CHALMERS
UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG

Recommendations for Introducing Model Based Systems Engineering

Master's thesis in Software Engineering

Oskar Selberg
Vidar Åsberg

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHEBURG
Gothenburg, Sweden 2017

Master's thesis 2017

Recommendations for Introducing Model Based Systems Engineering

Master's thesis in Software Engineering

OSKAR SELBERG
VIDAR ÅSBERG

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2017

Recommendations for Introducing Model Based Systems Engineering
OSKAR SELBERG
VIDAR ÅSBERG
oskar.selberg@gmail.com
vidar.asberg@gmail.com

Supervisor: Michel Chaudron, Dept. Computer Science and Engineering
Examiner: Eric Knauss, Dept. Computer Science and Engineering

Master's Thesis 2017
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2016

Recommendations for Introducing Model Based Systems Engineering
OSKAR SELBERG
VIDAR ÅSBERG
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Background:
Model Based Systems Engineering (MBSE) is widely used in large scale projects within the embedded systems domain with benefits being shared understanding and improved communication. The transition to MBSE is however not as easy as a flip of the switch. A common and major challenge is effective adoption among developers.

Aim:
The goal of this thesis is to create a framework which provides recommendations for how to introduce MBSE by finding and addressing areas of particular concern for system engineering companies.

Method:
In order to accomplish the goal, and answer the research questions, a literature review and a case study were conducted. The literature review consisted of two parts, a systematic literature review, and an additional literature review with a broader scope. The case study consisting of eleven interviews at two different Saab AB locations that have introduced an MBSE approach. The literature review and case study were then synthesized in order to achieve the goal.

Results:
A framework was created by synthesizing the literature review and case study. The framework provides recommendations for introducing MBSE into projects. The framework is also presented on a timeline, helping practitioners practically apply the recommendations.

Conclusions:
The created framework reduces risks discovered during the literature review and case study. A limitation of the framework is that it has not been assessed in an industrial project. It should in the future be tested in a real setting.

Keywords: MBSE, guideline, framework, introduction, case study

Acknowledgements

We would first like to thank to our supervisor, Michel Chaudron, for his guidance and expert opinion throughout the entire project. Furthermore we would like to thank Per Gustavsson for being our supervisor at Saab Surveillance in Gothenburg. Finally we would like to give our deepest gratitude to the employees at Saab Linköping, Järfälla and Gothenburg that volunteered for interviews, allowing us to obtain valuable experience to produce the final results.

Oskar Selberg & Vidar Åsberg, Gothenburg, June 2017

Contents

List of Figures

List of Tables

1 Introduction

Model Based Systems Engineering (MBSE) is widely used in the embedded systems domain (Liebel, Marko, Tichy, Leitner, & Hansson, 2016) and there are many benefits of using models, (Chaudron, Heijstek, & Nugroho, 2012) for example, state that models in large scale projects lead to shared understanding and improved communication. The transition to MBSE is, however, not as easy as a flip of a switch. A common and major challenge is effective adoption among developers; this is needed in order to realize the benefits of model based engineering (Liebel et al., 2016). Reports also imply that the cost is high for implementing MBSE into projects (Voirin, Bonnet, Normand, & Exertier, 2015; Andersson, Herzog, Johansson, & Johansson, 2010). The reason costs are high might be because the papers and books specifying the notations and use of the most prominent modeling languages UML & SysML (Object-Management-Group, 2015b, 2015a) are lacking in regards to how MBSE is practically introduced into different company contexts. (Herzog, Andersson, & Hallonquist, 2010) for instance integrate models by first analyzing the company and its desired use of models. They also state that in order to successfully introduce models, the focus needs to be on finding the key capabilities of modeling that improve the performance for the organization.

This thesis attempts to gather the main areas of concern when introducing MBSE, to reduce the risks associated with the introduction of modeling. Knowing what areas to provide extra attention on can ease the initial introduction of MBSE. The identified areas will be compiled into a framework which will act as a guideline and provide a recommendation for how the introduction process should be done in systems engineering companies.

1.1 Problem Statement

The department for Fighter Radar and Data Link at Saab AB in Gothenburg, hereafter referred to as Saab Gothenburg, has during the past two decades been developing highly advanced radar systems, which has grown large over the years. The system has now grown so complex that it has become difficult to traverse the extensive amount of textual documentation. This also makes it hard to get a good understanding of how the systems function and how they are actually implemented. A common problem for any large system that has been under development over a long period of time is that the original people that built the foundations may have retired or left the company, meaning that the original ideas and reasoning might have been lost. Since the scale has grown large, it is challenging for different teams to communicate with each other, maintain or modify the software and hardware. These challenges can be solved via the use of models.

Saab Gothenburg does not currently utilize models for its radar system, something they believe can provide great value to their projects and business; which is why they want to integrate models into their current way of working. Two other locations, Saab Järfälla and Linköping, have already introduced different modeling techniques. Both initially faced problems which from which lessons could be learned. During the prestudy for the model introduction at Gothenburg, we discovered that the lessons, setbacks, and pitfalls that Linköping and Järfälla encountered were about to be repeated by Saab Gothenburg. Therefore a framework containing recommendations for how the introduction of modeling into projects should be done had to be created; so the problems encountered at both Saab Järfälla and Linköping could be avoided in Saab Gothenburg's future projects and at other companies introducing modeling.

The primary intention of this report is to provide recommendations for Saab Gothenburg, on how they should handle and prepare for the introduction of models into their current way of working; drawing from the lessons learned at other Saab locations. The recommendations will be presented in the form of a framework, that will make the process easier to understand and implement.

1.2 Research Questions

The goal of the study is to create a framework which provides recommendations for introducing MBSE by finding areas of particular concern for system engineering companies.

To achieve the aforementioned goal the following research questions need to be answered:

- **RQ1: What risks are there when introducing MBSE?**
- **RQ2: Which aspects of introduction and use of modeling can be tailored?**
- **RQ3: How can a framework for introducing MBSE be created?**

1.3 Delimitations

The goal of the thesis is to create a framework which provides a general recommendation, therefore no advice on specific tools, modeling languages or processes will be provided. Problems related to the cost for the framework or implementation process will not be included since it is out of scope.

Research question 2 will present which aspects can be tailored, and only provide basic information about how they can be tailored.

For research question 3 a framework will be created, but due to the time constraint of 20 weeks, it will not be possible to test the framework in a real company setting, as the time to introduce MBSE is suspected to exceed those 20 weeks. As such this thesis will be unable to test the effectiveness of the proposed framework.

1.4 Thesis Outline

This thesis is structured in the following way: Chapter 2 presents the motivation and methodology utilized to accomplish the result and reports on validity threats. Chapter 3 presents the literature studies that was undertaken. Chapter 4 presents the case study and its results. Chapter 5 contains the synthesis of the literature review and case study, and presents the final framework. Chapter 6 contains the conclusion and future work that should be undertaken.

2

Method

This Chapter describes the overarching method for answering the research questions of the thesis, see Section 1.2. Two literature reviews were conducted, followed by a case study of the model introduction process at Saab Järfälla and Linköping. The reasoning behind each of the individual studies is presented in their respective chapter. The method for the systematic literature review (SLR) and additional literature review (ALR) is presented in Chapter 3 and the method for the case study is presented in Chapter 4.

2.1 Overarching Method

The information gathered during the literature review and case study was used to answer the presented research questions in Section 1.2. The gathered information was then used to achieve the thesis goal of creating a framework that provides recommendations for introducing MBSE. Figure 2.1 presents how the different parts in the report build upon each other to finally reach the goal, the numbering in the figure represent the order in which the steps were performed.

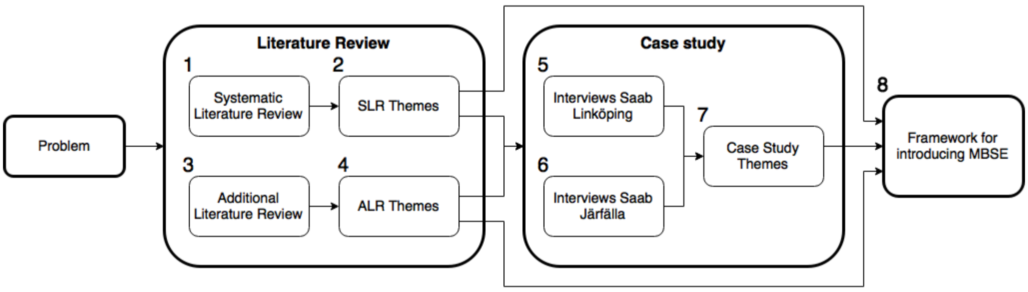

Figure 2.1: Overarching method for thesis

The literature provided the foundation for the framework. The systematic literature review presents its own research questions and answering them provided valuable information that helped to answer the main research questions, see Section 1.2. The combined scope the systematic and additional literature reviews had together was broad, finding many different themes from published literature related to modeling. The themes were used as a basis for research questions that were formulated for the case study and provided a foundation in literature for the resulting framework, see Figure 2.1.

The case study conducted at Saab Linköping and Järfälla provided insight into which risks that were common, what areas of the MBSE introduction process were crucial, and their lessons learned. The research questions presented in the case study combined with the answers gathered from the literature review answered all the main research questions. The information uncovered during the case study then fed directly into the framework, giving it a strong connection to the current practices used at Saab Järfälla and Linköping.

The data collected in the literature review and the case study was synthesized and applied in the creation of the framework. The framework will thus have both a strong connection to the current literature surrounding modeling and industrial practices related to the introduction of modeling. The framework created after the synthesis answers research question three, and finally achieves the thesis goal presented in Section 1.2

2.2 Literature Review Motivation

The literature review needs to be unbiased since the different modeling methods upon which the framework builds should be as generally applicable as possible, i.e; be useful for as many companies as possible. Therefore a systematic literature review (SLR), see Section 3.1, was conducted. (Keele, 2007a) explains that an SLR can be conducted in three different ways, although only one is applicable to this thesis: "*To summarize the existing evidence concerning a treatment or technology e.g. to summarize the empirical evidence of the benefits and limitations of a specific agile method*". The SLR aimed to provide information about common themes between different modeling methods, providing unbiased information about benefits, disadvantaged, and similarities. In addition to the SLR, an additional literature review (ALR) was conducted.

The ALR provided additional information about topics that were found during the SLR but was outside its scope. It focused on providing additional information that could help support the theories later gathered in the case study. The ALR included relevant papers provided by domain experts that might have a large impact on why to utilize methods and procedures for specific problems.

The method utilized when conducting the literature review can be found in Chapter 3, the SLR method is described in Section 3.1 and the ALR method is described in Section 3.2

2.3 Case Study Motivation

Since Saab Linköping and Järfälla already have introduced MBSE into some of their projects, they have discovered different lessons, pitfalls, and problems that should be documented and analyzed. By analyzing the different risks that are associated with the introduction of modeling they can be further reduced in future projects, and directly be adapted into the framework.

The primary data collection for the case study was semi-structured qualitative interviews, gathering both positive and negative aspects of the modeling methods utilized at both Saab Järfälla and Linköping. The information gathered during the case study was utilized during the creation of the framework.

The information gathered by the case study will provide the framework with methods, techniques, and risks that are associated with the introduction of MBSE, i.e; the case study will provide the industry context into the framework.

The method utilized during the case study can be found in Chapter 4, the reason the research method was chosen to be a case study is presented at Section 4.3.

2.4 Framework Creation

The result from both the literature review and case study were synthesized during this stage, see Figure 2.1. The synthesis did not follow a preexisting method, rather a combining the answers the main research questions presented in Section 1.2, into a single body of recommendations. The method, techniques, and risks associated with the introduction of modeling gathered during the case study are the main inputs the framework was built on. The literature review was also referenced, ensuring each presented argument also had a foundation in literature.

The reason for not using any preexisting method was because the authors needed to be able to make logical considerations about why, when and where a specific solution was required. Each recommendation is presented in a chronological order and is tightly tied to the literature review.

2.5 Validity Threats

This section discusses the four validity threats categorized by (Runeson & Höst, 2009): Construct Validity, Internal Validity, External Validity and Reliability.

2.5.1 Construct Validity

A risk to construct validity is that the interpretation of terms might differ between the researchers and interviewees, which could, for example, result in the interviewee answering a way not originally anticipated by the researchers. A mitigation strategy for this is data triangulation, using multiple data sources (Runeson & Höst, 2009). Data triangulation was used during the interviews by conducting them at two different location, Saab Järfälla and Linköping. The interviews were semi-structured and utilized probing. Using probes the interviewee could be asked follow-up questions if the researchers felt that something was unclear. To ensure a stronger validity the interview protocol was also reviewed by a contact at Saab and a researcher from Chalmers University of Technology. Another mitigation strategy was observer triangulation, as described by (Runeson & Höst, 2009), by having two observers always be present during all the interviews instead of only one. Observer triangulation was also utilized during the literature review, where both researchers read all the papers in parallel, confirming the findings of each other.

2.5.2 Internal Validity

Internal validity is the threat of unknown factors affecting the study, e.g. interviewees answering insincerely due to the fear of repercussions. This threat was mitigated by providing anonymity for the interviewees, and by triangulating the data by selecting interviewees with different roles within the company. The selection was, however, not done by the researchers, but rather by managers at each respective location. This could be a threat to validity as the interviewees might be biased towards MBSE since the managers knew the topic in advance.

(Haselton, Nettle, & Murray, 2005) explains cognitive bias as something the authors might be affected by when, for example, choosing to accept or reject papers for the literature review. This was to an extent mitigated by having the two authors make their own judgment individually on all the papers and then comparing the papers and accepting or rejecting them in a predetermined way see Table 3.2.

Once the papers were selected and data was being collected, there is a risk of detection bias. Two researchers might not find the same data points when reading the same paper, this was mitigated by having all the data points found by each of the researchers validated by the other, see Data Collection in Section 3.1.1

2.5.3 External Validity

External validity is the threat to generalizability and relevance of the study. When searching the online databases for the SLR, the search was confined to titles, this was done to reduce to amount of hits on the search term. This could, however, result in relevant papers being left out which in turn can be a threat to the external validity. This was mitigated to an extent by performing an additional literature review, where more information surrounding the domain and research topic are analyzed.

The case study only examines one company but by looking at two different locations within that company the validity of the findings increases. The sample size of the number of interviewees and companies can always be larger, but unfortunately, it was difficult due to the time constraint in the setting of being a Master Thesis.

The framework aims to be as generally applicable as possible by not explaining the exact details for each a step and allowing for the framework to be used as a guideline more than a strict procedure.

The framework does not explain the exact details for when a step is considered done. It does not either present concrete things that should always be done, this is done in order for the framework to be as generally applicable as possible.

2.5.4 Reliability

Reliability threat is the threat of the study not being able to be reproduced by other researchers. The mitigation strategy utilized is having a well-formulated methodology for the literature review and case study, which should ensure that the execution of the study can be replicated.

Since the research almost entirely consists of qualitative data, the bias imposed by the researchers might be a threat to the reliability of the data. A few things were done in order to mitigate researchers collecting data differently and drawing different conclusions in their analysis. The first was the cross reading of SLR and ALR papers, see Section 3.1.1, while the second was that during the case study, two researchers performed the transcribing and coding of the interviews and then reviewed the work done by the other.

3

Literature Review

A literature review was conducted to gain domain knowledge about practices and modeling methods. The information gathered from the papers will be the foundation on which the framework and recommendations are based upon.

The literature review consists of two parts, *Systematic Literature Review* (SLR), and *Additional Literature Review* (ALR). Section 3.1, describes the steps taken when planning, conducting and reporting the SLR. Section 3.2 goes on to present the findings from the ALR. The systematic literature review will be performed to gather data on how modeling is currently implemented in industry. The SLR should provide a scientific background, strengthening the conclusions that can be drawn. The ALR focuses on papers provided by a domain expert that could be relevant to the study.

3.1 Systematic Literature Review

The SLR performed in this study is based on the guideline provided in (Keele, 2007b) but is slightly tweaked, building on the method used in (Hebig & Bendraou, 2014) by accommodating additional findings. The SLR does not only focus on answering the research questions but also on discovering interesting pieces of information in the domain.

Conducting an SLR ensures that the results are more unbiased, and enables other researchers and practitioners to replicate the results. It will ensure that the final result to have a strong foundation in literature, with no bias towards specific methods. (Keele, 2007b) described an SLR as the following:

> "*[An SLR is] a form of secondary study that uses a well-defined methodology to identify, analyze and interpret all available evidence related to a specific research question in a way that is unbiased and (to a degree) repeatable*" (Keele, 2007b)

3.1.1 Planning

The planning is done in order for other researchers to be able to reproduce the findings, and come to the same conclusions; the thesis findings will because of this have a stronger scientific basis. The planning stage for the SLR is therefore written in such a way that the results are reproducible.

Background
The produced framework needs to be generally applicable and accepted within both the research and practitioner community, which is why it needs to be built on an unbiased per-

ception of how companies currently utilize models and modeling methods. The unbiased information about benefits and drawbacks of modeling methods will be the foundation the thesis builds upon. Information about how the industry is currently using modeling, and the practical benefits of using them need to be uncovered.

Research Questions for the SLR

The research questions aim to provide a foundation that the main research questions presented in Section 1.2 build upon, therefore, the research questions do not directly answer the main research questions but rather provide context.

The following research questions will be answered through the literature review:

- **SLRRQ1: Are there prevalent similarities between different modeling methods?**
- **SLRRQ2: How are models currently used in the embedded systems domain?**
  - **What are important factors when choosing an abstraction level?**
- **SLRRQ3: What are the risks when using models?**

SLRRQ1 and SLRRQ2 helped answer RQ2 *Which aspects of introduction and use of modeling can be tailored?* by providing current practices from literature. SLRRQ3 answered RQ1 *What risks are there when introducing MBSE?*.

Search Process

All searches were performed in February 2017. A keyword search was applied to three online sources; IEEE Xplore, ACM Digital Library and Google Scholar. In all searches, the keywords were matched solely on the title of the papers. The results produced using the search string should ideally include information about different modeling techniques utilized in current industry and answer how abstraction levels are chosen. The search query will be made up of the following parts:

- UML: A common modeling language
- SysML: A common modeling language
- MARTE: Abbreviation for *Modeling and Analysis of Real-Time and Embedded systems*
- Modelling: British spelling of modeling
- Modeling: Relevant search term
- Method: Relevant search term
- Methodology: Relevant search term

The search query will be applied as following:

$$A = UML, B = SysML, C = MARTE,$$
$$D = Modelling, E = Modeling, F = Method, G = Methodology$$

$$searchQuery = (A \vee B \vee C) \wedge (D \vee E) \wedge (F \vee G)$$

Selection Criteria

Every paper produced by the search query will not be relevant, this is why a procedure for selecting and rejecting papers must be defined. The procedure that will define if a paper

Table 3.1: Inclusion (I) and exclusion (E) criteria for the initial SLR selection procedure

No.		Criterion
1	I	Published article, conferences journal
2	I	Written in English
3	I	Describes a company context/setting
4	I	Discusses models
5	E	Duplicates between sources
6	E	Not accessible for Chalmers students
7	E	Published before year 2000

is included in the first result produced by the search query can be seen in Table 3.1. The procedure determines which papers are included and which are excluded with the criteria based on the SLR research questions.

