of service personnel, and constraints in use of the machine. The economical consequences could be substantial both in the production process, service activities and for the customer.

3.2. Product Development Models

According to the Merriam-Webster’s Collegiate Dictionary, a process is defined as “a series of actions or operations conducing to an end”. Further, Software Development Process is defined as “a structure imposed on the development of a software product” (Wikipedia, 2010). In order to structure and govern the actions and operations to conduct the development of a software product, several models have been proposed.

New Product Development (NPD) is a more general definition, related to the management of the disciplines involved in the development of new products Trott, 2008. Currently, NPD models are valid for software development from a design perspective. Paul Trott (2008) presented seven categories in which the NPD models can be classified:

- **Departmental-stage model.** A linear model of innovation in which each department is responsible for specific tasks
- **Activity-stage models and concurrent engineering.** Similar to the departmental-stage model with focus in the activities and using feedback loops for iteration of the activities
- **Cross-functional model (teams).** Emphasises communication as a central part of the process. Requires interdisciplinary teams which end up in organisational modifications
- **Decisions stage model.** The progress of the projects are based on a series of decisions
- **Conversion-process model.** A high amount of inputs into a “black box” are converted into an output
- **Response models.** Focuses in the response of the organisation or individuals to new projects and ideas, using a behaviourist approach to analyse changes
- **Network models.** Suggest the NPP view as a knowledge-accumulation process, requiring inputs from an extensive diversity of sources

3.2.1. The Stage-Gate Process

A combination of the Activity-Stage Models and Concurrent Engineering together with the Decision Stage Model was presented by Robert Cooper under the name of Stage-Gate process (Trott, 2008). The Stage-Gate process that has become a widely used process is defined by Cooper (2008) as:

“The Stage-Gate process, in its simplest format, consists of a series of stages, where the project team undertakes the work, obtains the needed information, and does the subsequent data integration and analysis, followed by gates, where go/no-go decisions are made to continue to invest in the project”

Cooper (2008) presents the process structured by a series of stages composed by best-practices activities required to progress the projects to next gate, the characteristics of each stage are:

- Reduce the uncertainties and risks
- Incremental cost handling
- Parallel activities performed by teams
- Cross-Functionality, no department owns any one stage

The stages will end in a gate for prioritization and decisions. The gates are go/kill decision points, and consist of:

- Deliverables
- Criteria
- Outputs

### 3.2.2 The Global Development Process (GDP)

Planning and Control of development projects in Volvo CE are governed by Volvo’s Global Development Process (GDP), which is based in six main phases, Figure 4. A list of deliverables distributed through the GDP phases is handled and described in the Project Assurance Planning (PAP)- document. The deliverables in the PAP- document constitute a decision package that must be managed and approved before any phase starts in GDP. Hence, how these deliverables are handled and how well the GDP is follow during the development process should be reflected in how successful the project will be.

The Volvo GDP methodology can be categorized as a Stage-Gate Process containing a series of decisions based on the deliverables in PAP and designed to control the project through the development process. The Volvo CE’s Booklet defines and describes the GDP as follows:

The primary focus of the GDP process is the delivery of the right product with the right quality, at the right time, cost and risk level with features that meet or exceed customer expectations.
Phases

The Global Development Process is built and distributed in six phases. The phases start and end at Gates:

- **Pre-Study Phase.**
  This phase defines the scope of the project by balancing project targets, developing requirements and alternative solution concepts

- **Concept Study Phase.**
  This phase analyzes alternative concepts and selects one for development. Document and sign off the preliminary Project content

- **Detailed Development Phase.**
  This phase defines and approves the solutions to be implemented and the project’s delivery commitments from all areas. Freeze and sign off the Project content

- **Final Development Phase.**
  This phase defines the activities to build, verify, validate and refine the product solution. Refine Market, Aftermarket, Manufacturing and Assembly solutions

- **Industrialisation and Commercialisation Phase.**
  This phase defines how to install, prepare and verify the industrial system. Launch product and Aftermarket products. Finalise Product qualification verification and validation to approve Product Release

- **Follow-up Phase.**
  This phase defines the hand over product to line organisation, follow up project target fulfilment, summarise project experiences and close project

Project Assurance Plan (PAP)

The Project Assurance Plan (PAP), defined in the Volvo CE’s Booklet, is; documents, required activities, and required results for the project as developed by the Project Management Team. It is also used as documentation during the Project Review. The activities in the PAP are based on:

- Applicable activities from the GDP,
- The project time plans,
- The project defining documentation,
- Globally accepted process requirements and instructions.