Primary study selection process
In order to reduce the number of irrelevant papers, a primary study selection process will be used. Two researchers will examine the results of the searches and place their initial judgment of the relevance for each paper, based solely on the paper's title. This scoring is done by marking each paper with; *Accept*, *Candidate* or *Reject* and compiling the results in a list. Once the two researchers have created their own separate lists they are compared and combined through the formula shown in Table 3.2, resulting in a single list.

Table 3.2: Formula for combining the individual judgments

Marking 1	Marking 2	Result
Accept	Accept	Accept
Accept	Candidate	Accept
Accept	Reject	Candidate
Candidate	Candidate	Candidate
Candidate	Reject	Reject
Reject	Reject	Reject

Once the list is combined all papers are either marked with; Accept, Reject, or Candidate. If a paper is marked as a Candidate after the final list combination, a domain expert will give his opinions if the Candidate paper is relevant. If the paper was deemed relevant by the expert it will be included in the study. An attempt will be made to access all accepted papers.

In total 30 papers were marked as candidates in the final combined list, see Table 3.3. The final result from applying the search query to the aforementioned sources, and applying the primary study selection process can be found in Table 3.4.

Table 3.3: Number of candidate papers turned into accepted after domain-expert opinion

Marking	No.total	No. Accepted	No. Rejected
Candidate	30	7	23

Finally, the remaining papers that are marked *Accepted* will be thoroughly read, if a paper then is deemed to be out of scope, irrelevant or unfit for further scientific research it will

Table 3.4: SearchQuery Results

Source	No. Found	No. Excluded	Included
Google Scholar	126	118	8
IEEE Xplore Digital Library	35	24	11
ACM Digital Library	17	14	3
Total	178	156	22

be presented in a table, with a clearly stated reason for exclusion, see Table 3.5.

Data collection
The data extracted from each paper will contain the most relevant information that either answer the SLR research questions or could be of interest. The papers are read in their entirely which means that related information that does not directly answer the research questions can be encountered. Therefore the collected data from each paper will be structured in a table, so that all information has full traceability back to the source, see Table 3.6.

The selected papers from the three aforementioned sources will be read by two researchers. First, each researcher marks down their preliminary findings in the texts, secondly, the researchers swap papers with each other in order to validate the findings, and potentially find data that the other researcher missed.

Data Analysis
The data will be synthesized using a form of thematic synthesis drawing influences from the method discussed in (Cruzes & Dyba, 2011), where reoccurring themes are analyzed and noted. The data will not be coded in any particular way, rather the data found to be interesting is extracted from the paper. The higher-order themes that will be used are the SLR research questions that the SLR aim to answer.

The themes will be explained under each of the research questions. The analysis aims to answer the SLR research questions and provide a foundation on which the recommended framework is developed, see Section 5.

3.1.2 Conducting

The next part in the SLR is to conduct the literature review, apply the search string, exclude and include papers, and finally, read and gather important information. Table 3.4 shows how many relevant documents were found after the *title relevance analysis* was performed.

After the relevance of each title is checked for all papers, the abstract and conclusion for each selected paper were read to give further evidence that the paper was relevant. Some papers were excluded in this process, a list of these papers and the reason for exclusion can be found in Table 3.5.

Table 3.5: The excluded papers with reason for rejection stated

Title	Exclusion reason
Modeling Method for Information Model of Fault Tree Diagnosis based on UML (Suochang et al., 2010)	The abstract and conclusion are unclear, and almost no information can be gathered from them. The paper also have a focus on fault tolerance which is out of scope and too loosely related to Validation & Verification.
A Systems Engineering Methodology for Analyzing Systems of Systems Using the Systems Modeling Language (SysML) (Osmundson & Huynh, 2006)	although this paper might seem relevant at first glance the focus is on executable models and the tools for achieving such a model, a tool for building such models are also presented. This paper therefore is out of scope.
Aligning SysML with the B Method to provide V&V for Systems Engineering (Bousse et al., 2012)	This paper focus on integrating the B Method (Mathematical representation of system states) with SysML, while having a too small investigation of the modeling, and is therefor deemed not relevant
The COMPLEX methodology for UML/MARTE Modeling and design space exploration of embedded systems (Herrera et al., 2014)	This paper seems to assume that models are already in place, based on those a new methodology is presented. It discusses executable models which was deemed out of scope.
Environment Modeling with UML/-MARTE to Support Black-Box System Testing for Real-Time Embedded Systems: Methodology and Industrial Case Studies (Iqbal et al., 2010)	The paper have a main focus on building a model for a specific environment, and trying to create a black-box system that can be automatically tested. This report is deemed out of scope.
Modeling Method of Military Aircraft Support Process Based on SysML (Li et al., 2011)	The abstract and conclusion are written in such a way that no relevant information can be extracted or gathered from them, the report therefor was deemed not relevant.
Modeling Method of SysML-based Reliability Block Diagram (Liu et al., 2013)	The abstract and conclusion are written in such a way that no relevant information can be extracted or gathered from them, the report therefore was deemed not relevant.
A Failure Time Series Prediction Method based on UML Model (Xin et al., 2015)	Although interesting, the paper has almost no focus on the modeling process, but rather focus on how time to failure can be predicted. The process is made through modeling, but is deemed out of scope.
Model-Based Methodology for Implementing MARTE in Embedded Real-Time Software (Zaki & Jawawi, 2011)	The abstract and conclusion are written in such a way that no relevant information can be extracted or gathered from them, the report therefor was deemed not relevant.
Efficient System Modeling of Complex Real-Time Industrial Networks Using the ACCORD/UML Methodology (Gérard et al., 2001)	This paper is old, written in 2001, and aims to introduce a SysML like system for UML, it is therefore deemed not relevant.

Table 3.5: The excluded papers with reason for rejection stated

Title	**Exclusion reason**
Modelling a Software Architecture for Robots Control using UML and COMET Architectural Design Method (Ortiz Zaragoza et al., 2002)	The paper does not focus on models, main focus is the use of ADA language. It is deemed as not relevant.
Study of Product Modeling Method Based on UML (Zhu et al., 2010)	The paper seem to be a study, but in the conclusion the authors claim they have created a method, which does not seem to be presented in the paper. The paper were deemed unfit for further scientific research.

After the final elimination made after reading and judging based on the abstract and conclusion, each remaining paper was read in its entirety and the most relevant information was extracted. The extracted information from each paper can be found in Table 3.6.

Table 3.6: The included papers from Accepted papers

1 - Title	**A Methodology for UML Models V&V** (Baruzzo & Comini, 2008)
Summary	This paper has a large focus on tools and the way they can be utilized to achieve better: consistency, correctness, quality and long term maintainability for models. The authors propose a model which allows a UML-based development environment to integrate V&V. They believe this method of collaborate tools will lead to increased productivity and overall quality. The paper also mentions a problem with the current modeling process, referred to as the “state explosion problem”, this means that if a model grows large the number of states grow with it, making it hard to validate. It is also explained that in order to test the functionality for a model some code also has to be integrated into the model. The paper also argues that it is important to choose the right level of abstraction since a model should only communicate the relevant information for a user-specific goal.

2 - Title	**A Methodology for Multidisciplinary Modeling of Industrial Control Systems using UML** (Estévez et al., 2007)

Summary	This paper proposes a modeling methodology with models consisting of three related views and the relationship between them. Each view describes a domain and uses an UML profile. The UML profile is created by grouping additional labels to basic UML elements (classes, packages, e.t.c.) making them stereotypes to be used as elements of its view. The three domains/views presented in the paper are; Control Engineering (Components, Connectors, Channels and Ports), Electronic electrical engineering (hardware components such as; bus segments, nodes, modules and boards) and Software engineering (elements from IEC 61131-3; Configuration, Resource and Task). According to the authors it is possible to model the complete system using the profiles and subset of UML diagrams (class, sequence and use case). The order of modelling should be; first Functional Specification, secondly Hardware Architecture and lastly Software Architecture.
3 - Title	**Using DSM and MDM Methodologies to Analyze Structural SysML Models** (Maisenbacher et al., 2013)
Summary	The paper explore the use of graphical and matrix representation for a system, saying that graphical system representation (e.g. UML and SysML) increase the traceability and make it easier to visualize the system, while *Matrix-based* representations is said to increase the system understandability. The authors propose a method for extracting Matrix representations for the system through an automated process. The big problem with the approach is discovered to be that the transformation from SysML to Matrix is only unidirectional, and that the Matrix itself cannot capture the models to a full extent. The authors also argue that other researchers have found that graphical models provide a better understanding for port-specifications and interdisciplinary connections for a system.
4 - Title	**A SysML-based methodology for manufacturing machinery modeling and design** (Bassi et al., 2011)

Summary	The paper presents its modeling process as a guideline rather than a set of formal rules as they can sometimes turn out to be a hindrance rather than an asset. The paper also makes a difference between a modeling procedure and a design process, stating that a procedure is generally independent from the process where as the process must be compatible with both the procedure and other constraints coming from the company or institute context. Software and System models are fundamentally different since software can be executed whereas system models need to be simulated in order to be tested. SysML is suitable since it is compatible with most UML and can incorporate additional parts, however not everything can be included in the SysML standard due to the possible differences between projects. SysML models in this paper is used as a high level abstraction for the system under design. In the presented process the first step is to lay down core concepts and a general structure, after that additional detail is progressively added until a global executable model is present. Having the information organized as hierarchy of models at different abstraction levels solves the problem of low-level information being absent if only using a high abstraction level. In this paper the purposed hierarchy consists of three levels of model abstraction; Detail level, Global level and High-level. These abstraction levels are linked by defining modules as loosely coupled subsystems and having them reference all models that are connected to that subsystem regardless of abstraction level. The paper states that an implementation issue that cause some reason for concern is the fact that for every refinement step the diagrams become larger and more inter-correlated resulting in cluttered diagrams and tedious validation.

5 - Title	**An approach for modeling architectural design rules in UML and its application to embedded software** (Mattsson et al., 2012)
Summary	According to this paper a substantial part of the architecture consists of design rules and constraints in the form of text documents. The enforcement of these are left to manual reviews which can be tedious and there is also the risk of mistakes. The proposed solution is to specify the rules using UML at a high abstraction level but formal enough for the enforcement to be automated. The stakeholders benefiting from this are both the architects, who do not need to do as much manual reviewing, and the developers, who do not have to wait for their design decisions to be accepted. Future research was suggested to be investigating how easily architects, developers and other stakeholders can learn the approach and apply it into their way of working.

6 - Title	**A Safety Modeling Method Based on SysML** (Zhou et al., 2014)

Summary	The main focus of this paper is to provide a way in which a model can be utilized in order to provide better V&V, and for less human errors to be made. The authors claim that there is a problem with hazard propagation although a system if there aren't acquit safety requirements in place, meaning one problem might lead to several new. They explain that it is difficult for designers to find potential hazards which are caused by interactions between different components within a system. They also claim that a complex system can be identified by three different characteristics; increase in complexity and coupling, new types of hazards, decreasing tolerance for new hazards due to design. The authors then propose a method for developing SysML models, by which the probability for introducing new hazards are kept to a minimum. They also separate the system to three different abstraction levels; system-, subsystem and component level, all of which can be constructed with the same method.

7 - Title	**A UML based methodology to ease the modeling of a set of related systems** (Alhalabi et al., 2008)
Summary	The paper introduces a methodology for creating and maintaining UML models, which makes it easier to model a set of related systems. The methodology in essence divide the creators of the model into three groups; UML-Expert, Architect and Developers. The UML-Expert is essentially the person that have a broad knowledge of how UML works. The architect is the person that has a holistic view of the domain, and thus know how to model the system. The UML-Expert and architect could even be the same person, while the developers are the people that code the system. The UML-Expert and Architect build a high abstraction-level model, by which enough detail for the developers to understand what the architecture is supposed to do. The developers then add or derive a lower abstraction-level diagram (e.g. component diagram) from the architects model. In this way the developers always maintain their own models, which means that the overall model is maintained. The authors claims the only downside for this method is that it require a basic knowledge of modeling and UML, they also claim that the approach is suitable for complex systems.

8 - Title	**On the impact of UML analysis models on source-code comprehensibility and modifiability** (Scanniello et al., 2014)
Summary	Often times software maintainers only have models produced in the requirements analysis process, *UML analysis models*, at their disposal. This paper tries to answer how useful these early models are later on by performing a controlled experiment with three replications for a total of 86 participants in the form och students and practitioners. The paper concluded that UML models created in the requirements elicitation and analysis "seemed to not improve the comprehensibility or modifiability of the source code". It was speculated that the lack of improvement found may be due to fact that the models used did not provide information on the actual implementation. According to the authors it is possible that the analysis models needed to be combined with design models to be useful.

9 - Title	**Object oriented methodology based on UML for urban traffic system modeling** (Chabrol & Sarramia, 2000)
Summary	A holistic view is required for a successful design planning and management of something as complex as an UTS. The paper proposes an object-oriented modeling methodology to be used that creates a communicative model comprising of both a structural and functional models at different levels of detail. The paper goes on to call the structural model for knowledge model and the functional for action model. The knowledge model comes from analyzing and specifying the domain into three subsystems; logical (moving entities), physical (characteristics of the roads) and decision (functioning rules for traffic). The knowledge model is then translated into the action model, which is a mathematical formalism or programming/simulation language. What diagrams are used depend on what subsystem is described but they originally come from UML. The decision subsystem need the most diagrams due its dynamic nature, it uses Class, Use Case, Object, Collaboration, Sequence, State chart, Activity and Deployment diagrams. Both the logical and physical subsystems are described using Class and Object diagrams. Additionally the logical subsystem also employs Activity diagrams and the physical subsystem uses Deployment diagrams.
10 -Title	**UML modelling in rigorous design methodology for discrete controllers** (Łabiak et al., 2012)
Summary	This paper presents a UML modelling process with three initial steps. First a Object-oriented analysis is conducted, this identifying material objects in real world and describing them in the form of UML class and object diagrams. The second step is to perform a functional analysis which is is done by creating UML use case diagrams. The last step, Behaviour analysis, result in activity diagrams, sequential diagrams and collaboration diagram. Based on the results from the initial steps the paper present two approaches for further formal description and validation of the system; State machine based or Petri net based. The State machine based approach uses UML state machine diagrams and are easy to develop, this approach does however still requires Petri nets to be used for validation algorithms to be used. The Petri net based approach is basically creating Petri nets directly without doing the extra step of adding any additional UML diagrams as documentation. The paper concludes that if the modelling method is followed it would make it so that client requirements changes can be efficiently applied.

3.1.3 Data Analysis

This Section aim to answer the research questions connected to the SLR. A thematic approach was selected when answering the questions since the questions do not have a clear single answer. It will start off by answering research question one, then two, a.s.o. Each question will reference themes found in the reports and focus on the information disclosed in Table 3.6.

1. **Are there prevalent similarities between different modeling methods?**

The majority of the analyzed papers [paper 1,2,4,5,6,7,9,10] discuss that the modeling process needs to be divided into either different abstraction-levels or sub-models. The reason being that a single abstraction-level cannot capture the full extent of the system, and that some low-level information might get lost [paper 4]. The method in several papers [paper 2,4,7,9,10] argue that the modeling should be done at three different abstraction-levels; high, middle and low. The right level of abstraction should only communicate the relevant information for a user-specific goal.

Each paper proposes a method for improving some aspects within their respective area or domain. This could mean that it does not exist a method that fit all needs, and that it could not exist since different methods provide benefits to only specific areas, e.g. [paper 3] discuss that their method increases the understandability and traceability for the system, while no recorded benefits are mentioned regarding other areas. [paper 6] on the other hand say that their method increases the V&V for a system, while not mentioning other areas.

All different methods require some form of understanding of modeling, as mentioned in [paper 7], where they say that the method requires a basic understanding of modeling. This is generally true for processes or methods that involve models to improve upon or solve problems within a specific area. Although different methods require different levels of understanding, some might require basic understanding by an expert, while others might require several people to have expert knowledge.

From [paper 2,9,10] three general steps were found to be taken when modeling system architecture. The first step is to specify and model the "real world" parts in the context. The second step is specifying what the system should be able to do. The last step is to model the systems software. From this parallels can be drawn to how abstraction of the design goes from a high to low abstraction level. The first step could be seen as specifying the domain, by just specifying the broad strokes. Going on to the next step, more detail is needed in order to solve the problems at hand. Lastly, modeling the systems software is at the lowest abstraction level. Just how low the lowest abstraction level used is varies depending on if the models should be executable, or used for exact implementation or not, i.e, depending on the model's purpose.

2. **How are models currently used in the embedded systems domain?**

Different problems have different solutions, those solutions are realized with models. Each utilized model or modeling method tries to solve a specific problem. Not all encompassing answer could be determined from the systematic literature review, but some indications could be extracted.

[paper 1] explain the “state explosion problem”, a problem present in the current modeling process. The problem is that if a model grows large the possible states the model could contain grows rapidly, in the end it becomes nearly impossible to validate. Similarly [paper 4] state that this is reason for major concern that the more detail is added the more cluttered and more difficult to validate the models become. The proposed solution is a modified UML-model that utilize design patterns to check if the model is using them and that no anti-patterns are used. In other words, the problem is closely connected to that the complexity of the system has grown unmanageably large, and models that improve the validation process need to be introduced.

[paper 3] explain that large and complex systems often utilize Matrices and graph-based

representation, since they both offer unique benefits. They describe that graph-based representation (UML, SysML, e.t.c.) offer "*increased traceability for users and an intuitive way of visualizing information*"[paper3]. Matrix-based representation is, on the other hand, explained to offer "*better analyzing criterions to achieve further system understanding. Mathematical analysis tools can be used in the matrices to solve certain problems*"[paper 3]. Choosing when to implement matrix-based representation is highly contextual, and therefore the problem domain is required to be fully analyzed.

[paper 4] state that since projects are different there can be no complete modeling language that specifies the models use down to the last detail. Models have to be specialized and adapted to each setting depending on what the desired goal is. [paper 4] has a goal connected to software validation and creates executable models for that purpose.

Models can be used for many different purposes, in [paper 5] for example, the design rules for a system are specified using UML-models. This provides the benefit of enabling automatic validation of the created design.

During the development of a system [paper 6] express the need for better V&V, because different development hazard can propagate through an entire system unnoticed. The solution to the problem is an adequate set of safety requirements mixed with graph-diagrams, namely; requirement, structure, and behavior diagrams. The authors choose to automate the process by using tools, because this would decrease the amount of human-made errors. As discussed in research question 1, the authors also divided the system into three different abstraction models, which better reflects how hazards propagate through different system-levels. The solution, just as the aforementioned methods, is focused on a specific area within the domain.

According to [paper 8] *UML analysis models* are often the only models available for the systems maintainers. These analysis models are use cases and use case diagrams, class diagrams that abstract objects from the problem domain and sequence diagrams that model the behavior of both users and system. The UML analysis models were found to not actually add to the comprehensibility or modifiability of the source code, while they at the same time are the most common models.

[paper 4,7,9,10] all utilized models at multiple abstraction levels as there are different benefits at each level. This means that it is not as simple as choosing an abstraction level for the design. [paper 9] states that a high-level holistic view is necessary for a successful design planning and management of a system but [paper 4] reminds us that low-level information is absent if only high-level models are produced. There is a trade-off [paper 4], every refinement step results in more cluttered models making the validation process more tedious. With multiple levels, it can be difficult to see how the levels are translated and connected. [paper 4] solves the traceability issue by using something the paper calls modules for grouping models that described related functionality regardless of abstraction level.

Each abstraction level has its purpose, high abstraction should have enough detail so that the developers can understand what the system does [paper 7]. The medium level describes how things are connected, while low details the implementation. Models can be separated in other ways than abstraction levels [paper 9] breaks down its system into three subsystems based on functionality. Then depending on which subsystem is modeled, different models are used. This is because not every model is applicable or useful for every subsystem.

It was found that models are used in different ways and for different purposes. Although the main use is for specifying the design and being able to validate the system before implementation.

3. **What are the risks when using models?**

When using models new problems may occur that are directly related to the constructed model and the development method. If the chosen method does not have the know risks in consideration, problems may later occur. During the SLR three major risks connected to modeling were commonly mentioned; maintenance, verification and validation, understanding.

Papers [paper 1,3,5,8] aim to solve problems that are somewhat connected to maintaining models, during or after a project. Even although graphical representation is something that helps understandability and traceability [paper 3], it can become difficult to maintain the model to a sufficient standard. Maintaining the models used during the project is, therefore, something that might need to be incorporated into the work process. That the models are maintained is important for the entire process, and if it falls behind affect all other aspects of the project.

The verification and validation, i.e that the model is correct and fulfill its intended purpose, is something that grows harder the more complex and large the system is [paper 1,3,5,6]. The models need to be so precise that it is possible to verify that it satisfies the requirements, but also easy to maintain and validate. A risk when the system grows large and complex is that it becomes nearly impossible to validate and much harder to maintain. It is also important to recognize that modeling at a lower abstraction level requires more diagrams and details, leading to a more tedious validation than only using high-level models [paper 4]. If the process contains inadequate V&V for the model's different hazards can propagate through an entire system [paper 6] and/or make the system impossible to maintain and verify. To decrease the V&V risk associated with modeling it is important to have processes and procedures that actively try to reduce the associated risk.

Each method requires experience ranging from basic to advanced understanding of modeling, to be able to be implemented and utilized to its full effect. Sometimes methods just require a single person to have the required experience [paper 5], while at other times all the developers require at least a basic understanding [paper 7]. This quickly becomes a problem related to cost and might be something smaller companies struggle with. Just as the aforementioned risks this is something that needs to be considered in the planning process.

It is important to consider the development process in which the models are going to be implemented in, as failing to do so will most likely result in an ineffective use of models. From [paper 4] it is found that the procedure and process go hand in hand it does, however, make a difference between a modeling procedure and a design process, stating that a procedure is generally independent of the process whereas the process must be compatible with both the procedure and other constraints coming from the company or institute context.

3.2 Additional Literature Review

The ALR was carried out in order to gather more information by expanding the scope of the literature review. It was believed that the SLR, while productive, might have missed relevant information surrounding the field being studied. The papers for the ALR were gathered in parallel to the SLR, so when interesting details or comments were referencing sources outside the scope of the SLR. Additionally, papers came in the form of suggestions from a domain expert. The papers found or suggested were kept but not read until after the SLR was complete as to not influence the analysis of the SLR as that would affect its validity. With the SLR completed there is also a better understanding of in which areas the literature review is lacking thus informing where to put the focus in the ALR. The findings from the ALR were synthesized into themes that connect to each other.

This Section discusses the themes found during the ALR and how they relate and affect each other. Section 3.2.1 describes the first theme; tailoring and factors to consider that affect how models should be used and how development processes relate. In essence, models fall into either one of two categories; Descriptive or Prescriptive, what they are and their different purposes is described in Section 3.2.2. Relating to the research question of what abstraction level is chosen, Section 3.2.3 presents findings on how to effectively use the correct abstraction level. Finally, Section 3.2.4 delves deeper into the benefits by using specific diagrams and methods.

3.2.1 Modeling and Process Tailoring

Several different papers mention that because each company is different, the modeling process needs to be customized in order to effectively benefit from the use of Model Driven Engineering (MDE) as stated by (Clarke & O'Connor, 2012):

(Clarke & O'Connor, 2012) argue that context is the most important factor to consider during the problem specification. This is also identified by (Coleman & O'Connor, 2008) who state that "*contextual issues... [are among] the main inputs to the tailoring process*". (MacCormack & Verganti, 2003) argue that software development and process managers need to "*evaluate a wide range of contextual factors before deciding on the most appropriate process to adopt for any given project*", this view was also shared by (Subramanian, Klein, Jiang, & Chan, 2009).

(Kalus & Kuhrmann, 2013) has compiled a list of 49 different tailoring criteria that could be relevant in a software development project. For each criterion, the paper presents exemplary actions for dealing with problems or an increased focus on that area. In total 20 different actions are presented, all of the 49 presented criteria are mapped to one or more of these actions

In short (Clarke & O'Connor, 2012) describes all the situational/contextual factors affecting the Software Process while (Hebig, Schmied, & Weisemöller, 2016) tries describes what to consider depending on factors are deemed important. (Heldal et al., 2016) also explains that the consumer of a model, the consumer could be a model-to-model transformation, needs to be properly analyzed so what is produced is useful.

(Whittle, Hutchinson, & Rouncefield, 2014) state that the increase in productivity from using code generation can range from -28% to +800%, this could be because of a wide range of things. Bad implementation could be one of them.

A problem companies face when introducing MDE for a project (Hebig et al., 2016), is failing to recognize the interrelationship between MDE and the process, and it could lead to scenarios where what should work in theory needs to be changed in order to work practically. Drivers that force either the process or use of models change are a high additional manual effort, uncertainty in responsibilities, arising optimization potentials. Mismatches in expected and actual quality of automated code, scalability, and skill required also typically result in some type of evolution to make the process functional. The topic of code generation will be further discussed in Section 3.2.4.

3.2.2 Prescriptive and Descriptive Modeling

Some papers discuss the intended purpose for the model, if it is either prescriptive or descriptive (Heldal et al., 2016)(Pelliccione et al., 2016). Prescriptive models or architecture is the "*as-conceived or as-intended architecture*" (Pelliccione et al., 2016), models that guide the development process this type of models is often called design models (Heldal et al., 2016). Descriptive models on the other hand describes the "*as-implemented or as-realized architecture*", models that document the existing system (Heldal et al., 2016). Both of these model types have different benefits and it is common that both are used in the same project.

(Heldal et al., 2016) analyzed how the industry is currently using prescriptive models and found that the main benefactor and driver for that model type is development. After development however prescriptive models tend to degrade in importance and relevance. Prescriptive models are typically at a low level of abstraction as to be detailed enough for implementation from its design as such models used for code-generation are always prescriptive.

(Heldal et al., 2016) also further analyzed the perceived benefits for descriptive modeling, and concluded that the industry mainly uses descriptive modeling for documentation and increased communication between teams.

(Clarke & O'Connor, 2012) discuss the need for a context analysis to be made before the models are implemented, which can be seen in (Heldal et al., 2016) that states that the abstraction level for the model have is heavily influenced by the intended purpose of the model; "*The presumed consumer would be new employees whose primary purpose would be to get a coarse-grained understanding of the system and its architectural principles*" (Heldal et al., 2016).

3.2.3 Model Abstraction-level

As mentioned in Section 3.2.2 companies use both descriptive and prescriptive models, i.e. models at both high and low abstraction levels, how the models are organized and related can be displayed different ways.

Production systems can for example be divided into three different abstraction levels; "*structural, behavioral and a control part*" as explained by (Schönherr & Rose, 2010). The theme of three different abstraction levels also occurs in (Pelliccione et al., 2016), although the abstraction levels are instead named; "*continuous integration and deployment, ecosystems and transparency and cars as systems of systems*". Another method for creating the different abstraction models is by viewing the system as a black-box at the highest level, where all internal system information is hidden and only the I/O is known and shown(Lane & Bohn, 2013). Once the black-box is expanded into a white-box, the inner workings are exposed, although the new white-box can contain new black-boxes, so that only the important information is shown. Opening those boxes would, in turn, result in an even lower abstraction level or the same level of abstraction but with other functionality described.

The abstraction level chosen for a specific model depends on the intended purpose for the model, in (Heldal et al., 2016) for example, a high abstraction level was chosen by a company because for them it was important that the system was understandable by new developers.

Another important aspect discussed by (Schönherr & Rose, 2010) is that the models need to scalable, either by separating the domain and instance model or by preparing design patterns.

Importantly (Heldal et al., 2016) explains that "*The consumer and intent strongly influence the abstraction level as well as the concrete syntax used*" this can also be seen by the uncovered information described in Section 3.2.1.

3.2.4 Modeling Benefits and Disadvantages

(Whittle et al., 2014) discuss that the real benefit is usually not code generation through MDE, but rather the support it provides in the form of documentation; "*It turns out that the main advantages are in the support that MDE provides in documenting a good software architecture*" (Whittle et al., 2014). The documentation advantage is mentioned by Forward and Lethbridge (Forward & Lethbridge, 2002) that state that "*Documentation is an important tool for communication and should always serve a purpose*", they also further mention that "*Learning to cope with the fact that documentation is almost always out-dated and inconsistent, we can then appreciate and utilize it as a tool of communication*".

(Whittle et al., 2014) also uncovered that some architects feel forced by their managers to create goldplated (i.e. unnecessary complex) models since "*[...] their managers don't understand that a simple model is better; rather, their managers perceive simple models as not properly thought out*".

(Schönherr & Rose, 2010) points out that in order to achieve satisfactory code generation from models, a tool that fits the purpose needs to be uncovered or created. It is also explained that there at the time did not exist any suitable modeling tools to support satisfactory code generation from models; "*A problem of using SysML for production systems is that no suitable modeling tools are available*" (Schönherr & Rose, 2010).

(Lane & Bohn, 2013) describes that Model-Driven Software Development (MDSD) provide a holistic overview of the system, this relates to (Heldal et al., 2016) that states; "*Large and complex systems need some high-level descriptive model to obtain an overview of the system*".

Both (Heldal et al., 2016) and (Whittle et al., 2014) explain that the use of Domain Specific Languages (DSLs) are currently popular in the industry, furthermore (Whittle et al., 2014) discusses that it allows the company at hand to customize the modeling practices that are implemented to fit their specific needs. (Whittle et al., 2014) even says: "*Whatever the context [...] modeling languages requires significant customization before the languages can be applied in practice*".

3.3 Literature Review Conclusion

The main goal of the literature review was to provide a foundation for the framework to be built upon. The literature review consisted of an SLR and ALR which can be utilized in both the final framework, connecting it to literature, while also allowing the interview questions in the case study to have a closer relation to literature.

Some of the finding during the literature review will be included in the case study interview questions, such as: models leading to better communication, models having either a prescriptive or descriptive approach, way of working being closely tied to the model processes and methods a.s.o. The different interview questions and associated probes can be found in Appendix A.

4

Case Study

4.1 Background

Saab Aeronautics in Linköping has been using modeling for both code and document generation in projects for a number of years and Saab Surveillance in Järfälla has recently introduced modeling into their way of working by using their model to generate documentation. Linköping, Järfälla, and Gothenburg all work on different parts of the same product. This is partly why Saab Gothenburg wants to look into how the other two have introduced modeling. Saab Järfälla encountered setbacks and problems during the introduction but now most of those have since been solved. The problems could be the same problems Saab Gothenburg might potentially face during their own introduction. This case study aims to identify what caused the problems and how the setbacks that Saab Järfälla faced can be avoided.

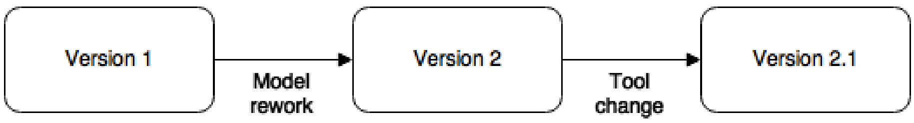

Figure 4.1: Changes to the modeling at Saab Järfälla

Figure 4.1 presents an overview depicting the different stages of Järfälla's model in the project and the transitions between them. Saab Järfälla has had two major shifts during their model introduction. In Version 1 the model was considered too complicated which is why a complete rework of the modeling method and the model was required arriving at Version 2. Järfälla changed their modeling method and the new method is less complicated and the model's purpose is narrowed down. The second shift was a tool change which was necessary in order to do version management of the model; the current model can be considered version 2.1. Both of the changes had large implications on the project as a whole.

4.2 Objective

The goal with this case study is to investigate which different aspects of the introduction and use of modeling that Saab Järfälla and Linköping found problematic, and to report what can be done in order to avoid them. To achieve the aforementioned goal the following research questions need to be answered:

- **CSRQ1: What problems occurred during the introduction of modeling**

in Saab Järfälla and Linköping?

- **CSRQ2: What was done in order to solve the problems related to the introduction of models in Saab Linköping and Järfälla?**
- **CSRQ3: What are lessons learned of using models at Saab Linköping and Järfälla?**

The case study's research questions relate to the main research questions of the thesis in different ways. CSRQ1 answers RQ1 *What risks are there when introducing MBSE?*. CSQR2 and CSQR3 together help answering both RQ2 *Which aspects of introduction and use of modeling can be tailored?* and RQ3 *How can a framework for introducing MBSE be created?*.

4.3 Method

Using a case study allows for a more flexible approach when analyzing data as explained by (Runeson & Höst, 2009), they argue that a case study is exploratory, the primary data set is qualitative, while the design remains flexible. This study is conducted to gain additional insights in order to answer the main research questions, see Section 1.2, the aforementioned reason corresponds closely to what (Runeson & Höst, 2009) mention in their report: "Exploratory—finding out what is happening, seeking new insights and generating ideas and hypotheses for new research". Examining the Table 4.1, it can be concluded that a case study is well suited for reporting the findings.

Table 4.1: Overview of research methodology characteristics. Note: Reprint from (Runeson & Höst, 2009)

Methodology	**Primary objective**	**Primary data**	**Design**
Survey	Descriptive	Quantitative	Fixed
Case study	Exploratory	Qualitative	Flexible
Experiment	Explanatory	Quantitative	Fixed
Action Research	Improving	Qualitative	Flexible

(Runeson & Höst, 2009) describes qualitative data as words, descriptions, pictures, diagrams etc. When analyzing data (Runeson & Höst, 2009) also explain that qualitative data should be analyzed by categorizing and sorting. The method that was used for the data synthesis was thematic analysis, (Cruzes, Dybå, Runeson, & Höst, 2015) describes thematic analysis as:

> "*A method for identifying, analyzing, and reporting patterns (themes) within data. It organizes and describes the data set in rich detail and interprets various aspects of the research topic. It can be used within different theoretical frameworks, and it can be an essentialist or realist method that reports experience, meanings, and the reality of participants. It can also be a constructionist method, which examines the ways in which events, realities, meanings, experience, and other aspects affect the range of discourses*"

The interview data was organized using the method described by (Saldaña, 2015), coding the data and thereafter finding themes within those codes, this is discussed further in Section 4.3.3.

Table 4.2: Number of interviewees per location

Location	Interviewees
Linköping	4
Järfälla	7
Total	11

Table 4.3: The ID, location, current title and number of years of modeling experience of the interviewees

ID	Location	Title	Modeling experience
J-AR1	Järfälla	Architect & Modeling method	20+ years
J-SE1	Järfälla	Responsible for methods & processes	5 years
J-SE2	Järfälla	System engineer & architect	3 years
J-AR2	Järfälla	Systems engineering leader	5 years
J-VM1	Järfälla	Verification Manager	4 years
J-SM1	Järfälla	Systems Design Manager	8 years
J-PM1	Järfälla	Project Manager	-
L-SE1	Linköping	Systems engineering Management	10+ years
L-FD1	Linköping	Function developer / CSCI Manager	7 years
L-FD2	Linköping	Function developer	5 years
L-FD3	Linköping	Function developer	2 years

4.3.1 Case study design

The interviews were held at two different locations, Saab Linköping and Saab Järfälla. The interviews focused on the introduction of modeling, the problems encountered and what was done to address those problems. Nine employees people were interview across the two different location. Saab Järfälla provided information that had a higher chance of answering the research questions, this because of how recent their introduction of modeling was. Saab Linköping had been using models for years and could provide a more long term understanding of how modeling worked for them.

Seven interviews were conducted at Saab Järfälla, see Table 4.2, all interviewees are employees that was or are highly involved in the model introduction, the personnel that could provide the most relevant information where interviewed, see Table 4.3 for the interviewees roles and experience. Two interviews, see Table 4.2, were held in Linköping where one of the interviews was with a highly regarded modeling expert within Saab and the other was a group interview with people who worked with models on a daily basis.

4.3.2 Data collection

The interviews were semi-structured as described by (Louise Barriball & While, 1994). Utilizing a semi-structured interview method is especially well suited for an exploratory study, since it allows for probing for additional information, or clarification of answers (Louise Barriball & While, 1994). Secondly (Louise Barriball & While, 1994) mentions that in a world that constantly changes it is difficult to schedule standardized meetings.

The interview questions, see appendix A, were designed as to avoid biases the interviewers might have that would affect the answers. If the interviewee mentioned something interesting the interviewer could probe for additional information, either with a improvised probe or from the predefined list, see appendix A.

The interview structure is based on (Bryman, Bell, & Nilsson, 2005), where it is stated that the interview protocol used during a semi-structured interview does not have to be as strictly followed as a during a structured one. The questions do not strictly have to come in the same order as in the protocol. Probe questions allows the interviewee to expand on answers more freely. (Bryman et al., 2005).

All the interviews were administered in the same way, reducing interfering factors between different interviews. During the interviews two researchers were present at all times, one taking notes on a computer, while the other was asking questions and actively interacting with the interviewee. All the interviews were recorded and transcribed in Swedish. The transcriptions were able to confirm the correctness of notes from the interview and identify remarks from the interviewees that were initially missed. For transparency it should be noted that at some points the transcriptions had a leisurely approach, summarizing long expositions into more manageable notes. The collected data was analyzed, more on this in Section 4.3.3.

4.3.3 Data analysis

Thematic synthesis is well defined by (Cruzes et al., 2015), see initial reading of Section 4.3, whom also propose a preferred way for synthesizing the primary data. The method proposed by (Cruzes et al., 2015) also closely corresponds to the synthesis method suggested discussed by (Saldaña, 2015). (Cruzes et al., 2015) describe the process for thematic analysis in five steps:

1. Initial reading of data
2. Identify specific segments of text
3. Label the segments of text
4. Reduce overlap and translate codes into themes
5. Create a higher model of higher-order themes

It is important to note that (Saldaña, 2015) does not include the first step, initial reading of data, although the authors deemed it as an necessary step in the process, some of the steps have also been adapted to this case studies context. (Saldaña, 2015) and (Cruzes et al., 2015) use different terminology, but the methodology process is almost identical between them. The method used in this case study was based mainly on the method presented in (Cruzes et al., 2015) report. This Section describes each of the steps in the process and how they were applied to the interview data.