3.3. **The Embedded Software Process (ESW) at Volvo CE**

At Volvo CE the Embedded Software Process is defined as:

The Embedded Software Process includes systems, tools, methods, processes, and forums supporting embedded software related operations within Product Development, Production and Aftermarket. The purpose of the process is to make new embedded software components (ESW parts) available for Production and Aftermarket, and to ensure the correct ESW parts at correct point of destination in end product, at correct point of time.

The ESW at Volvo CE is organized in three main processes: Product Portfolio Development (PPD) corresponding software development process, Order to Delivery (OtD) corresponding production
process, and the Delivery to Repurchasing (DtR) corresponding to the after market process. The three main processes are aimed to handle ESW-Parts from design and development to the after market operations.

ESW parts are defined in Volvo CE as the parts building Embedded Electronic Systems in construction equipment (Åkerholm, Fröberg et al., 2007). The following items are classified as an ESW part:

- Embedded Software Applications
- Embedded Software Parameters
- Electronic Control Units (ECU) – hardware

The handling of an ESW-Part starts, in ESW at Volvo CE, when the Software Development Process is complete to delivery a new ESW-Part, ready to be introduced in the Production- and After Market- processes. The ESW parts are then introduced in the systems by the ESW release procedures, from that time the ESW parts can be ordered for production and after market.

The OtD process is currently triggered by a custom order. The ESW systems will package the ESW parts per machine order and send it to the right factory for assembly. In the factories the ESW parts are handled and assembled into the machines with the ESW production tool, PROSIT+ (Production Systems Integrated Tool). PROSIT+ is a complete computer system, including server and clients, specifically developed for assembly of ESW parts in the Volvo factories. The assembly procedures include; download-, check-, test-, and calibration- operations. Besides the main production operations, the tool has functionality for follow up and report of every activity concerned the machines regarding ESW activities. Similar to PROSIT+, the after market operations are supported by the VCADS Pro tool (Volvo Computer Aided-Diagnostic System Professional). VCADS Pro is a PC based aftermarket tool with specific operations to support ESW service activities at the field. The basic operations in VCDAS Pro support functionality for; testing, calibration and downloading of ESW Parts in the machines.

3.4. Project Directives

The main directive of the study was to find out methods and tools in order to verify and validate possible results and impacts of software design decision in the subsequent processes at Volvo CE, Figure 5. Consequently, the study would consider two aspects:

- The time at any design decision arose
- Potential areas, in production- and service- processes, with risk of impacts
3.4.1. The first phase

In the first phase, the objectives of the *Impact of design decisions on the product life cycle process* project can be summarized as an assignment aimed to find improvements or methods for:

- Early identification of new or revised requirements/design
- Continuous verification and validation to predict possible adaptation needs in production and service processes

The deliverables were defined as:

- Identify design decisions that greatly affect following processes
- Define methods to identify new or revised requirements that will affect the final solution
- Define process improvements
- Propose needed actions

The project delimitations were set up to cover processes for:

- Design of ESW for construction equipment
- Machine production regarding ESW handling
- Service and support regarding ESW handling
Based on the assignment, deliverables and scope above, the research methods were dependent on a couple of assumptions that inevitably must be considered in order to define a direction of the work:

- Most of software design decisions should be made during the concept phase or at latest before the final development phase (refer to GDP description above)
- There are software design decisions based on new or changed requirements/needs introduced during the concept phase or at latest before the final development phase with potential impact in the production and services processes
- If the assumptions above are true, then, the production and service processes should be updated and prepared in time in order to produce and support the complete scope of the delivered software components from the development process