Initial reading of data
This phase is extracting the data from the primary data source, in this case the interviews, for aims, context, and results. The extracted data in this phase was the combination of interview transcripts and notes, as discussed in Section 4.3.2, i.e. the notes taken during the interviews were complimented with the transcribed data. An estimate of the 49 pages was the amount of extracted information from the interviews, i.e. the "Initial reading of data", see Figure 4.2. The 49 pages from the interviews were then read by two researchers in order to obtain a holistic view and thus finding additional viewpoints that originally might not have been recognized.

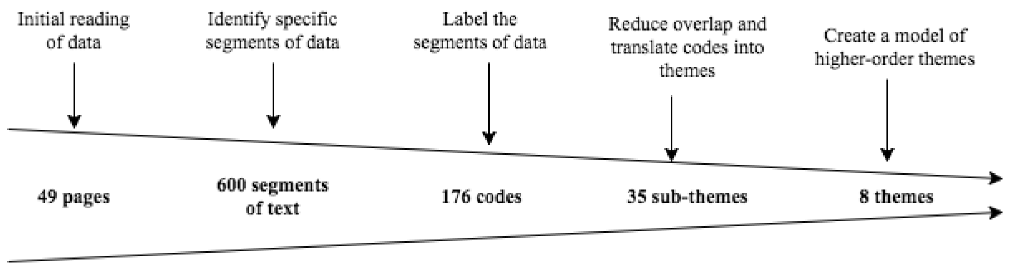

Figure 4.2: Thematic analysis overview

Identify specific segments of data

After the initial reading was completed the step was to identify specific segments of code. During this step the extracted information was analyzed and exported into a table where each segment of text that was found during the initial step was inserted. Each interview was kept in a separate table so that the different interviews could be held differentiated between to not lose traceability in the process. During this step 600 points of data was transferred into tables for further processing, see Figure 4.2

Label the segments of text

This step is best described by (Saldaña, 2015) whom called it "initial coding". Each segment of data that was extracted and exported into the table was read, analyzed and coded. Coded means that the data is labeled with a word or simple phrase describing the context, the result is exemplified in Table 4.4. Note that the text segments were in Swedish and the codes were in English. The coding was done by the same person who did the transcription for each interview. The researchers did the coding independently without knowing the other researchers codes. This meant that there was an overlap and different terminology between them. The labeling process resulted in 176 unique codes, see Figure 4.2.

Table 4.4: Example output after initial coding in step 3

Segment of text from transcription	**Code**
[00:25:21.12] Interviewer: What were the biggest pitfalls? The cost was much higher than we anticipated. You have to think about dimensioning the team correctly. Sure you can put a lot of people on the task and produce a lot but then it takes a long time to review everything.	problem (cost), team size
⋮	⋮

Reduce overlap and translate codes into themes

As the name implies this step focuses on reducing the overlap, and translating the codes into more general themes i.e. group similar codes, this step is also refereed to as the "second coding cycle" by (Saldaña, 2015). After the initial coding was done each of the codes from all the interviews were compared in order to search for similar codes. Once all the overlapping codes were found they were categorized into more general codes, reducing the initial number of codes from 176 to 35. These became the sub-themes in the next step when they are grouped, see Figure 4.2.

Table 4.5: Table displaying higher order themes, and their sub-themes

Higher-order theme	Sub-themes		
Purpose	• Communication • Complexity	• Goal	
Management	• Leadership • Change Management	• Core Team • Team	• Consultant
Investigation	• Opinions • Stakeholder Analysis	• Prestudy • Prototyping	• Test Project
Model design	• Diagrams • Method	• Abstraction Level • Consistency	• Hardware • Code Generation
Tool	• Version Management • Tool Opinions	• Maintainability • Tool Requirement	
Planning	• Vision • Time	• Roadmap • Future	
Way of working	• Verification • Project Structure	• Baseline • Maintainability	• Documentation • Traceability
Training	• Learning • Mentor		

Create a higher model of higher-order themes
To create the higher order themes the methodology (Cruzes et al., 2015) argued that the relationships between themes, i.e. sub-themes, and then create a model of higher-order themes. (Saldaña, 2015) then mentions that the higher order themes should be use to draw conclusions or to form new theories. The 35 codes were grouped independently by the two researchers and the groupings were compared and discussed resulting in higher-order themes that both agreed on. The grouping and themes can be see in Table 4.5. After the relationships between all sub-themes where thoroughly analyzed eight higher order themes remained, these themes are more carefully be discussed in Section 4.4.

4.4 Findings

This section presents the findings from both of the interviews, organized under the same higher order theme and sub-themes found in, Table 4.5. Note that all of the statements have traceability to citations from the conducted interviews.

4.4.1 Purpose

Purpose is a combination of goal and problem. When analyzing the interviews several different sub-categories were identified that are closely related to the purpose, these will be analyzed further in this section:

- Goal
- Complexity
- Communication
- Project differences

Goal

In this respect, goal refers to the underlying purpose, vision or problem that was used as the reasoning for the model introduction. This was a view shared by many interviewees. Deciding the purpose is of utmost importance when introducing models as it will guide the entire modeling process. Everyone involved during the implementation process needs to agree on and understand what the change will actually entail.

- J-SE1: “[...] the most important thing, that I almost always mention when I discuss [modeling]; remember why you are doing something”
- J-SE1 implied: Trying to understand the purpose is focusing on the most important thing.
- J-SE1: “[...] six months [after deciding to introduce MBSE] I understood that those seven people had made seven different decisions because we had all had different views of what MBSE was”
- L-SE1: “I have written about many different purposes, simulation, analysis and [...] documentation purposes. There you should perhaps apply more force, what is the reason we want this new method. [...] You need the purpose to be very clear [...]”
- L-SE1: “In our SysML model we haven’t had a programming purpose, rather it has just been about understandability and documentation”
- L-SE1: “If you aren’t going to make any changes [to the system], you should consider if it’s worth it. But if you for some reason need to make some change, then it could be worth it to model the already existing system, the purpose again”

The purpose was not fully understood when Saab Järfälla introduced modeling during the project version 1, see Figure 4.1, and was seen as one of the reasons the first version of the model did not work. During the model rework, see Figure 4.1, the purpose was more clearly defined, which in turn helped guide the rework process and make version 2 better than its predecessor.

- J-PM1: “For us, it was only about the complexity and to find a tool to keeping everything organized [...]”
- J-SE1: “The original purpose was that the model should be the only source of information, that was the mindset that fueled the work”
- J-SE1: “We needed to slow down, we needed time to think about our vision, we made some sort of plan [..], a risk analysis and we composed a purpose for modeling”
- L-SE1: “It has been a bit like we haven’t had a clear purpose”

When there is no consensus on what the purpose of the model is a ‘purpose creep‘ can occur. A purpose creep is when a model or method was created as an address a specific purpose but as time goes on additional purposes are added to it, making the model spread in different, potentially conflicting directions. Purpose creep was about to happen with model version 1 since there were suggestions coming in that did not align the what the goal of the model was.

- J-SE1: “Then you started to hear; ‘we also need the requirements in the model’, and since we already had the design and architecture in the model, why not write the verification cases so we can code-generate directly from the models? [...], and suddenly we had everything [in the model]”
- L-SE1: “It is a crossroad, either you say that the model is only for understanding

things in the beginning and then it does not need to be exactly correct, but it does not matter because it was useful during the development period [...]. The other is that you decide that you want a model that is a live representation that always is correct, then that demand more work but you can also get more out of it."

- L-SE1: "We had a purpose when we kicked things off, but then after some time [...] we reached that purpose pretty fast, and then new questions are formulated, [...] what is the purpose now, in the next phase? [...] Then the model explode, it becomes huge and points in all different directions, someone have added information about functions, and someone else has added information about budgets, then all of a sudden it's not good at anything. [...] Maybe we should just keep it [the model] to the purpose we started with"

Complexity
When Saab Järfälla started modeling, version 1, the main purposes was to manage the complexity of the new project. Writing, organizing and keeping consistency in documentation the old way would be an almost impossible task for such an advanced project. Many of the interviewees expressed that the use of models in their current form, version 2.1 with document generation was what made it possible to keep the project under wraps.

- J-AR1: "We would be nowhere near this state in this project if we didn't utilize models"
- J-PM1 implied: The system is so complex that it is impossible to organize everything without some sort of tool, and modeling is something that can manage this
- J-SM1: "We wouldn't be able to build this system [without modeling], it is much more complex than [the previous project]"

Communication
This Section discussed how people converse with each other and that people use the same vocabulary, i.e. use the same definitions for the same thing. A perceived benefit and purpose presented by a majority of the interviewees was increased ease of communication, i.e making it easier to communicate with other teams what you are currently working on.

- J-SM1: "Making people to able talk to each other is the most difficult part in my experience, if you make a change in one end it has to propagate through the organization and show in the other"
- J-SE2 implied that it is beneficial that a common language is used making people have the same interpretations and understanding of what is being discussed.
- J-AR1: "Most people [at Saab Järfälla] know the basic concepts, the whole [new] project is discussed in a number of terms which all come from the model"
- L-SE1: "to stand around a whiteboard and talk to each other, draw those lines; 'ahh, I see what you think, this is what I thought'. Having that type of discussion is almost just as much worth as the other benefits. [...] I would argue that it is not just the language and tool that is important, but the ability to be able to quickly create a model and use it as a foundation for communication"

Another important aspect that was presented by one of the interviewees is that the creator of a model or concepts takes responsibility for making sure that the receiving person understands what was received. That it is the creator's responsibility to start from the bottom and then make a logical transition to more detailed things.

- J-SE1: “A super important thing is that the person delivering the information takes responsibility for knowing how to communicate with the receiver. [...]”
- J-SE1 implied that in version 1, see Figure 4.1, the perception was that everyone was going to understand and work with the model was widespread. The team creating the model just handed them over to the developers and implementors, without communicating how to use them. The communication between the creator and receiver were insufficient.

4.4.2 Management

If MBSE is to be introduced successfully management needs to be on board. It is important that they understand the purpose and potential risks they might have to deal with since they have to organize the introduction later on. This theme refers to what managerial parts of the organization need to take responsibility of, these sub-themes are:

- Leadership
- Change Management
- Core Team
- Team
- Consultant

Leadership
Leadership in the sense that there has to be a driving force for a change to take hold. This is important because a lack of support and leadership from management was stated as one of the main reasons for problems initially.

- J-SE1: “It was a problem in leadership, communication, and steering of the introduction [...]”
- L-SE1: “There have to be few people in both project management and in company management who support this for many many years. Because it takes years introducing something like this”

Management should not accept to proceed with the introduction of modeling if the prestudy and understanding of the problem are insufficient. Management should be involved by demanding evaluations on who will be affected by introducing modeling, what connections there are and what steps should be taken. A big part of understanding the purpose is to agree on what MBSE is to them and the company.

- J-AR2: “The prestudy was simply not good enough, the ones who took the decision should have said no and said that the prestudy was insufficient.”
- J-AR2: “According to me it was the prestudy [that was the reason for problems] and that the decision was made on a prestudy that wasn’t good enough [...]”
- J-SE1: “In a meeting where seven people sat, we discussed it [MBSE] and we decided, yes we will introduce MBSE. Six months later I realized that those seven people made seven different choices because we in our head had had different perceptions on what MBSE was.”

In order to have the models created be maintainable, there need to be people who are dedicated to making sure that the model conforms to the set standard. You need someone who is structured and responsible for keeping everything in order. The person or people will be the support and guidance the team needs in order to create correct models.

- L-SE1: "[It is important to know] who owns the model, who can take the decisions and say: No we're not using [the model] for this but we are using it for that. We have something that's called Chief Systems Engineer, that is such a roll that can make such decision"
- L-SE1: "[If people disagree regarding the model] then you have to be able to go to someone who is sufficiently well versed and can make a decision and stand for it over a long period of time. That roll is terribly important"
- J-AR2: "[Correct models] requires leadership and a model manager who keeps track of it"
- J-SM1: "It is my biggest fear right now, that we won't have people that can uphold this model so that it is possible to maintain, that is what happened 10-20 years ago. We started building this model 5 years ago, it should last for 20-30 years. We must constantly be able to develop our system further. If you compare this to [the previous project], we started with that in 2000 and we still maintain it today"

Change Management
This Section deals with the fact that during changes it is not always the case that everyone will be keen on the changes. There will always be people that have different opinions on how things should be done. Depending on the extent of the change compared to how it was previously, introducing MBSE will require a strategy for change management. The key to change management is making sure communication works.

- L-SE1: "There have been more and more focus on change management"
- J-PM1: "Change Management is something very important"
- J-PM1: "[...] We realized that we will never get everyone to think that [MBSE] is the right way"
- J-PM1: "The most, most important thing I believe is having the people with you. That they think it's a good thing. [This is done] by showing that it works"
- J-SE1: Interviewer: "[If people say this is new, it seems scary] then you have learned something else, you have learned that you have a question of change management, a question of culture"
- L-SE1: "if the ones who fill out the information in the models don't quite understand why and they don't benefit themselves they will think it's tiresome. You give feedback to those who have added the information so they understand the purpose better"

Something that has happened a lot is improvements being suggested but that are not used or used as much as intended. The reason for this has been either that the aim was not high enough or that the change had to compete with other changes at the same time. There has to be room within the organization for a change which means that only a few changes can happen at the same time.

- L-SE1: "We have previously a small group developed a lot of improvements that they suggest. It can be a well documented new method, tool updates or new tools. But getting to the point of beginning to use it and benefit from it, there we haven't succeeded. The reason for this that the aim has not been high enough or that sometimes changes we want to do compete with other changes we want to do at the same time”

Since the project grew quickly and not everyone agreed on how the modeling should be done there was a lot of friction between architects, project managers, and line managers.

- J-SE2: “I am very pleased with how we work. It seems like some systems engineers are not prepared to work on that [lower abstraction] level that we work on, there is a difference in the level of detail we have in the system work now compared to how we have had it before”
- J-SE2: “When we started with this work there were some that murmured a bit and said: now you are removing all the fun from the software work”
- J-AR2: “Yes [it was better to involve more people next time]”
- J-PM1: “Since the project grew so quickly and not everyone agreed on how the modeling should be done different groups formed. A lot of frustration. This led to a lot of friction between systems engineers, project and line organization”
- J-PM1: “The main advocates for this are from the line organization, with bosses who felt that you should do something general that everybody could use. this led to the first two years having a lot of friction between different groups within the organization”

If everyone is a part of the decision making you will never get anything done as it is impossible for everyone to agree on everything. Sometimes management has to step in and remove difficult people who do not listen to others.

- J-SM1: “You can have someone that ruins the whole team and it has happened here, we removed people from the team, [...]. I think it was the only way to get forward. [...] You don’t work in a team if you don’t listen to others”
- J-PM1: “It led to the number of systems engineers being cut in half since they could not manage more”

Getting everyone to use the model takes time but as the change settles in more and more will see the benefits. The aspect of the time it takes for the change to become fully adopted is also discussed in Section 4.4.6 and 4.4.8

- J-SE2: “[The way of working in the form of looking at model] has not spread, but there are at least many developers now who interpret the documents and understand the syntax in there”
- J-SM1 implied that it is now that people agree that modeling is here to stay and that people are happy with it that people no longer dismiss it.
- L-SE1: “You have to make room for the change aspect. Consider the human aspect, you have to get used to doing one thing instead of another thing, it takes time and it can be tough. You can compare it with learning a new language”

Core-team

The team that together with a steering committee is going to have the main responsibility of introducing MBSE can be called core team. This way the risk of the project growing in different directions is avoided. The core team should provide alternatives and propose solutions that the steering committee decides on. The steering committee consisted of project managers, head of development and a few other people in the surrounding the made sense to involve, the committee’s purpose is to make sure the team’s decisions were anchored in the projects that are going to use the models.

- J-AR1: "I come with the proposals but it is a group that jointly makes the decisions since it is important that there is agreement"
- J-AR2: "4-5 people from different system levels who were involved in order to have opinions and propose how we should work with the method, what was important"
- J-AR2: "Yes, it was absolutely a good idea [to involve two people from system level] we focused on the architectural level"
- J-AR2: "About three people who worked on the design level were also a part and gave their opinions. They took part in deciding how interfaces, function, and stuff on that level should be defined"
- J-SM1: "You never finish if everyone has to give their opinion [which is the case if the team is big and everyone has opinions]. I think you need to be a smaller team and then maybe later expand it. [...] We brought in the competence on how we should do it and then we shrank it down again so now we're not that many again. Now it works, now we're in motion"
- J-AR2: "From this you would create two alternatives, preferably more if you can, one shallower and one deeper [model structure]. Then you have a mix of external and internal modeling competence that we have in the building and people who are knowledgeable in the domain and are good at what they do"
- J-AR2: "Bring in two to three modeling experts, they could be consultants. These three would be able to come with different approaches to modeling, then you could complement these three with five to six people from the organization who know how the system works and what needs to be modeled in the system"

Team
The core team becomes a team when additional members come in to help create and later on use the models. The team will see the introduction of modeling as a project with a start and an end building much of the model in one go. Some in the team will continue to model future components after the introduction is done. The team's size needs to be considered as more people entails more disagreement and more reviewing. On the other hand, a single person creating a modeling method might result in its own problems as the method does not need to be as stringent.

- J-SM1: "[...] it was a very small team that started in the beginning, maybe only two people who did a small study. Then we got the idea that we are going to do modeling. Then it grew very large and that is when the problems started"
- J-AR2: "A team can be six to seven people, eight maximum"
- J-SE1: "[...] If you have ONE systems engineer then you don't need a stringent method since you know what you've done, but the more people you are the more explicit you have to be with what you've done. Otherwise fundamental things around modeling do not work"
- J-SM1: "[The biggest pitfalls were that] it cost a lot more than we thought. You have to think about dimensioning the team correctly. Sure you can shove in a lot of people and get a lot produced but then it takes a long time to review everything if everyone needs to approve everything"

The team needs to be able to work well together which is why the team composition is important. Avoid having people that have very strong opinions and are very stubborn as there are a lot of things to agree upon for example how interfaces should be defined and what the naming conventions should be.

- J-SM1: "[The biggest downside is] that it has taken time, people are different and we have had different phalanges that want to do thing differently. So I believe that it is very important to have a tight team where you agree. When we started there were quite many who modeled. There was quite a lot of disagreement. A lot of time was spent on discussions"
- J-SM1: "[The way of making the introduction process better is] I believe starting with a few people that are tight and making sure that you know these people work well together. You shouldn't put together phalanges that think very differently I think. If you want to succeed you should have a tight team that know the domain, the tools, and the method"
- J-SM1: "The team composition is important, [...] if you have someone in the team who takes a lot of space, sure it can be positive in some cases but people can be very stubborn [...] it takes a lot of time if there is someone who always has very strong opinions. You have to agree on things all the time. [So that] the modeling is done in the same way so that everyone can understand each other's models because if everyone runs of in different directions it will not be any good"

The team should have knowledge in the domain, tooling, and method. Having the team members be the people who were the most knowledgeable of the system ensures that what is created is correct in a technical manner. The team should be based on someone that has experience with modeling and who believes in modeling and has a passion for it.

- J-SE1: "We took the people who were the most engaged and skilled in modeling and put them in a team and had them do the rework. The people we had doing this were the ones who were the most knowledgeable in the product [...] so that we ensured that the technical aspect was done correctly"
- J-SE1: "[...] the projects let go of their two systems resources [engineers] for five weeks, to let them rebuild the model, that was the success factor, bringing those people together"
- J-SE2: "I think the solution is lifting coders to systems engineers. You have to lift up a coder to take responsibility for a system function and that it works with everything else [in the system]"
- J-SM1: "[...] it has to be someone who is educated in the method. It's not that common that somebody is very talented and knows both the method and the domain. We don't have many of those"
- J-PM1: "I think some could have been saved if it was done as a smaller project first, where everyone has a passion for modeling, where everyone believes in modeling so that you don't have to convert nonbelievers at the same time you're trying to create something"

Consultant

This Section will discuss if the undertaking of introducing models should have external consultants involved. Some believe that that aim should be to not use consultants as far as possible as domain knowledge is essential for good modeling. Another factor to consider is how reliant the company would become on said consultant. Consultants might also have secondary motives as is suspected by one participant.

- J-SM1: "I think that as far as possible we should have employees in this, we started this having a consultant and then we were a little addicted to this person so he went out and we hired some others from outside who would come in and speak their mind. However, it was not positively received"
- J-SM1: "You have to know what you are going to build. There I think we started wrong, we picked up some people who could model and whom should tell the domain experts how to build a radar-warning system. Who knows best? If you do not have the domain knowledge, you do not really have the respect"
- J-SE1: "He is a consultant for a company that aims to be major representatives of [a certain modeling language] in Sweden. Now afterward, we have realized that his company might have an interest in getting [that language integrated]. We think there is politics involved but that's nothing we know for sure"

Others believe that hiring consultants is recommended or at least something to consider. A consultant was used for the creation of Saab Järfälla's modeling method and one participant thought it could be possible although just after pointing out that internal modeling competency is also needed.

- J-SE1: said that an Internal expert together with a consultant was responsible for the modeling method
- J-AR2: "bring in 2-3 modeling experts, you could bring in consultants. And we should have had modeling competency at Saab like we have now."

4.4.3 Investigation

A thorough investigation is required before a modeling method is chosen to be adopted and integrated into the current way of working.

- Stakeholder analysis
- Prestudy
- Prototyping
- Opinions
- Test project

Stakeholder Analysis
This Section points to the analysis that needs to be performed in order to specify who are going to use the models, or whom the suggested modeling method would affect.

During the model rework, see Figure 4.1, a new way of analyzing the different requirements that should be imposed upon the model was tested. It started by finding who the different stakeholders are, and what their needs were.

- J-SE1: "We started with defining; who are the stakeholders and what are their needs?"

By analyzing the stakeholders and their needs a vision for the modeling process can be created. Different requirements related to the tool, method, learning criteria and organizational changes can be discovered when the affected stakeholders are found. Since their expertise, knowledge, and resistance to change can be analyzed.

- J-SE1: “We gathered a wide range of different requirements and looked at what they might affect, we found they would affect: the method, tool, required training and the organization. By organization I mean if the organization need to create one or more new positions”
- L-SE1: “Take the stakeholders who you know will contribute and those who might oppose it, and then divide them into three categories; know, understand and create”

Prestudy

The prestudy is the research made before the models are introduced, but after the purpose for introducing models is decided. The prestudy should have a wide scope and ask people from different stakeholder groups. The result of the prestudy could, for instance, be whether the company should utilize a model based approach at all or if there are other approaches that could address the same problem.

Saab Järfälla did a prestudy before version 1, see Figure 4.1, but the general consensus was that the prestudy had a too narrow scope, since it only analyzed one modeling tool and modeling method.

- J-AR2: “A consultant did the prestudy, but it involved too few people and the scope was too small, only one modeling tool and one modeling method was studied and then it was approved”
- L-FD2: “I felt like the investigation was pretty shallow in regards to the code generation and how it should fit together.”
- L-FD1: “they had talked to people but they didn’t really ask us in the development teams, rather some enthusiasts they trusted”

Once the goal for introducing models was determined by Saab Järfälla, after the model rework, see Figure 4.1, they analyzed what different parts would be affected when moving from the current state of the project to their modeling vision. The prestudy should analyze how modeling could produce the largest benefit and how it would affect the organization as a whole.

- J-SE1: “So after we decided to introduce models we asked ourselves: ‘how does this affect us?’. Thereafter we looked though differed work tasks and the way of working and how that might be affected by moving from where we are today, which is a world full of documents, to a future where we only have one or more integrated models”

The prestudy for version 2, see Figure 4.1, led to limiting the scope of what the model should do. Limiting the scope decreases the potential risk of the purpose creeps, as previously mentioned in Section 4.4.1, making the model more effectively fulfill the set purpose.

- J-SE2: “The model is delimited so that it fulfills its purpose”
- J-SE1: “[..] why did we start this [modeling process]? Because we had a problem with architecture and design. We are already experts on requirements and verification, so let’s not touch that”
- L-SE1: “With SysML, there are so tremendously many possibilities, there are so many views and there are so many types of analyses you can do. That’s what it’s for but that also makes it difficult, there are so many possibilities. What it most important, what should we choose?”

Prototyping

A small-scale example of how the solution is going to be realized with models, the prototype can be seen as a way of validating that the modeling method fulfills the chosen purpose.

After the prestudy, a management team at Saab Järfälla held a meeting where they decided to introduce modeling into their new project. Although the created models never seemed to fulfill their intended purpose, and since the were too many people involved during the decision process, changing something was a slow process. This was because the decision could not agree on what the solution should be.

- J-SE1: "[...] during that meeting when we decided to introduce MBSE, six months later I understood that those seven people [at the meeting] took seven different decisions because we all had different views of what MBSE actually was"
- J-PM1: "This project [...] is the first real project that modeling is applied on [inside the company]"

It is beneficial to create a prototype in order to determine if the goal is achieved with the proposed solution and that everyone working with the models has the same view on how the modeling should be done. The prototype will enable the people that are going to use the models to give their feedback on how they think the solution should work or how it would affect them. The time it takes to create a prototype, for a team of experts, could take as long as six months.

- J-AR2: "They could have made some sort of prototype, that could have shown how it would look in different models. A prototype [could be built] in six months"
- J-SE1: "I think prototyping is very useful, it can be used to show; 'this is how we think this should work because if people don't see how it will work you can't really get a good response from people"
- J-SE1: "This is the model, you as a developer will get this on your desk, will this work?"

The prototype will find risks and problems earlier which mean costly retracing can be avoided. It might also be easier to convince people to accept modeling as the new method of working.

- J-SE1: "Once we have evaluated what we needed to do [...] we had also identified a problem with the tool, which meant that we had to change it"
- J-PM1: "I think the most important thing is showing that [modeling] works, this is how you convince people"

Opinions
This Section reflects the different opinions that were found during the interviews, this Section describes different views that people had regarding the gathering of information before the modeling actually started.

During the entire lifespan of the project, there have been different opinions, something that is mostly considered a good thing. But it was noted that it is important that people agree at the start since it is when the method is the most vulnerable. In the beginning of projects, the definitions of the most commonly used terms should be set. When everyone knows what something means, opinions will be easier to convey.

- J-SE1: "We had to decide what MBSE meant because that was a part of the problem [..]"
- J-SE2: "The experts had ideas about how they thought things should be done, but

there were no details, those we had to gain through work"

- J-SE2: "We had an expert that told us how to do things, but usually it turned into our interpretation of that"

It is also important to keep the number of people that are involved in the initial startup process small since having a smaller team allows for quicker decisions and a higher level of parallelism. The team members should trust each other.

- J-SM1 implied: If you are a large team everyone have to get a say in everything, he argued that one should start with a smaller team and then expand upon it.
- J-SE2: "At the very beginning we all had almost zero knowledge [about the modeling implementation process], so we didn't really know what to do, and then we were already quite a few people working on this. At the same time, we had to gain steadiness about how we were to work with each other"

Test Project
A smaller scale project could be used to test the new modeling method. The test project should be conducted after a prototype has been accepted.

Saab Järfälla started modeling in a large project which meant that every time something in the modeling method changed a lot of work had to be put on updating the model. It could be a good idea creating the modeling method for a smaller project first. One interviewee did, however, argue that that much can be tested beforehand but how it scales to a larger project might be difficult to say.

- J-AR1: "[...] everything can be tested in small project, but how they scale is a completely different thing"
- J-PM1: "The project was too large to start with since the project was growing so fast and everyone didn't agree how the modeling should be done [...]"
- J-PM1: "I think you could have gained a lot by trying this [the modeling techniques] on a smaller project first"
- J-PM1: "You could try the new [modeling] method in a sub-system first so that you can see that it works"

When testing if the modeling method works it should be tested in a live project so that people take the modeling seriously. If the testing is done in fake project people would know it and the results would be inconclusive.

- J-PM1: "I think it's a good idea to do it on a live project because if you do something in a pure test environment everyone will know that it really doesn't mean anything and there will be to little focus on it"

The team that tests how the method works should have an interest in modeling beforehand since the training would not have to be as extensively provided.

- J-PM1: "I think that if you would have run this on a smaller project first, where only people interested in modeling are involved [it would be better] because then you don't have to focus on making people like modeling while at the same time you have to produce something"

4.4.4 Model Design

This Section discusses how the model should be designed. Most everything regarding the design of the model can be adapted to fit the context and purpose. These are sub-themes found, all with their own considerations:

- Diagrams
- Method
- Abstraction level
- Consistency
- Hardware
- Code generation

Diagrams
In modeling languages such as SysML, there are a number of different diagrams. Findings regarding the uses of diagrams are discussed below.

When creating your model the emphasis should be on the information and if it is being conveyed properly. What diagram type it happens to be is not as important. Also if a diagram is difficult to understand nothing prohibits the use of textual descriptions.

- J-AR1: “It is important to know that when you’re building a model it’s not the diagrams that are important but the information around them”
- J-AR1: “Those who don’t know SysML should still be able to open the same diagram, read it, and get enough information and for that there has to be text”
- J-SE2: “We allow information that is not SysML to exist in the model, you have to if it’s going to be understandable”
- J-SE2: “It’s important to choose what information should be in the model”
- L-SE1 implied that you should not fill diagrams with too much information, “less is more”.

Important to note is that even if SysML is used not all SysML diagrams need to be used. It is possible to define company specific additions and removals to better fit the model’s purpose. In Saab Järfälla’s case, the system is very distributed so a lot of work has to be directed toward allocation and dynamic behavior prioritizing models that deal with those issues.

- J-AR1: “We don’t change SysML, we remove a bunch of parts. We only use a subset and we’ve added a number of notations that enable us to discuss different views”
- J-AR2: “We don’t use raw SysML, but we have introduced some of our own stereotypes. Different stereotypes are color coded so that it is easy to quickly identify if something is built in the wrong way”
- J-AR2: “I think the biggest gain is that the interfaces are unambiguous. It is possible to follow an interface through the system”
- J-AR1: “We have a very distributed system and we need to work a lot with allocation and understanding how the dynamics of different protocols look like”
- L-SE1: “It’s good having a DSL because get what you want, but it is bad in the long term since new people coming in don’t recognize themselves”

Some types of diagrams can have similar uses, for example, Use case vs activity diagram and Sequence diagram vs Activity diagram. These were big topics of discussion and will need an additional specification of how to use them.

- L-SE1: “I suggest weighing [use case and activity diagram] against each other if you are describing a function [...]”
- L-SE1: “We’ve had difficulties in drawing the line when we want to use [use case and activity diagrams] together in a good way”
- L-SE1: “Activity Diagram and Sequence Diagram are really the same things, they describe the same thing but in slightly different ways. There was a huge discussion a few years back over which we were going to choose”

In both Linköping and Järfälla DOORS was used rather than Requirements diagrams as there already existed infrastructure for handling requirements.

- J-AR1: “We have too many requirements to manage using diagrams”
- L-SE1: “We don’t use requirements diagrams for requirements since we have the DOORS tools for that so it would be double if we used it”

Method

This part deals with the method used for creating models and how it can be tailored. Other considerations in the methods use and who has what responsibilities is also discussed. In Järfälla it was successful having the team sit and model together.

- J-AR2: “Everyone sat together, we worked together to develop the method”
- J-AR2: “It is the modeling guide that dictates the way of working”
- L-FD3: “Even if you say ‘you work with this and you work with that’ it might be necessary to change something somewhere else and then it falls apart because of that”

Having a clearly defined method is important in order to achieve consistency in the model but if it is too extensive and inaccessible it will not be used.

- J-AR1: “The modeling guide says how you go between different views, how to interpret them and why you interpret one way or another”
- J-SE2: “We are few that model but when we are going to become more you have to tell them how things are connected and what they are supposed to do, how they should name things and so on, like a coding standard”
- J-SE1: “[In the first modeling version] we hadn’t managed to make anything more consistent in spite of going on for a pretty long time and in spite of having very talented engineers, I interpret that as the [old] method was too complicated”
- J-SE2: “My hope is that modeling guideline is going to get updated to a point where it is very clear how we work but we aren’t there yet”
- J-SE2: “Since the modeling guide was a bit difficult to read people didn’t read, over time things started to diffuse away and then we’ve gotten into situation where things sometimes are named in different ways”

The model is used in a prescriptive manner as it is created before implementation as a specification of the design. At Saab, there is an extra step of generating documentation that the code is developed from.

- J-SE2: "The model we create is what the implementation is based on"
- J-SE2 explained that the model generates textual documentation from which the code is written

Modeling architecture and modeling design or doing implementation require different skills and all the skills are not necessarily present in the same person. People are good at different things, cooperation is advised between team members.

- J-SE1: "The architect might not be an expert in the specifics of [Electronic Warfare] systems but he can do a fine job as an architect in the system. He does need domain knowledge about the product but he can get that through the help of others."

Abstraction Level
Determining how the abstraction levels should be defined does not have a clear cut answer which is why it can be difficult. It can also vary depending on the system which is why only pointers can be given.

- J-AR1: "It is important to be able to abstract at the right level and it is difficult for many"
- J-AR1: "What's in the architecture, design, and implementation depends on what project you are in, and that you have put up rules for, but the base is the same"
- J-SE2: "There is no easy way of choosing abstraction level, rather you have to go on your intuition"

The abstraction levels that ended up being effective for them was separating between Architecture and Design and leaving the implementation for developers/coders. Multiple participants stated that they believed it to be a good separation for most systems.

- J-AR1: "We are building an architecture and we are building a design, two different levels"
- J-SE2: "I think this project structure is a distribution that should be done in all projects"
- J-AR2: "I think there is a logical division for all different kinds of projects"
- J-AR2: "[...] We have a pretty loose connection between architecture and design, which make it easier to describe the system on a higher level"
- J-SE2: "One of the important things were that we divided the architecture and design into two [parts]. Because earlier we tried to break things down from the top, break it down to finer and finer [components] that should turn into the design"

The architecture is the overall and pretty loose representation of the system, it should be that simple to understand that people who are not versed in models still can see how the system is built.

- J-SE2: "[The architecture is a holistic summary of how the program is held together] you could say"
- J-AR2: "The architecture for the system components [the smallest parts] should reflect classes in the software. In the class-structure all system components are visible. They are using the function descriptions that exist"
- J-AR2: "There is a level where you can work more freely, where you could describe interfaces a little looser, rather than the level you are using to detail design. You

can create a document that is easy for many to understand”

- J-AR2: “Architecture and holistic principles should be so generic that people that are not that well informed [about modeling] should still understand how they are used or what they mean”

The design is under the architecture in abstraction and closer to implementation. It is split up into three abstractions, functionality, software and hardware. At this level, it should be clear how a function interacts within the system.

- J-AR1: “Our design consists of three parts, functional design, software design and hardware design.”
- J-AR1: “[The abstraction level of the functional design] should be low enough for understanding the function”

Avoid having too many levels of abstraction in the model. Too many levels result in a deep model which is difficult to comprehend. Saab Järfälla started with a deep model and was forced to do a rework into to a shallower model.

- L-FD2: “Don’t decompose things too much, it only becomes more cumbersome”
- J-AR2: “There were too many levels of the system hierarchy, the model became too deep”
- J-AR2: “It wasn’t possible to follow interfaces as they were handled on all levels; there were too many levels with interfaces. Dividing the functions and interfaces into so many levels [4-6 levels] made it impossible to get a holistic grasp over the system”
- J-AR2: “ To generate documentation from that [version 1, too many levels] was also worse”
- J-AR2: “It had a flatter structure, a clearer definition of interface, that was the goal with the whole migration [between version 1 and version 2]”

Low abstraction level is sluggish and expensive, this was at least what was concluded and thus the model does not go further down than to software component.

- J-SM1: “We don’t model software, we model down to software components. I think it would have been too costly [to model software]”
- J-AR2: “They didn’t want to be on the level of stating what data types should be sent and so on. They had chosen a higher level and didn’t want to lower it, some people disagreed [...]. Without any basis, the assessment was made that it would be too much work or that it would be too slow to work with”
- J-SE1: “Too many details in the modeling made it difficult to progress”

Consistency

There has to be an agreed upon method that everyone modeling in the same model complies to. Consistency is very important otherwise there is a risk people cannot understand parts of the model. The same amount of details needs to be present throughout the system if the goal is to do some type of analysis.

- J-SM1: “You have to agree [...] so that you model in the same way so that everyone can understand each other’s, models. Nothing will work if everyone runs of in their own direction”

- J-AR1: “We have defined 10 blocks that everyone knows”
- J-SE1: “In the model, you have to know what is what [...] when you describe something, a system, component or product you have to know what you’re talking about. Are you talking about the system or the software?”
- L-SE1: “If 10 different teams are filling out details on memory usage but only 7 do [...], then the total analysis won’t be of any use and with that, the whole idea falls. You need the same information with the same precision”
- L-SE1: “The biggest advantage is that everybody understands and have the same direct experience of what they’re seeing on the screen or in that diagram [...]. New employees and people coming in can understand the system quicker, how it works and that it is consistent throughout the company”

Hardware
At Saab hardware is not developed based on the system model. Since the hardware is developed separately on its own time schedule the system designer adds in hardware into the model after the hardware’s completion. Most of the hardware’s information is still kept in the form of documents.

- J-PM1: “We chose early on not to model hardware since we were forced to build hardware almost directly on more or less speculation”
- J-SM1: “Regarding the system work for the hardware is is still mainly textual documents but there is a bit of hardware modeled”
- J-SE2: “Hardware is developed for the most part before the functional work, it is not something that comes from the model”
- J-AR2: “The hardware arrives so early that the system designer doesn’t have time to model those parts. But in order to work with requirements it’s necessary that you are as good at modeling hardware and software”

Code Generation
This sub-themes discusses both generating of complete code and also generating of partial code, for example, shells with the correct interfaces.

In Järfälla the jump to code generation might not be far away from how the model is currently designed as the system is largely modeled in a similar way as in Linköping where thy do code generation.

- J-AR1: “Function Blocks are largely modeled in the way that the code generation parts are modeled in xtUML (what they use at Saab Linköping) but in SysML here. So adding a code generation engine is not that much work, we have built for the future.”

In Linköping, code generation is done using us Rhapsody for generation code skeletons that are used to validate the code generated from the design model in xtUML.

- L-FD1: “Rhapsody supports code skeletons and complete code, we use SysML and we use code generation from it so we get classes and then we code in c++”
- L-FD1: “If you have good enough tools, like Rhapsody for example then you can almost take code and ask Rhapsody to create a model from it”
- L-FD2 described how Bridgepoint with xtUML was used for prototyping by generation code from models. This was done by using an in-house developed code

generation engine for the xtUML model.