3.4.2. The second phase

The second phase intended to validate the assumptions in the first approach showed, that design decisions with potential consequence in production- and service- processes seem to be forced by specifications and requirements arising late in the process development. These facts would make any method or tool to find new or revised requirements in earlier phases completely ineffective due to the nearness to production start. This new revealed situation forced a new discussion about the targets and scope of the project and also new revised assumptions to be considered:

- There are possibly deviations from the established GDP-process in software development projects.
- A large part of the software design decisions based on new or changed requirements/needs seems to appear late in software development projects.
- The production and service processes have short time to be prepared for producing and supporting the complete scope of the delivered software components from the development process
4. Data Collection & Analysis

Data was collected in two steps: The first through explorative interviews and the second performing an analysis of the deliverables in the PAP’s document.

4.1. The first phase: Explorative interviews

Five engineers acting as project leaders participated in the explorative interview study and were asked to describe one chosen project each. The collection of data during the interviews was performed by recording that approximately took 1 hour each. The leaders chose projects that they found relevant in the topic: “software development and production interface”. The interviews did not contain any specific question, the project leaders got free space and time to describe the projects, how the project followed the process development model and how the project team experienced the process. However; they were requested to comment the projects in the sense of “what went wrong / well?” and suggest improvements or actions in order to enhance the process management related to project execution. Hence, the real challenge aroused when the interviews would be converted to a more detailed data source, the target was to have a matrix from which quantification could be done. The whole content from the interviews was broken down in two steps; first in comprehensive sections aimed to identify main problem areas and then in to small phrases which were entered in an Excel worksheet for further analysis. A verification of the complete transcriptions was also performed with the interviewed project leaders and the interviews content were updated accordingly.

4.1.1. Problem areas

The interviews were partitioned first in comprehensive sections in order to found main areas to relate overall problems. The partitioned sections were thereafter broken down in small phrases aimed to recognize detailed sub-areas. To recognize sources and targets of the pronounced problems, three categories were added to each sub-area; Orderer, Performer, and Receiver.

4.1.2. Process follow up

The next challenge was to find a method to identify where in the process any software-design decision was made. The purpose was to match the fragmented texts (phrases) against the GDP-phases, prior to analyze or decide if the phrase implies a design decision.

4.2. The second phase: Deliverables in the PAP document

The second phase in this research work was defined and based on the results of the explorative interviews. A workflow in the projects would be governed by the GDP and controlled through the deliverables from the PAP-document, which were identified as critical components of the software development process at Volvo CE. An analysis of these deliverables should show how the decision process is distributed along the GDP-phases and gates.

The PAP-document is made aimed to enter only specific data, like; Status, Initial week, Revised week, Resp. person, etc. An analysis, of the contents, demanded to filter, sort and also add attributes to the data. These conditions forced to extract the deliverables to other place for further processing. The data were extracted and the PAP-structure was kept in order to reflect the original document and a composite of all phases were created in an Excel worksheet. Five categories, representing critical areas in the development process, were added intended to act as attributes; Req (Requirement), Risk, Dev (Development), Ind (Industrialization) and PCtrl (Project Control). The objective was to add attributes to any deliverable, if the deliverable had an impact on the process areas (attributes). Table
1 presents how the deliverables were structured and matched against the five criteria classed as Impacted Process Areas.

### Deliverables classified against Impacted Process Areas

<table>
<thead>
<tr>
<th>Phase</th>
<th>Gate</th>
<th>Role</th>
<th>Cat.1</th>
<th>Cat.2</th>
<th>Deliverable</th>
<th>Impacted Process Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI</td>
<td>1</td>
<td>Proj. Manag.</td>
<td>General</td>
<td>Scheduling</td>
<td>Resources secured for Pre-Study</td>
<td>Req</td>
</tr>
</tbody>
</table>

Table 1: The deliverables were structured and matched against five criteria classed as Impacted Process Areas

The deliverables in PAP were quantified related to three basic areas:

- Requirements changes
- Risk Management
- Project Planning & Control

and two process areas:

- Development
- Industrialization

The basic area “Communication Strategy” could not be identified in a preliminary classification of the deliverables, only a few identified deliverables indicated a relation to “Communication Strategy”, therefore this area was excluded from the subsequent analysis.
4.3. Analysis of the first phase

4.3.1. Problem areas

Explorative interviews are often connected to an analysis from an inductive perspective (Wallén, 1996). Trochim (2006) describes inductive research approach as, “Inductive reasoning works, moving from specific observations to broader generalizations and theories” Based on this reasoning, the interviews were first partitioned in comprehensive sections and then into sentences. The amount of problem areas identified in the content of the sections, and later in the content of the sentences, was counted as hits per problem area. From the first break down of the interviews six main areas could be observed as either the source or the target, of a specific problem. Figure 6 below shows the amount of hits per respective identified main area. The six areas were identified and categorized as:

- Competence. Need of competence about a specific area
- Data / Information. Lack of or deficient Information and / or Data in a specific area
- Organization & Management. Unclear defined organization
- Project Management. Complications in project management
- Process. Weaknesses in the development process
- Set Up / Infrastructure. Insufficiencies in the set up and / or infrastructure

![Amount of Hits per Main Area](image)

Figure 6: Main Areas that could be extracted from the interviews after the first break down and the amount of hits per Main Area
Thus far, vague patterns can be observed as a result of the first break down. “Data / Information” and “Competence” possibly are connected with the communication strategies or procedures that must handle the relationship in the project team and also towards the stakeholders. “Process” (Project Execution) also denotes a high amount of hits.

Figure 7 illustrates how the subsequent break down helped to identify 22 problem areas. The main areas could now be distributed in more specific problem sub areas.

![Figure 7: The amount of recognized problems and sub areas, presented as hits per sub-area](image)

The analysis of the second break down (sentences) indicated five sub areas where high probability of drawbacks exists:

- Documentation about procedures routines and /or instructions
- Data handling
- Roles and Responsibilities
- Requirements and Specifications
- Project Control

### 4.3.2. Process follow up

Early in this study, the explorative interviews reveal that the original assumptions did not match the expectations and targets of the study. Two different pictures appear in the software development process applied to the GDP. The first one showing how the process should be accomplished to the rules and conditions dictate by the GDP through the deliverables in the PAP-document, refer to Figure 8 below, and the second one that presented how the process was performed from the engineering perspective.
The Design Decisions should be handled early in the project execution according the GDP-model. To open a gate and to start the next phase, the GDP demands primarily to accomplish the respective deliverables in the PAP, an evaluation of possible risks for the project and a mitigation plan for the risk with high impact and probability of occurrence. These basic demands are a part of the quality assurance activities in the GDP that should guarantee a high project performance and the success of the projects.

Figure 9 visualizes the view of GDP from an engineering perspective. This view, which is based on the explorative interviews, reveals another picture of GDP in the projects: Additional software design decisions, close to final phases in project execution, are required by the fact of a great part of the specifications and requirements changes appear late in the development process. One project leader comments this situation as follows:

“..the specs were not really in place... other complex problems of x-SW and y-SW and stuff there arose later, .. it was a good while later, .... it caused production problems for the XX machine model, which was first out and... I can not remember how much we had to postpone the production start..”
The engineers view, Figure 9, denotes and increasing of the requirements changes and specifications in the last phases of the projects.

The analysis of the results from the first phase, led to three substantial inferences:

- Revoke one of the original deliverables in the project: “Define methods to identify new or revised requirements that will affect the final solution”
- To identify design decisions, with high risk for impacts on the subsequent processes, close to start of production will not have any relevant importance since the production process is fully occupied with the launch activities at that point, and it is not accessible to perform any verification work
- The main target for this sub project: “To find improvements or methods for early identification of new or revised requirements/design and continuous verification and validation to predict possible adaptation”, becomes not applicable at the moment.

4.4. Analysis of the second phase

The second approach was partly focused in the literature study, GDP activities and particularly aimed to analyze the deliverables in the PAP- document.