Code generation has its advantage in the form of consistency however since it requires the model specify a low abstraction level with more details it was decided for the time being that it is not something Järfälla is going to use.

- J-AR2: "Initially it is slower to get going with modeling [if code generation is the goal]. Now it probably would have been advantageous for it to require more work to change the interfaces between the parts, at least the interfaces between the big function blocks. In order to guarantee that it is consistent. So I would like to see that we imposed that now."
- J-AR2: "[The reason for not doing code generation was that] they didn't want the strong coupling between model and code. [...] I did promote it myself, but got shut down"

4.4.5 Tool

During the interviews most had opinions on the different tools they are using this is most likely because the tool is the interface the people work with every day. This Section will further analyze:

- Tool Requirement
- Maintainability
- Version Management
- Tool Opinions

Tool Requirement

Modeling tools can range from whiteboards sketches to advanced modeling tools. The tool needs to have all the functionality required to fulfill the purpose and goals.

- J-SE1: "While we were building a new roadmap, we also realized we had five different requirements for our tool"
- J-SE1: "The configuration management and version management didn't work as we wanted [or as the requirements specified], so we changed tool"
- J-AR1: "In other words, it's tool requirements that are the problem"
- J-SE1: "[The tool] could not satisfy our requirements, so we decided to change tools"

Something to consider is the possibility of becoming a "slave to the tool", becoming locked in with the tool and being unable to export the model to another tool. This is generally something Saab has to think about since they are going to maintain the project for years to come.

- J-SM1: "A challenge for us is the traceability [of requirements] between [different tools] when we had Enterprise Architect all the requirements automatically synchronized between the different tools. That does not work now [when we have switched tools]"
- J-AR2 implied: There is no tool connection between IBM Rational Rhapsody and IBM Rational DOORS because the dependency could make us slaves under the tool.

- J-AR1: “We have IBM Rational DOORS for requirements, requirements only live in DOORS. While the design only lives in IBM Rational Rhapsody”
- L-FD2: “If you model in Bridgepoint you are stuck with Bridgepoint”
- L-FD1: “The point with Rhapsody is that it supports exporting to other modeling formats”

Maintainability
Models and tools will be used over a long period of time which is why the maintainability aspect needs to be considered. If the model is incompatible with other tools than the one it was created in, the tool might get obsolete over time. This would either force the company to maintain the tool by themselves or losing the models altogether.

- J-SM1: “17 years later you still need to keep the tools up and running”
- J-AR1: “We have linked ourselves closely with IBM Rational Rhapsody but it would be fairly easy to switch tools, just very expensive and bothersome”
- J-SM1: “[The biggest problem with tools] is that the get obsolete”
- L-FD2: “Work with easy tools, things break down when something is too advanced”
- L-FD3: “Check so that the tool is worth having a long time”
- L-FD1: “You have to understand that if you choose a tool where the information is located in the tool then you might not be future proof”

When maintaining a large and complex system, using and updating textual documentation is a daunting task. The textual documentation is updated in the beginning of the project when it is first created, but after some years it could start to fall behind. Using models with tool support can help with keeping everything consistent and up to date, even years after they were initially created.

- J-PM1: “It feels safe to have the tool support now when the system is becoming so complex”
- J-PM1: “In the long term it feels obvious that this is how [the usage of models] should work”
- J-SM1: “This new project is so advanced that we would never be able to manage it in the old way [using textual documentation] since there are so many different system dependencies. Previously there were thick construction specification documents that were difficult to maintain. It always contained errors since it’s so difficult to maintain when the documentation is thousands of pages long. There the modeling tool help us, it’s consistent, everything is linked, and that’s what’s exceptional about the system model”

A problem with models is that Saab deems them to be non-archivable, meaning that they do not suffice as documentation. To circumvent the problem construction specification are generated from the models that can be archived, solving the achievability problem.

- J-SE2: “Generating documents is the only way information is considered permanent”
- J-SE2: “Formally we don’t trust tools, rather we want the information we create to be stored as documents, it is first then it is safe for the future”

Version Management
The tools ability to manage versions is an important functionality as it allows developers

to work in parallel. Variant management is the ability to have a common core of the model being reused in similar projects. Development in Järfälla spans over several increments and if there is a need to go back to check how something worked in a previous version, version management is a tremendous help.

Working with a graphical representation of code is very different from working with traditional version management for code since the models are graphical, several people can not work in the same file simultaneously. It is because of this necessary to think about how the selected tool or modeling method handles version management.

- J-PM1 implied: There needs to exist a capability to go back to previous versions of a software, and this was one of the main reason the tool had to be changed.
- J-SE2: "I don't think it's just creating a new branch [to create new versions from], I think you have to work with many small models for it to work, so there exist some sort of distribution"

The variant management was one of the things leading to the tool change, see Figure 4.1. It was a necessary change since a functioning version management was one of the tool requirements Saab Järfälla had. Variant management is something that the developers rarely encounter, meaning that they value the tool interface higher. In the new tool the user experience was worse, therefore, it is best to always inform everyone why the change was necessary so that no misunderstandings arise.

- J-SE2: "I don't know if [version management] is that complicated, I think it works similarly in the new and old tool. But it's important, no doubt about that"
- J-SE2: "But then i though it was completely unnecessary to change tool, it is just much worse [..]. We got no use for the benefits offered by the new tool, we just have to live with its downsides"
- J-SE1: "Not everyone agrees on if the first version of the model in the old tool were cumbersome and hard to use"

Tool opinions
During the interviews, almost all of the interviewees had opinions about the tool they were using and often brought up the topic by themselves. Put effort into choosing the right tool, do this by involving developers asking for their feedback. Making the developers feel more a part of the process might increase their positivity towards the tool and modeling method.

- J-SM1: "Everything revolves around tools. Being able to do different things with a tool is one thing, but it could completely change the modeling process"
- J-PM1 implied that the tool they are currently using does not seem to work that well
- J-SE2: "Our new tools interpretation of the modeling language is not as elegant as it was, it does not support everything you are supposed to be able to do with this modeling language"
- J-AR2: "Slowly but surely it gets better, the graphical interface have been quite bad [...]"
- J-AR2: "Our new modeling language have a better stringency, the implementation of the modeling language [in the tool] is done more carefully [...]"
- L-FD2: "nobody likes Dimension's version management since it is a complicated

program, we use git for everything it is possible to use it for"

- L-FD3: "We are pretty happy with modeling but we're not so pleased with the tool Bridgepoint"
- L-FD1: "It is really nice when the tools cooperate, it's entirely possible to connect Rhapsody and DOORS"

4.4.6 Planning

When introducing MBSE it is important to have both a long and short-term plan. It is not possible to plan everything in detail but having an idea in what steps lie ahead is recommended. The idea is management together with the core-team consider following identified sub-themes:

- Vision
- Time
- Roadmap
- Future

Vision
A long term plan should be in place as to make sure the modeling effort falls in line with the company's plan for the future.

- J-SE1: "how can we get from where we are today to being world leading in product development in 15 years"
- J-SE1: "[...] we took some time to think about what our vision was, we made some sort of plan for what the change would mean, we did a risk analysis and then when setting a purpose with the modeling, why we should work with modeling"

Time
Introducing MBSE will take time. In Järfälla it took almost three years to get to a point where the modeling works, this is because you learn over time, see learning in Section 4.4.8. It is important to understand the time aspect, because as with most projects, working under stress will probably negatively impact the quality.

- J-AR1: "For three years we've worked on the introduction of models"
- J-AR1: "I don't want to scare anyone but it takes a good while to learn how to build models, it's a craft, you shouldn't think you can manage to build complex models, neither day one, year one or year five. It takes time"
- J-AR1: "You cannot plan everything, you have to take it as it comes, we've been going at it for three years. We have had a guideline that was created two years ago that already didn't really match up [with the model]. So we've worked on it for two-three years and now I have [a method] that works, that is possible to follow. [This is] because we didn't have the possibility to do it before, we didn't have the knowledge"
- J-SE2: "At the same time there was an expectation that, already the first year, we should be able to create something from the model that was possible to implement. This was attempted, there was a time schedule and we have to have this document done until this. The models were stressed to completion"

Roadmap
The plan can be organized into a number of steps coming after each other. All working toward the long term goal. The reason for having steps is because it is impossible to introduce everything in a large change at once. The strategy for moving towards the goal could be two roadmaps in parallel where one deals with the technical aspect of what the model does and the other the cultural aspect i.e. what groups of people need to learn to model.

- J-AR1: "You cannot plan everything up front, it would take too much time. If you want to bring this in you have to start at once and you have to do it in small steps forward"
- J-SM1: "It is like everything else, all other projects you do, secure the start, how are we going to do this now, having some strategy from the beginning is important [...]"
- J-SE1: "And they worked with a roadmap. This is how we should do it, we will introduce this, then this, so gradually change towards more and more modeling. At the same time, they have a cultural roadmap that states which groups and staff that need to work with and how I create acceptance and receptiveness to working with models. And what steps do I need to do in parallel, it's a great idea"

Future
Even if the introduction is successful there will always be potential things to improve going on. The following improvements are a collection of suggestions could be good ideas as they are not yet tested.

Code generation could be a future improvement, however, those who mentioned it were tentative as to if it was something going to be introduced later on.

- J-SE2: "You could use the model for creating shells for things because we don't do that currently"
- J-AR2: "You could use the model to create shells for code, we did not do it from the beginning because it would have sluggish to work with. But it would be good now as it would guarantee consistency"
- J-SE2: "I don't really believe in it [code generation], I don't know if it will ever make sense to do, but to some degree, you could consider it, the problem is that you need even more detail [in the model]"

Creating views/projections from a (single consistent) overall model. Maybe even using the same model or connecting models across different locations within Saab could be beneficial.

- J-AR1: "We want to be able to extract certain information, certain parts, certain functions or certain hardware out of the model and deliver to an another part"
- J-AR2: "It would be positive to be able to converge models with Linköping in some cases since then both would work towards the same foundation/code"

4.4.7 Way of working

The way of working is closely connected to the modeling method and the way models are utilized. Implementing a model based approach into a project almost always infer some changes to the current way of work. While most of the interviewees did not directly speak of it, these are the different themes regarding the topic were found:

- Maintainability
- Documentation
- Traceability
- Verification
- Baseline
- Project structure

Maintainability
When starting to model a new maintainability perspective is required. Models work differently since modeling is not as widespread as conventional coding, the developers have to follow a modeling standard, this needs to be accounted for in the way the developers are working with models.

- J-SE1: "[During version 1] we had failed to make anything more consistent even though we had been doing it for a fairly long time, and despite the fact that we had skilled engineers"
- J-SM1: "You also need someone that is structured and willing to maintain [the model], that is probably my biggest fear right now, [not having someone maintaining the model] because that is what happened 10-20 years ago [in a previous modeling attempt]"
- J-PM1 implied: Sometimes people forget to update the documentation.

When building models even the short term maintainability has to be considered. Especially if the team working on the model is distributed. One change might lead to another which is why well-formed modeling methods and guidelines need to be in place.

- J-SE1: "When we are working with incremental development and have three different increments [for different variants] running at the same time you need a model for each one. If a change then is needed to the models, then all six different places have to change the same thing, then people started saying: 'this doesn't really work' "
- J-SE1: "I have said that we need to do this using a modeling tool because you can model in a drawing tool, but then you don't have support handling all the relations"

Documentation
Saab handles their documentation by using the model to generate construction specifications for the developers. Previously, Saab used the textual documentation describing their system manually. The documentation consisted of multiple different files with each file could be several hundred pages long. Generating the documentation instead, see Figure 4.1, ensures the documentation is consistent, with the references between different files being updated each time a new baseline is created. Textual documentation is the only type of documentation considered future proof at Saab, this is the main reason for still generating documents.

- J-PM1 implied: After a while, no one trusts the documentation since no one knows if it is still correct, manual inspection of the code is required at that point to know how things work.
- J-AR1: "We generate a [top level documentation] of 200 pages, where around 30 pages are diagrams"

- J-PM1 implied: The documentation is more consistent and only exist in one place, and that the developers trust it more than before.
- J-SE1: “There is much text [in the generated documentation]”
- J-SE2: “Generating documentation is the only way that information is considered permanent”

Generating the documentation has its drawbacks as well. One interviewee mentioned that the documentation has become more difficult to understand since it is not longer written by a human and the diagrams are sometimes cramped and small. For this reason, it is very important to accompany the diagrams and models with textual descriptions.

- J-AR1: “It is difficult for people to understand that they have to write text as well, modeling isn’t only drawing diagrams”
- J-SM1: “[The documentation] is much more difficult to read on a paper today, the diagrams are really small and almost unreadable”
- J-SE2 implied: A construction specification can be around 200-300 pages long, one can’t help but wonder if there isn’t a way to make it look better. The documentation is more of an encyclopedia, that can be used if needed in the future.
- J-SE2 implied: Information from the model easily turn into diagrams, and that there is always a way to generate things that look better.

Traceability
Two examples of traceability are being able to follow a customer’s requirement to the implementation and see how a change propagates through the system.

In previous projects, Saab Järfälla has used a traditional form of textual documentation from which construction specification is created. This way of working also meant that if a change to the documentation was made, affected documents had to be manually updated and failing to do so would result in an outdated and unreliable documentation. This new way of working puts the models as the primary source of information, where the information is guaranteed to be consistent.

- J-AR1: “The models are your database”
- J-AR2: “The greatest benefit [the model provides] I think is that the [model] interfaces are unanimous, that they can be followed all the way [through the system]”
- L-FD1: “These packages [in the software model] can be defined in a SysML model. So that you have traceability as well, from the higher system model down to your software model”

Verification
This Section refers to the process of verifying that the requirements are met and that system executes in the way it should.

When introducing models into a project, the methods surrounding the verification process needs to be well executed. Providing model access for everyone who is going to utilize the model is a step in the right direction, this will allow developers to validate each other’s design.

- J-SM1: “Everyone knows each other’s design since everyone validates each other’s design and can see their work in the model in a very open way”
- J-VM1: “I haven’t found any [benefits from modeling that affect me]”

When working with model validation new baselines need to be set fairly often, to allow testers to test the most frequent changes. Models are ever changing, making the validation process harder. The methods surrounding validation need to be thoroughly explored, so the process is as effective as possible.

- J-VM1: “It would be good to have more frequent baselines, [..], sometimes they make changes without generating new construction specifications directly”
- J-VM1: “I want the specifications before the increment start, because when it start the design specifications should be done, and then I also need to write the test specifications”
- J-VM1: “Models move a bit too much for my liking”

Project structure
In his context project structure describes the different processes and methods surrounding the modeling approach.

When implementing a new process or method there can be resistance to change, see Section 4.4.2, it is, therefore, important to train people, plan for the change, and create processes incorporating the change. Note that although the project uses a modeling approach, it does not mean that everything needs to be modeled.

- J-SM1: “We don’t use models everywhere, we mostly use them for larger projects and a few smaller ones. It is almost more common not to use them”
- J-SE2: “We currently using documents to communication with developers. But I think the more we work with this [models] the more people will understand the benefits of the model”
- J-SE2: “I’m very pleased with the way we are working, but it seems that some systems engineers are not willing to work on this [higher abstraction] level”

Line managers need to understand what a change mean, support, and include it into the new modeling methods. Even if the process or method are not connected to modeling, it might be something that needs to be accommodated.

- J-SE1: “A steering group should consist of [line] project managers for the [new] project and the chief developer. By doing that I got some guarantees that the line and the project we are trying it on [are going to cooperate]”
- J-AR1: “I don’t think we are quite successful yet [with the way of work outside the modeling process]”

A process that Saab utilizes when a major change has been made to a model is that a meeting invitation is sent to everyone working with the model. During the meeting, the changes are presented, by the person who did them, and in that way, everyone understands what is currently happening.

- J-SM1: “A downside and large risk is when we are going to implement this, you have to explain to them, this is how I have thought when I designed this [model] block you are supposed to implement”
- J-SM1: “We made a process where each time you make changes in your block, you hold a presentation, where the other developers can give feedback and you can communicate what it is you’re doing”
- L-FD2 implied: Working with models in parallel does not really work because of

problems with merging.

It is not certain that everyone wants to use models, this is something that needs to be taken into account when planning and designing the implementation. Even if people are willing to learn the introduction process takes time, which should be taken into account.

- J-SM1: “We thought most people would want to model, but some people were not interested in learning how to model, so we have actually stopped trying. You cannot teach people that don’t want to learn, so if they write documents other people later hack it into the model”
- J-SM1: “It’s now people agree that [modeling] is here to stay and that people are happy with it. Because if it is new it’s easier to say; ‘that is just a passing fad, I don’t care about that’”

4.4.8 Training

This Section describes how to learn to read and create models. With training there are a couple of things to consider, the two major considerations are the identified sub-themes:

- Learning
- Mentor

Learning
When introducing something new and possibly unfamiliar it will take time for people to learn the new method, new tool, new definitions and way of thinking needs to be considered. It takes time becoming proficient at modeling. Modeling is a skill that has to be developed, both in regards to the technical aspect of creating a correct model and in regards to adopting the new way of working.

- J-AR1: “I don’t want to scare anyone but it takes a good while to learn to build models, it’s a craft, you shouldn’t think that you can manage to build complex models, neither day one, year one or year five, it takes time”
- J-PM1: “We’ve been going for four years and we still haven’t managed to really bring them in into the model”
- J-PM1: “It takes more work drawing things in the model since there is a learning curve”
- J-SM1: “[...] a long starting period in regards to learning tools and methodology, I think that is where we lost our money, we spent a long time before we decided how we were going to do modeling”
- L-FD2: “The tool can be a big hurdle when getting started for new employees”

Not everyone needs to become model experts as is would be very expensive to train everyone. For many, it is enough being able to read what they need from the model. In some cases, skilled systems engineers either cannot or will not create models themselves, in those cases people who are proficient in models can help them insert their ideas into the model.

- J-SM1: “We thought for a while that everyone was going to do modeling, train every software engineer, we shut that down. It is almost impossible to focus on how you’re going to code functions at the same time as you’re learning the tool and modeling”

- J-SM1: “The whole endeavor of training everyone from software became very large. We had a big education plan for systems engineers as well”
- J-SM1: “Some were not interested in learning how to model, so we had to let it go, it’s not possible teaching people who don’t want to learn, they write on paper and others put their work into the model”
- J-PM1: “Systems engineers who can’t manage to do modeling explain their design to people who can insert the ideas into the model”

Learning by doing seems to be the best form of modeling education, actually working in the model together and having presentations of what each one has done.

- J-SM1: “It is almost impossible to teach this in any other way than that you sit together and model. [...] having presentations for each other of what we did.”

In Järfälla two tools have been used to host the model, both with their advantages and disadvantages. The new tool is more difficult to learn, although the tool is necessary to accommodate the tool requirements. The tool can be changed when new requirements emerge.

- J-SM1: “[Rhapsody] is a more difficult tool, it takes longer for the engineers to learn to use it. But there are pros and cons, you think carefully about that it fits into the whole chain. But we live in a changing world, you have to be able to switch tools”

If a guide is too extensive it is less likely that people will read it and learn from it. There is a point to be made that it can be seen as an encyclopedia but when learning the basics a simpler version would be a better help.

- J-SM1: “We’re working on making a simplified guideline, more like a manual so that you actually can put a sensible manual in the hands of a new [employee] who is starting. It became a thick pile of papers [...] it’s not bedtime reading. Sure it’s like an encyclopedia but the maintenance for something like that is terrible. I hope they make a slimmer version that’s easier to process”
- J-SE2: The guideline is extensive and difficult to read [...]
- J-SE2: If we had a document on our desk daily that was more easily accessible, it would be possible to point and say, this is how it should be done. But that is not how it works right now.

Mentor
There needs to be a person or people that will guide the company in the introduction of MBSE. The mentor will be able to help both in the creation of the modeling method and be a support for the team that does the modeling. There need to be at least a few people that are well versed in modeling.

- J-AR1: “[The mentor] is someone, if you hire or rent whatever, but it is someone who is on the project for a long time and knows how to work with questions like these”
- J-AR1: “You have to have a mentor, a mentor is the absolutely most important thing, someone who has done this at least three times before, otherwise you’re toast”
- J-PM1: “There are about five people who really understand how modeling works, who would be able to start from a blank sheet”
- J-PM1: “Every team has someone [who is a] system engineer in their team, who is

familiar or comfortable with models”

- J-SE2: “We would have needed an introduction, someone would have had to point and say: “this is a good way of doing it”, we didn’t really have this, we had a few that were talented, but they had not worked on this level”

4.5 Discussion

This Section will discuss the result of the case study and its meaning, it will also answer the research questions presented in Section 4.2. Each of the research questions will have their own subsection discussing the result for each of them. Since a thematic analysis was used to analyze the interview the findings are a combination of what was mention in Saab Järfälla and Linköping, the answer will separate the two locations rarely, see Section 4.4.

4.5.1 CSRQ1: What different problems occurred during the introduction of modeling in Järfälla and Linköping?

During the synthesis of the interviews a number of higher-order themes emerged from the data, this was presented in Section 4.3.3, this research question is answered with the same higher-order themes. The answer will be structured in the following way: first the problem will be presented in a bullet point, directly below the description of the problem will be presented.

- P1: Purpose was insufficiently described

The notion was that having a clear purpose was the most essential component of the introduction process, this is because the purpose guides the entire modeling process, and determines the way the modeling practices and methods should be shaped. Järfälla had a problem with their purpose, it was not known from the beginning which led to that that everyone had differing visions in mind of what the final goal was, leading to both confusion and frustration.

- P2: Purpose-creep

Since the purpose was not clearly defined at the start of the project in Järfälla, the ones doing the decision making regarding the model method did not know what the models purpose was, which in turn meant that the purpose and things the model should be able to do constantly grew. This problem is also known as purpose-creep since new purposes creep into models or modeling methods that already have an existing purpose.

- P3: Communicating changes in the model

Another problem was that the person creating the diagrams and models was not responsible for making sure the reader understood what was originally written. System engineers could create models that were later just handed off to the developers, that did not know how to read them.

- P4: Prestudy was too small, had too few alternatives and too many disagreed

The prestudy that was made had a to small scope, it only presented one method and tool for utilizing models, and only one alternative was researched, which in turn mean that if the modeling strategy it presented did not work, no alternative strategies existed, which severely increased the risks associated with the introduction.

- P5: Nobody took responsibility for decisions regarding the model

When the modeling method was introduced, the need for someone being responsible for the model was discovered, someone that makes sure the model is maintained since it is going to be used for several years. This person needs to support and guide the team that created the model so that it is kept updated.

- P6: Initial team that started to model was too large

When the initial modeling started, the team that introduced modeling and started modeling was too large. Which lead to decisions not being effectively taken, leading to even more frustration. One interviewee explained, that it is hard to create a product and update a method, while you at the same time have to convince people to use models.

- P7: Getting everyone to use the model takes time

When Saab Järfälla introduced models, the idea was that everyone involved in the project that could use models should use models, something that they later understood was impractical since many people did not want to work with models or learn how to use them.

- P8: Team composition

Initially, the team composition can be problematic, since people that have very strong opinions and are stubborn might delay changes that are necessary, e.g. how interfaces should be defined.

- P9: Unknown affected stakeholders

When introducing models all the stakeholders that will be affected by the change need to be discovered, they can then further develop the modeling methods and processes, and make sure that the model meets its intended goal.

- P10: No long term vision

When working a long term vision had not been formulated, which meant that the developers did not have a common goal to work towards, this, in turn, lead to people not fully understanding what the optimal path forward was.

- P11: Different opinions about what the solution is

Even although people agreed on a solution, everyone had different understandings of the initial problem, i.e. the purpose for introducing models; this, in turn, meant that everyone that came up with a solution had different paths to it, leading to nobody agreeing how the solution should be achieved.

- P12: People cannot say what they need until they see it

When people or stakeholders affected by a solution are asked how it will impact them and their daily work, they do not know how to properly respond, since they do not know how the proposed solution will work or look until they can actually see or use it.

- P13: The initial project was too large

The first project that the new modeling methods was tested upon was too large. That the project is large means there are many involved parties that have to agree when a change to the method is needed.

- P14: There does not exist a clearly defined modeling method

If no clearly defined modeling method is available once the modeling start the model will lose its consistency since everybody will model a bit differently from each other. This problem led to decreased understandability which makes the system more difficult to maintain.

- P15: Not everyone who needs and want access to the model has it

When the model exist, all affected stakeholders need access to it. Although sometimes all stakeholders have not been identified, effectively hampering those that need the model.

- P16: No easy way to determine what abstraction level to utilize

When starting to model choosing the right abstraction level is of critical importance, as also described by the findings of the literature review, see Section 3. No single abstraction level can capture the full extent of the model as described by (Bassi et al., 2011), which is the same problem that was encountered by Linköping and Järfälla.

- P17: Difficulties understanding the model

The model needs to be easy to understand, while at the same time providing enough information to be useful for the people how to use it. This can be connected to the information that was uncovered during the literature review, see Section 3, (Heldal et al., 2016) state; "Large and complex systems need some high-level descriptive model to obtain an overview of the system".

- P18: The model had too many different abstraction levels

This problem resulted in a model that was too deep and contained to much detail, which made it hard to maintain and also decreased the traceability within the system models. Since the models also grows large the loading times, and effectiveness slows down.

- P19: "Why don't we have code generation?"

This might not be seen as a problem by some but is something that almost always comes up when models are mentioned, and there are misunderstandings regarding the code generation topic and its perceived benefits.

- P20: Tool opinions

Everyone have different opinions about the tools that are used to create and maintain the models, this can be a big point of conflict between different team members and users.

- P21: Slave to the tool

Since the software and hardware Saab produces is going to be used for many years to come it is important that it can be maintained for all those years. If the tool that that project uses stops being supported it can become a problem if the project is entirely dependent on it and there is no way of exporting the model to a different tool.

- P22: Models can not be archived

Models are not considered to be future proof, meaning that they cannot be used as the sole form of documentation, the tool might become outdated, or the knowledge of how to model works lost in time.

- P23: Version & Variant management

Version management is something that was discovered to be harder with models than regular code since models are a graphical representation of a software system. The files which are worked in need to be locked so no two people can work in the at the same time. Variant management is also difficult since many of tools lack support for the functionality.

- P24: Introducing modeling practices and methodology takes time

The time from introducing models to management wanting to see results regarding the modeling processes was too short. This makes the integration rushed because it forces the developers and people who integrate the new methods and processes to at the same time be productive and meet set deadlines enforced upon the project.

- P25: Specifying new baselines is problematic

Working with models is a not like working with just code when using models the architecture and design for the system constantly change. When a large system architectural change happens it is important that the change can be validated towards the actual requirements.

- P26: Resistance to change

When changing a current way of work, there can be resistance towards that change, especially if the people have worked with using a different method for a long time.

- P27: Training takes time

When a company starts to use models, the people that are going to use the models need to be trained in how to use them, if they do not already know how to use them, this takes a long time.

- P28: The modeling guide is too complex

The method guide/book given to the people that create the models can not be too complicated or cumbersome to read. If the guideline is too complicated the developers will not use it, leading to that the model becomes inconsistent.

4.5.2 CSRQ2: What was done in order to solve the problems related to the introduction of models in Saab Linköping and Järfälla?

During the interviews at Saab Järfälla and Linköping, the interviewees presented different problems and potential solutions to them. The problems were carefully disclosed in Section 4.5.1, where both a problem statement and description were presented.

This research question aims to present the solutions that both locations discovered while using models, or during the introduction process. The answer will be structured in the following way: Each problem stated in Section 4.5.1 will be presented, directly thereafter a solution will be presented. the solution will draw direct parallels to the data synthesis and the literature review, see Section 3.

A combination of research question one and two can be found in Table 4.6. The higher-order themes presented in the table is the same as in Table 4.5. Note that the problems presented in Table 4.6 do not map to the sub-themes, rather the higher-order themes.

Table 4.6: CSRQ2: Problems mapped to their suggested solutions

Higher-order Theme	**Problem**	**Solutions of interest**
Purpose	P1: Purpose was insufficiently described	S1
	P2: Purpose-creep	S1, S2, S3
Management	P3: Communicating changes in the model	S3, S4, S5, S6, S7

Table 4.6: CSRQ2: Problems mapped to their suggested solutions

Higher-order Theme	**Problem**	**Solutions of interest**
	P4: Prestudy was to small, had to few alternatives and to many disagreed	S3, S8
	P5: Nobody took responsibility for decisions regarding the model	S3, S4, S5
	P6: Initial team that started to model was too large	S3, S9, S10, S11, S13
	P7: Getting everyone to use the model takes time	S1, S6, S9, S13,
	P8: Bad team composition	S3, S14
Investigation	P9: Unknown affected stakeholders	S12
	P10: No long term vision	S1, S13, S15
	P11: Different opinions about what the solution is	S10, S11, S14, S15
	P12: People cant say what they need until they see it	S10
	P13: The initial project was to large	S10, S11
Model Design	P14: There does not exist a clearly defined modeling method	S3, S15
	P15: Not everyone who need and want access to the model have it	S12, S16
	P16: No easy way to determine what abstraction level to utilize	S17, S18
	P17: Hard to understand the model	S6, S7, S17, S19
	P18: The model had to many different abstraction levels	S17
	P19: "Why don't we have code generation?"	S1, S2, S16, S20
Tool	P20: Tool opinions	S3, S10, S15, S21
	P21: Slave to the tool	S21
	P22: Models can not be archived	S7, S21, S22
Planning	P23: Version & Variant management	S10, S21, S23
	P24: Introducing modeling practices and methodology takes time	S3, S9, S11, S13, S16, S24
Way of working	P25: Specifying new baselines is problematic	S4, S5, S15, S18, S21, S23
	P26: Resistance to change	S6, S8, S10, S12, S13, S14, S16, S19
Training	P27: Training takes time	S3, S7, S9, S11, S13, S19
	P28: The modeling guide is to complex	S3, S6, S7

- S1: Define a clear purpose

  Addresses: *P1, P2, P7, P10, P19*

Clearly determining, and defining the purpose will lead to a better understanding of the goal for the change. It will also provide a common ground which can be used for further discussion and a better understanding for the problem. The purpose needs to include a long-term vision, a problem, and a goal.

- S2: Restrict the model

  Addresses: *P2, P19*

By effectively stating what the model is supposed to do and not supposed to do, many problem problems related to purpose uncertainty can be reduced. The restricted model's purpose will grow clearer and rival purposes will not as easily get integrated into the model.

- S3: Use a core-team

  Addresses: *P2, P3, P4, P5, P6, P8, P14, P20, P24, P27, P28*

A core-team is a group of experts that all have a high domain knowledge and is put together by project managers, although project managers can be included into the team themselves, the team should also include people that know to work well together. The core-team answers to a steering committee that decide if the proposals put together by the team are accepted or rejected, the steering committee is therefore responsible for the decisions over a longer period of time.

- S4: Have a person responsible for model maintainability

  Addresses: *P3, P5, P25*

Each team can have a person responsible that the model is maintained, in this way team members and developers always have a person that can help and motivate them to update the model.

- S5: Have meetings after model changes

  Addresses: *P3, P5, P25*

After a change to a model have occurred all affected teams are invited to a meeting where the changes are discussed and the affected teams are informed.

- S6: Provide additional training

  Addresses: *P3, P7, P17, P26, P27, P28*

Providing adequate training is of vital importance. The training can be done in many ways, e.g. training one person in each team to become an expert and then let him teach his colleagues, or giving everyone enough training to be able to model by themselves from the start. As discovered in Section 3.1.3 every modeling method require some form of understanding of models, therefore training is almost always required.

- S7: Utilize a common language

  Addresses: *P3, P17, P22, P27, P28*

Using a widely used language within the industry, leads to that more people can understand the intentions of the models or methods. The longevity of the language and support for it might also be longer and better.

- S8: Broaden the prestudy scope

  Addresses: *P4, P26*

The prestudy needs to have a fairly wide scope and should contain at least two different alternatives. By having two different alternatives the risk of introducing models is reduced, should the first alternative fail, a second one is always available.

- S9: Scale down management expectations

  Addresses: *P6, P7, P24, P27*

When introducing a new way of working or modeling method, it is important to provide enough room for the change to fully take effect. Therefore it is important for the management to not have too high expectations on what the project produces during the initial stages.

- S10: Create a prototype

  Addresses: *P6, P11, P12, P13, P20, P23, P26*

A prototype can act as a guide, that can later be used as the foundation for the created method. The prototype can also be used to provide context to the developers that are actually going to use the models once they are done, thereby allowing them to provide more reliable information about the proposition.

- S11: Test the modeling methods on a smaller scale

  Addresses: *P6, P11, P13, P24, P27*

Test the new modeling methods in a smaller project before implementing them on a larger scale, this might provide useful information and help to smooth the rough edges the method might have.

- S12: Performed a stakeholder analysis

  Addresses: *P9, P15, P26*

A stakeholder analysis can be performed in many different stages of a project, it simply takes a step back to gain a holistic view of the people a proposed change might affect. The analysis can be performed by identifying three different categories: know, understand, create. Know aim to find stakeholders that only need to know that there are changes happening, information about the project is given on a need-to-know basis. The second category, understand, aim to find the stakeholders that need to know how the models work, i.e. having enough information to be able to utilize models. The last category, Create, point to the stakeholders that need to be able to create new items for the models.

- S13: Create a project roadmap

  Addresses: *P7, P6, P10, P24, P26, P27*

A roadmap should be created to plan for the implementation of models in the larger organization, the roadmap should specify different steps the project need to take in order to reach the vision, each step should be incremental, building on the success of the previous step. A cultural roadmap can also be created so that each step is associated with a special group of people that need to be given special attention, to reduce their resistance to change.

- S14: Consider the team composition

  Addresses: *P8, P11, P26*

During the initial stages it is important that the team is well put together, so they are as effective as possible, therefore try to avoid putting people with strong opinions together that might delay important changes.

- S15: Do a prestudy

  Addresses: *P10, P11, P14, P20, P25*

A prestudy is different solutions to a specific problem, it evaluates if the solution is feasible, how it should be implemented and how to achieve the vision. The prestudy should also explore some possible modeling methods, that can later be used during the prototyping stages.

- S16: Improve communication between management and developers

  Addresses: *P15, P19, P24, P26*

Well-functioning communication between different company hierarchy levels is very important, everyone should know who to talk to if any questions arise. When large changes happen within a project or method the information should be communicated to all affected stakeholders.

- S17: Choose the right abstraction levels

  Addresses: *P16, P17, P18*

Choosing the abstraction level is done by having a deep knowledge of the company domain, while at the same time being an expert in modeling. If the modeling competence is not available within the company a consultant with help from an addition of domain knowledge from company experts is recommended. The recommendation should then be presented in a prestudy. But as Section 3.1.3 and 3.2.3 found there is no easy way to determine the required abstraction level, although it is near impossible to capture the full extent of the system in one abstraction level, while at the same time the interviews stated that the number of levels should not exceed four.

- S18: Separate the architecture from the system design

  Addresses: *P16, P25*

Separating the architecture from the system design allows for a more abstract way of realizing the system architecture, where fewer restrictions are in place. The system design is then allowed to have more formal restrictions and a more specific role.

- S19: Use a domain specific language (DSL)

  Addresses: *P17, P26, P27*

When implementing models the utilized modeling language can be adapted to fit the company context better, if, for instance, a specific diagram type fits an area within the company the diagram originally was not designed for, the company can make the model fit their needs. Caution should be taken when changing an already existing language since it might make it more difficult to learn for new people or people that already know the original modeling language.

- S20: Be careful with Code generation

  Addresses: *P19*

Code generation is a topic tightly tied with modeling, although it is not required when doing models. It was found during the systematic and additional literature review, see Section 3, that the biggest benefit from modeling might be increased communication rather than increasing efficiency by introducing code generation. Although code generation might be beneficial, it is best suited for companies that already have modeling experience.

- S21: Specify tool-requirements

  Addresses: *P20, P21, P22, P23, P25*

By specifying the requirements enforced upon a considered tool make it easier to argue which tool to select, leading to fewer discussions. An important aspect to consider is the longevity of the tool if the models are going to be maintained over a long period of time.

- S22: Increase archievability

  Addresses: *P22*

During large industrial ventures, the longevity for the models can cause problems, since there is no insurance that the models will be able to work several years in the future. Therefore generating documentation is a good trade-off, where the documentation is generated from the models and then archived, making the information future safe.

- S23: Carefully consider version & variant management

  Addresses: *P23, P25*

Version and variant management is quite different for models compared to usual code, this is why it needs to be taken into great consideration when choosing tools and modeling methods and should be consulted with experts.

- S24: Establish methods for setting baselines

  Addresses: *P25*

Since models are constantly moving and changing, it makes it difficult to validate if the requirements are satisfied. Therefore it is important to create clear methods surrounding the process of creating new baselines so that the models and code can be adequately tested.

4.5.3 CSRQ3: What are lessons learned of using models at Saab Linköping and Järfälla?

During the interviews in Linköping and Järfälla many different opinions regarding the modeling process they utilized surfaced, the most prevalent one was related to the purpose, and the prestudy done in Järfälla. The themes found when synthesizing the interviews will be used to present the discoveries.

Purpose
The purpose of introducing models into a project is extremely important, the purpose will strongly influence: the modeling process, chosen abstraction level and tool. The interviewee J-SE1 said "[...] the most important thing, that I almost always mentions when I talk [about modeling]; remember why you are doing something", knowing the purpose will allow the people working on the model to share a common goal, understand the problem attempted to be solved, and ease the planning stages. This discovery was not found during the literature review but still considered to be a major concern during the interviews, and one of the major causes to why Saab Järfälla's first attempt at introducing modeling was unsuccessful.

Though it could be argued that since the importance of the purpose was not found during a single article, it could point towards an underlying problem; most current modeling methods assume that the purpose for utilizing models is completely clear. While at least Saab Järfälla and Linköping saw an unclear purpose as a large pitfall and problem during their modeling efforts. Extra emphasis should be placed on understanding the purpose when introducing and updating the models.