4.4.1. The literature study

The review of the literature related to the software development process area was performed from a design decisions point of view, made and connected to the project execution. Four areas appear common and preponderant, and therefore considered as basic components in the software development process:

- **Requirements changes.** Hsieh (2006) brings up, in terms of structure and coordination, the complexity that involves the handling of the requirements: “Requirements engineering is a problem of coordination. Developing requirements for a large scale system is seldom a one-person job. Rather, it involves coordination among application domains; contradictory requirements and priorities; and interdependent modules, tasks, and people”. Consequently, the focus on the requirement management structure, coordination and well defined strategy is vital to achieve a high performance in the project execution

- **Risk Management.** Risk management is often presented as a critical factor to achieve high or low performance in the project, the risk probability of occurrence and its impacts are directly connected to the project performance. An effective risk management is an essential demand for project success (Tao, 2008), to understand the basic components; probability of occurrence and impact on project performance, will denote an efficient and appropriate risk management strategy (Boehm, 1991)

- **Communication Strategy.** In the latest years, the communication strategy areas have progressively been considered as one of the most important in the software development process. Together with formalization and trust, communication and information are fundamentals for handling of the interface between product development and production process (Lundin, 2008). Timing of communication is described as a persistent challenge
in software development teams, which right managed ensure the ability of the team to effectively coordinate their work (Ehrlich, Valetto et al., 2007)

- **Project Planning & Control.**

  Planning and Control of the projects can be discussed from several angles. That may be the reason why project management is pointed out as a problem, when the factors that influence project performance are analysed. However, the planning and control is together with risk handling, requirements changes, and the team, in terms of communication, the most distinguished factors for low performance projects (Wen-Ming Han and Huang, 2007)

4.4.2. The deliverables in PAP

The deliverables in the PAP document were categorized against the new categories, three basic- and two process- areas, refer to 4.2. The results were summed up for each area. Figure 10 presents the amount of hits per “Role” and “Phase” for the Requirements changes area and its distribution across the phases in GDP. The same analysis was performed for the other basic areas, aimed to visualize when during the project execution the largest amount of deliverables was handled, related to correspondent area.

![Figure 10: The distribution of amount deliverables in PAP, related to requirement changes](image-url)
Table 2 summarizes the deliverables handling regarding the largest amount of deliverables in the GDP, according the PAP document:

<table>
<thead>
<tr>
<th>Area</th>
<th>Phase w. largest value</th>
<th>% of Total Deliverables / Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement Handling</td>
<td>Development</td>
<td>22 %</td>
</tr>
<tr>
<td>Risk Management</td>
<td>Development</td>
<td>18 %</td>
</tr>
<tr>
<td>Communication</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Planning &amp; Control</td>
<td>Concept Study</td>
<td>14 %</td>
</tr>
</tbody>
</table>

This analysis of the PAP-document brought up new assumptions to be considered:

- There were no specific deliverables in the PAP-document that could be connected to late embedded software design decisions with high impact risk in the subsequent processes
- Critical areas for software development process are missed or vaguely considered in the PAP document

On the other hand, the results of the data collection and analysis, in the first phase of this study, indicate that the view of the engineers about the process ended up in the following assumption:

- Late design decisions with high impact risk in the subsequent processes might arise late in the software processes due to, either missed specified requirements or non applied risk management, to the software product concept.
The data collected in the interviews reveals an increase of requirement changes during the last phases in the project execution. A consequence of this trend forces the developers to take corresponding design decisions in order to fulfill the new incoming specifications and needs. An analysis of the amount of recognized requirement changes against the phases in GDP, Figure 11, indicates a similar trend of the amount of software design decisions distributed through the GDP phases.

![Figure 11: The relation of software-design decisions to the requirement changes distributed in the six phases of GDP, presented as a trend.](image)

Figure 11: The relation of software-design decisions to the requirement changes distributed in the six phases of GDP, presented as a trend. The graph a result of the analysis of the interviews with project leaders.
5. Threats to validity

Four threats to the validity are currently discussed:

- **Conclusion validity.** The relationship between the treatment and the outcome concerns this validity Wohlin, Runeson et al. (2000). The data must properly be collected and analysed (Sundmark, 2009).

- **Internal Validity.** No relation must be found between the treatment and the study outcome (Wohlin, Runeson et al., 2000), the presented conclusions are reliable (Johnsson, 2008), and any disturbing factor that affect the outcome must be discussed (Sundmark, 2009).

- **Construct Validity.** Concerns theory and observation and the relation between them (Wohlin, Runeson et al., 2000). The construction of the methods for procuring data and information, e.g., interviews, must be truly related to the problems that are studied (Johnsson, 2008).

- **External Validity.** Concerns the generalization of the results of the study outside the scope of the study (Wohlin, Runeson et al., 2000).

5.1. Conclusion validity

The study is mainly based on five explorative interviews and literature studies. The low amount of interviews are probably not representative of a complete picture of the development projects execution at Volvo CE, though a general assumption in the software development areas is, that most of the projects are similarly executed in terms of problem handling. In order to quantify the extracted data, the interviews were broken down into sentences in two steps; first into comprehensive sections and then into sentences. Certain limitations such as bias in the selection of the sentences may affect the evaluation and reliability of the data analysis in the study.

Two presentations for the electronic department at Volvo CE were performed. An interest contribution to this validity discussion is that the engineers still recognize the results of the analyses in the present projects and agreed with the conclusions of the study.

5.2. Internal validity

The implementation of the GDP model is still not completed in the software development departments. The work to establish the using of the GDP was started at the time of the execution of some projects that were described in the interviews. This situation is not considered in the study, which possibly has a more negative influence in the result of the analysis.

5.3. Construct validity

The study started without specified objectives regarding what to measure or what type of data would be collected and analysed. The connection, amount design decisions to the phases in GDP, was developed after the interviews revealed another picture of the project execution activities than expected from the GDP model.

5.4. External validity

This study is performed at Volvo CE, in the electronic development department and focused in embedded software development process. Even if the automotive- and construction equipment-
industries should have similar processes, it is not possible to generalize the results and conclusions of this study. More work must be done comparing several similar processes in order to claim generalisation of such studies.
6. Conclusions

The study was aimed to map and understand the relationship between the development- and the subsequent processes from a design decision point of view. To develop methods and tools to prevent impacts of software-design decisions on the production- and service processes, the study found out that work should be done beforehand.

The targets as stated in section 3.4 turned over to long-term targets. Instead, the work performed in this study was redirected to focus on the present activities connected to software design decisions during the projects execution. The conclusions of this study became to concern four basic areas in the software development process:

- Requirement Changes
- Design Decisions
- Risk Management
- Communication Strategy

6.1. Requirement changes

A gap was identified between how the GDP project model specified the progression of activities for software development and the actual performing progression of the project activities in Volvo CE.

At the start of this study, a common assumption was that most of the software-design decisions were handled in the early phases, particularly the concept- and development phases in GDP, of the projects execution. This thinking is founded on a general picture of a development process model, like a stage-gate model (Cooper, 2008), and also supported by a general statement found in the literature study, which is expressed by Kiper and Feather (2005) as, “in any software system development, the most important strategic decisions are, by definition, those made early in the lifecycle” (Kiper and Feather, 2005). The analyze of the PAP document establishes that the highest amount of deliverables, which can be related to the requirement- and risk management areas, is encountered in the concept- and development- phases. This information leads to corroborate the assumption, in which most of the software-design decisions is expected to be handled in the early phases of projects. Based on the results of the interviews a new view of the Volvo CE software-design projects progress is revealed. A high amount of design decisions is taken late in the process.

6.2. Design Decisions

The analysis of the interview data, refer to Figure 11, denoted an increase of the amount of requirements and specifications changes at the final phases of the projects, which also results in an increase of software-design decisions at that time of the projects. The gap, between theory and practice, is driven by requirement changes carried on continuously during the project execution and which increases at the final phases of the projects:

- A high amount (> 50 %) of software-design decisions were made at the end of the software development projects in Volvo CE, as a result of numerous requirement changes at the final phases of the projects execution
Late design decisions in the projects execution result in less time for verification and validation unless the start of production is delayed. This may result in increased risk for production disturbances as described in the section 4.4.2.

6.3. Risk Management

The analysis of the PAP document, in the second phase of this study, was focused on four basic areas within the software development process: requirements changes, risk management, communication and planning and control. The risk management area is considered, in the reviewed literature, as a crucial area for any product development project. In a stage-gate model, as GDP, the risk management is one of the principal components. The stages are designed to gather information to reduce key project risks and the information requirements thus define the purpose of each of the stages in the process (Cooper, 2008). Still, the implementation and handling of risk management appeared to be hard to manage; Fairley (1994) stated that one reason risk management is not practiced is that few guidelines are available that offer a practical, step-by-step approach to managing risk. In 2008, Cooper still agreed with that situation; “it seems that as quality control checkpoints, the gates are not very effective in too many companies and allow a lot of poor projects to proceed” (Cooper, 2008). The relationship between a well performed risk management and a careful gate handling seems to be the key for a successful project execution in high-performance projects using the stage-gate model.

At Volvo CE the results of the PAP analysis reveal that the deliverables, related to the product concept and connected to the risk management area, are expressed in the PAP document in general terms. One such example is the deliverable; “Risk analysis - Design related risks analysis performed (e.g. Design-FMEA, FTA etc.)”. As it seems, deliverables that are written in such general terms and missing clear references, invite to perform diffuse interpretations leading the project to be exposed for possible risks of unsuccessfully risk management. Since the projects often involve many development areas, it should be favourable to implement standardized procedures, methods, and tools for risk analysis and management.

- At Volvo CE the PAP deliverables related to the risk management for software design are not clearly structured, nor have specified procedures.

6.4. Communication Strategy

Communication strategy has been discussed during the last years as an essential area in software development process. Lindvall and Rus (2000) argue that the process should help by guiding people on what to do-on how to divide and coordinate the work-and by ensuring effective communication. Seen from this perspective, the analysis of the PAP indicates a lack of communication strategy from a deliverable perspective. Only 5 deliverables out of over 800 have a straight relation to the communication strategy area, as presented in the PAP document.

- Communication strategies are missing in the current PAP document.

It needs to be defined who needs what data and when it must be delivered. A communication strategy is complemented with an organization around the projects, which often is structured and based on functional groups. The found literature regarding the interface between the software development- and production processes and the results from this study at Volvo CE, indicate a need of interaction between functional groups early in the project execution and reinforced by a strong communication strategy. Integrated Problem Solving is a method recommended in the reviewed
literature in order to prepare a successful implementation of the upstream design. A well structured communication strategy is a vital component to achieve integrated problem solving. The upstream and downstream groups in the process are linked by the participation of the downstream engineers on the preliminary and ongoing dialog, but using the information and insight to get their work a flying start (Wheelwright and Clark, 1992).
7. Future Work

The original attempts and targets, in this study, should still be considered as a well intended approach to secure quality and project delivery time for embedded software components at Volvo CE. As the current status of the software development process does not allow an immediate developing of methods and/or tools for identification and the validation of software design decision on the subsequent process, a first step aimed to update the software development process to a more controlled process should be performed. The updated process should be founded on standardized procedures for management of:

- Requirements Changes
- Risk handling
- Communication and Team

Thereafter, a further analysis of the process model and possible adaptations can be performed and implemented. Thus, the need of methods and/or tools for quality assurance will easily be identified.

The embedded software development process appears to be more complicated than the literature study presents. The project organization in construction equipment, or other automotive industry, often has several levels of subprojects. From a stage-gate process point of view, the lower the level which the projects are executed within the earlier they must deliver, e.g., the embedded software development projects must deliver when the component projects (engines, transmissions, etc) start the component verification phase, as the component projects must deliver when the machine projects start the machine verification phase. This procedure will generate feedbacks from both the machine and the component projects to the embedded software development projects, which will most likely end up in new or revised requirements and consequently new or revised embedded software design decisions. The same reasoning can be done in the other way, thus, when the requirements and specification are fed into the embedded software development projects. Shortly, the embedded software development projects possibly have late specification- and requirement- inputs, and also late new- and/or revised- requirements from the verification and test phases. From that point of view, the management of the embedded software decisions appears to be a very complicated area, the conventional statements on the importance of the decisions in the earlier phases, at least, should be further discussed and investigated.
References