When creating models purpose-creeps can occur, leading to the model losing its initial purpose. Purpose-creeps usually occur after a model is done, because then people realize that additional things can be achieved by the model. By broadening the models purpose the model might lose its effectiveness, making them more tedious to use and maintain.

Management
When using new methods there are always people that do not want change, this is where change management come in. Since Linköping have been using models for a longer period of time, they were quite happy with how their processes and methods worked, Järfälla, on the other hand, had more issues. Since the introduction of models is more recent in Järfälla they are still trying to get everyone to adapt to the new way of working. Linköping said that change management was one of the most important factors when they introduced the models, and Järfälla mostly agreed.

During the introduction of modeling, it is crucial to have the management support the project since they are ultimately what make or break the modeling effort. In Saab Linköping, for instance, it was mentioned that budgets for projects are set yearly, which in some cases made the projects unstable since the project did not know beforehand if they would be able to continue with their efforts next year. Management also needs to make sure to provide enough room within the company for the change to modeling does not compete with other changes simultaneously.

Some interviewees at Järfälla also suggested that the team should be small, i.e. only the core team, during the initial stages of the project, making the introduction more efficient. This is because a smaller team could be composed of people that management knows trust and work well together, making the decision process shorter, and allowing the team to work faster.

Investigation
It is essential that a proper prestudy is conducted before modeling is introduced, both Saab Järfälla and Linköping stated that their prestudies were lacking in scope. The prestudy should result in at least two alternatives so that they can be compared and there is something to fall back on if the chosen alternative did not work out.

It was also suggested that a prototype of the model should be created so the developers who were intended users and designers of the model knew how it would be like and give constructive feedback. The prototype will then act as a guide for how the modeling processes and methods should be integrated.

The new way of working with models should be tested in a smaller project before applying it on a large scale project. Saab Järfälla felt that the original project that they integrated the new modeling approach into was too large. The interviewees suggested that the new method should be tested in a smaller project first, a sub-system of some sort, where the developers know that it is a real project, but changes to the method can be done quickly if necessary.

Model design
Finding the right abstraction level was something Saab Järfälla expressed as problematic, having struggled with a model during the initial model as became too deep with too many

levels, resulting in it being unmanageable. A pitfall that Saab Järfälla also fell into was trying to use a single abstraction level to model the entire system, something that was found during the systematic literature review, see Section 3.1.3, to be problematic; the full extent of a system cannot be captured in a single abstraction level(Bassi et al., 2011). After the model rework, see Figure 4.1, two abstraction levels were used, architecture and design. The design is split into three abstractions, software, hardware, and functions. This falls in line with literature as it was found that having three abstraction levels is common, see Section 3.1.3 but at Saab Järfälla the lowest level, code, is left out of the model.

During the model version 2, see Figure 4.1, a clearer purpose had been defined, and it can be argued that Järfälla's intention with using models was better understood; "The consumer and intention strongly influences the abstraction level as well as the concrete syntax used" (Heldal et al., 2016) this can also be seen by the uncovered information described in Section 3.2.1.

The same modeling method will not fit every company as discovered in the systematic literature review, see Section 3.1.3, this is found to be true by examining Saab Järfälla and Linköping. Two branches within the same company utilize two different methods, from this it be concluded that if Saab Gothenburg wants to utilize models they will have to create their own method or at least modify an existing modeling method to their domain. Utilizing domain specific languages (DSL) is a common practice among many companies today, see Section 3.2.4. It is beneficial to utilize DSLs since then the language can be adapted to fit the specific company context, but also introduce some problems as new model users that know the language will not be familiar with the language additions.

Tool

Tooling seems to be a divisive topic with many differing opinions, it was brought up voluntarily by every single interviewee. The reason tool is mentioned so often is presumably because it is the interface people use to interact with models use on a daily basis, and also in some cases the reason for frustration. Linköping and Järfälla had different approaches regarding how the modeling tools were utilized, Linköping used them to generate code, while Järfälla utilized them to generate documentation.

Version management is more complicated than initially thought, this was mentioned by both Saab Järfälla and Linköping, and was something that both locations still struggled with. Since models are a graphical representation of a system different version management methods, compared to ordinary coding, have to be introduced. Both location also struggled with something called variant management, where a model could have a common core, that was used in several projects. The variant management would mean that if the core model updated all it's children also updated, something that was discovered to be much harder than anticipated.

Planning

As Benjamin Franklin once said, "If you fail to plan, you plan to fail", this is as true for modeling as many other areas. Planning was discovered to be an important step of the introduction process, planning is necessary to reduce the risks associated with the introduction process. According to one interviewee at Saab Linköping, the most important things to plan for was change management. It is also important to mention that this insight was first discovered once the modeling attempts had started, which can be connected to what (Chabrol & Sarramia, 2000) expressed in their paper, that a holistic view is necessary for design planning.

Way of working

Even although the two locations belong to the same company, the way they have implemented modeling into their current way of working are vastly different. That the way models are utilized differ between to location within the same company is no point of concern, as presented in the additional literature review, in Section 3.2.1 (Clarke & O'Connor, 2012; Coleman & O'Connor, 2008) state that the modeling process needs to be customized to the company context in order to provide the most benefits.

If a company wants to introduce modeling it is important that they understand that they will need to change or adapt the current way or working. This can be seen in the introduction at both Järfälla and Linköping where at both locations major restructuring to the way of working took place. How the way of working changes depends largely on if the model is used as prescriptive or descriptive, see Section 3.2.2, as it was mentioned during the interviews that the way of working needs to include some methods and processes that ensure the model is maintained, and up to date, i.e descriptive. Modeling methods and the way of working, going hand in hand is also confirmed in the additional literature review, see Section 3.2 and 3.2.1.

Adapting the validation process to models is difficult. Regarding this aspect, one interviewee stated that it had not become easier, maybe even more difficult to validate the system since modeling was introduced. The importance of the validation and verification process is also mentioned during the systematic and additional literature review, see Section 3, as one of the main areas of importance.

Not only problems have been discovered during the time spent with models, a large benefit is a reduction in perceived complexity that both Järfälla and Linköping reported. Some interviewees also noted that the communication had become easier, the models act as a common language, around which discussion can take place. Some interviewees from Järfälla also mentioned that the consistency of the documentation has improved since the introduction of modeling.

Training
Both Linköping and Järfälla both agreed that training is of vital importance when introducing new methods and working processes. During the systematic literature review, see Section 3.1.3, (Alhalabi et al., 2008) states that their method requires a basic understanding of modeling, this can then be assumed for all modeling methods. (Alhalabi et al., 2008) also explained that the number of people who need to learn the method might differ, some methods might only need one expert, while others require several. What was found at both Saab locations was that training was necessary, more than initially anticipated. The best way of teaching people was by letting each team create the model with the support a mentor that can answer questions and give pointers.

5

Recommendation

This chapter focuses on the framework that was created to achieve the goal presented in section 1.2, the validity treats connected to the conducted research can be found in section 2.5. As described in the method, see 2 the research questions presented in section 1.2 were answered by the literature review and case study. The questions were:

- **RQ1: What risks are there when introducing MBSE?**
- **RQ2: Which aspects of introduction and use of modeling can be tailored?**
- **RQ3: How can a framework for introducing MBSE be created?**

The answers gathered during these two studies are synthesized into the framework presented in this chapter.

The framework is divided into two different parts: a framework overview and a timeline. In the framework overview, each step of the framework will be discussed, what it is and why it is necessary. The steps are based on the literature review, see chapter 3, and the case study, see chapter 4. The timeline will show the sequencing of the steps, during which parts of the project they are executed and when extra attention needs to be placed on specific areas. This Section will also relate to the literature review and case study chapters.

5.1 Framework overview

The framework, see Figure 5.1, is a guide that advises practitioners on the introduction process of MBSE, while at the same time providing a foundation on which new models and methods can be based on. By following the framework risks identified during the literature review, see Section 3 and case study, see Section 4, can be avoided.

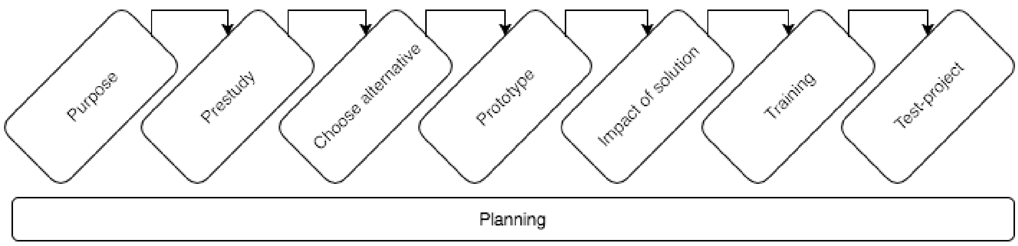

Figure 5.1: Framework for reducing risks associated with the introduction of modeling

The framework consists of eight steps, all representing a different part of the model introduction process. Note that this process is not static and can be changed depending on the company context or who implements it. The company context strongly influences many of the initial problems faced as described in 3.2.1, (Coleman & O'Connor, 2008) state that contextual problems are one of the main inputs to a tailoring process. The first three steps; purpose, prestudy, and choose alternative, help the practitioner to fully understand and analyze the current context.

It is also important to mention that the framework is iterative, allowing for trial and error during the introduction. It can happen that the practitioner has to go back and revise previous steps if the results are not satisfactory. All steps in the framework are presented in this section together with the reasoning behind each step.

5.1.1 Purpose

The purpose is the first and most critical aspect that needs to be identified in order to begin introducing modeling. Saab Järfälla and Linköping had discovered a clear purpose should be considered essential since it affects the entire modeling process. This is why the purpose needs to have its own step during the implementation process and is represented as such in the framework. From the literature it was uncovered that each company context is different, see Section 3.2.1 which is why the purpose can be different for each company as well. It was also found during the case study, see Section 4.5.3, that Saab Järfälla and Linköping utilized models in two completely different ways, despite both locations belong to the same company.

Start defining purpose of the models by determining if the models are going to be descriptive or prescriptive, see Section 3.2.2. (Heldal et al., 2016) described descriptive modeling is the as-implemented or as-realized architecture, and prescriptive is the as-conceived or as-intended architecture. The two modeling methods lead to vastly different end results but can help reduce the risk of misunderstandings which was discovered to be a big problem from the interviews, see Section 4.4.1.

Continue determining the purpose by performing a stakeholder analysis, so all stakeholders interested in the implementation process are found. It is through this stakeholder analysis that people who are going to be included in the core-team are found. The core-team is a small team with individuals that are interested in modeling, have a high level of domain knowledge, and that all work well together. Centralizing a method around a core group of people is something that was also found in the literature review specifically in (Alhalabi et al., 2008), who centralize their method around the concepts of a UML-expert. The core-team approach was also discussed in the case study and it addresses many different problems, see Section 4.5.2. The recommendation is to create a core-team is as it can reduce risks associated with the implementation process.

5.1.2 Prestudy

Perform a prestudy to gather information about what different alternatives there are to solve the problem presented by the purpose. The alternatives presented by the prestudy should be able to fulfill the purpose derived during the previous step. The goal of the prestudy is to gain a holistic overview of both the system and the project structure and use it to be able to find how modeling would be the most beneficial. Having a holistic system overview was perceived as important for finding potential solution areas during the systematic literature review, see Section 3.1. Both the methods presented by (Alhalabi et al., 2008) and (Chabrol & Sarramia, 2000), which were found in the SLR, include a person that have a holistic overview of the system so that because they then know how to model the given system. The prestudy should be conducted by the core-team that was assembled in the previous step, *purpose*, since they are domain experts as well experts in their fields.

During the case study, it was discovered that a prestudy had been conducted before the implementation of modeling at both Saab Järfälla and Linköping. The process of creating a prestudy was generally something that was perceived as a good thing, even if the actual execution of the prestudy ultimately failed because of scope was too small and did not cover enough alternatives, see Section 4.4.3 and 4.5.1. As presented in Section 4.5.2 and 4.5.3 the prestudy is important and should be done carefully. It was uncovered during the data analysis of the systematic literature view, see Section 3.1.3, that the methods used in other projects could not be directly transferred, but rather require some sort of modification to fit the company context.

The case study and literature review revealed that the prestudy should contain enough information to achieve the proposed purpose, i.e, goal, problem or vision. The prestudy need to consist of information regarding different; modeling languages, DLSs suggestions, tools, modeling methods, i.e, information that help solve or achieve the purpose, see Section 4.5.2 and 3.1.3

5.1.3 Choose alternative

During this step, the core-team should gather enough information to be able to commit to a specific prestudy solution for the found problem or process, that achieves the intended purpose for the introduction. The two previous steps aimed to gain a clearer understanding of why something should be done, while this step aims to answer how it should be done. During the case study, see Section 4.4.3, it was found that some developers, at both Järfälla and Linköping respectively, felt management made a mistake by accepting relatively weak prestudies. This step seeks to minimize risk by allowing for the developers to give feedback, and allow for the core-team to present the different alternatives within prestudies.

The information should be gathered by the core-team by talking to users, preferably in an interview where the interviewee can fully express their views of the prestudy and the answers can be noted. During the information gathering from the intended users there are some aspects that should be analyzed, such as:

- What is the current modeling experience?
- Does the proposed prestudy solve the actual problem?
- Which of the alternative prestudies have the largest impact?

All modeling methods require some sort of modeling knowledge as disclosed in the literature review, see Section 3.1.3, consequently it should be examined if there is modeling experience in each team that is going to utilize the new models and methods. As discussed during Section 4.5.2 in the case study, it is essential to provide adequate training for the developers and users of the model. Depending on what type of method that is going to be utilized different levels of modeling knowledge is required by different people, see Section 3.1.3, this is later used during the planning stages, see section 5.1.8. The current modeling knowledge for the interviewee will help to determine how much training is going to be necessary for the model users.

The interviews should ask the interviewee to give their opinion on the solution presented by the prestudy, if they think that it will solve the intended problem and if they believe the purpose is reasonable. The person being interviewed might hold important perspectives that contradict the originally identified problem, e.g. a problem that originally was believed to be related to documentation might, in fact, be related to the communication within the company, an insight that was gathered from the interviews. This perspective was discovered during the case study, see chapter 4, where some developers felt as they had been left out of the decision process, despite being the ones who were going to use it.

Finally, the interviewees need to answer, which prestudy alternative they think is most suitable for the presented purpose, problem, and vision. All the interviewee's answers should then carefully be considered so that the most relevant prestudy is later chosen and implemented.

5.1.4 Prototyping

Once a suitable alternative from the prestudy has been chosen, a prototype for that alternative should be created by the core-team. In the case study, see Section 4.5.1, a few problems were presented that are addressed with the use of prototypes. The problems related to that people from management, core-team, and the developers did not fully understand what the proposed solution was or how it was supposed to work, i.e., that they perceived different paths for the same alternative. This could be avoided by using a prototype, see Section 4.5.2, which would allow the intended users of the model to actually see how the alternative is supposed to work, thereby allowing them to provide better and more constructive feedback.

Since the purpose and context, see Section 3.2.1 and 4.5.3, are the main inputs to the tailoring process it is imperative that they are fully understood by the core-team. The prototype will also act as a guide for the core-team to make sure that everyone within the team has the same perception about how the purpose should be achieved. By having a prototype it is also easier to understand whom might get affected by the change, and make the process of conducting a stakeholder analysis easier.

5.1.5 Impact of solution

Analyzing who a change affects is very important since it reveals who needs training and at what time. It might also reveal important information about how to structure the roadmap that should be created, the roadmap is described further in Section 4.4.6. During the case study, see Section 4.5.1 and 4.5.2, it was found that three different problems were connected to a lack of stakeholder analysis. The problems can be addressed by conducting a stakeholder analysis, see Section 4.5.2. It was recommended by an interviewee in the case study that the stakeholders should be categorized into three different groups that are affected by the proposed change: *know, understand* and *create*.

The *know* category are people that need to know that a change is happening that might affect them, these people might not require any training but they still have to know that people are going start using models. The *understand* category are people that need to understand how to use the models or new methods, while the *create* category represent those that need to be able to contribute to the model, e.g. knowing the modeling method and being able to use the tools and so on. The latter group will require the most attention in regards to training.

5.1.6 Training

Training was a common theme during the interviews in the case study, see Section 4.4.8, and was generally perceived as a key aspect to a successful introduction. During the literature review, see Section 3, it was established that all of the examined methods required some form of modeling experience suggesting that training will be imperative if the company does not already have that kind of experience.

During the step *choose alternative*, it was discussed that the level of modeling knowledge should be gathered. That information can now at this point be used to provide the required level of training. A problem that was found during the case study, see Section 4.5.1, was that the time required for training was not accounted for, effectually making the training period too short, therefore as mentioned in Section 4.5.2, additional training should be provided.

There exist different ways training can be conducted, during the literature review, see Section 3.1.3, where several methods employed a single person was trained to have expert modeling knowledge, while the rest of the team did not learn as much but could ask the expert for help. This way of structuring the training was also recommended in the case study interviews, see Section 4.5.2 and 4.5.3.

The reason this step is after the impact of solution step is because the prototype needs to be accepted before the project is ready to be tested in a real project, this is described in more detail in the next step *Test-project*, see Section 5.1.7. The model designers need to be educated in the new method before they can test it.

5.1.7 Test-project

Several of the interviewees from Saab Järfälla mentioned that the project they started to integrate the new modeling techniques into was too large. It was proposed that the initially the new methods and modeling techniques should instead be tested in a smaller project first. The smaller project was suggested to be a real project that the company is currently working on because if it is only a dummy project there is a risk that it will not be taken seriously, see Section 4.4.3. The test project will provide valuable lessons for when modeling is introduced in the larger project. As mentioned in the additional literature review, see Section 3.2.3 the models need to be scaleable.

The scalability of the method was also discussed in the case study, see Section 4.5.3, where it is said that it is better to try a new method on a smaller scale before later expanding it to include larger projects. It was also argued that at some point the new method simply has to be tested in a large project since how well the method scales cannot be analyzed theoretically.

The test project is the last step before the actual implementation of the new modeling method. It is worth noting that additional training for all the affected stakeholders that was discovered in the impact of solution step might be necessary if the method requires it, see Section 5.1.5.

5.1.8 Planning

Planning needs to be done continuously during the entire span of the introduction, see Figure 5.1. Planning is done in parallel with all the other steps during almost the entire introduction process, with only a few steps pausing it. Project management should start planning after they have assembled a core-team in stage one, see Section 5.1.1, then as new information is discovered then the plan will have to be altered.

Section 4.5.3 states, that the modeling processes and way of working often go hand-in-hand, changes to the modeling process will affect the way of working and vice versa. This requires planning and is confirmed by the systematic literature review, see Section 3.1.3. The plan needs to be adjusted over the entire lifespan of the project to avoid many of the risks or problems presented in section 4.5.1. Sufficient planning plays a part in many of the solutions presented in Section 4.5.2 and the lessons learned in Section 4.5.3.

Planning will force practitioners to constantly think about who the model is for, and how it is going to be used. By planning the decisions they have traceability, and the core-team and project management can stand for them for a longer period of time. Which was one of the problems presented in Section 4.5.1.

Creating a roadmap was found during the case study, see Section 4.5.2, to be a solution to several problems presented in Section 4.5.1. The roadmap can be used to plan the introduction. The roadmap should contain the different steps the project needs to take in order to reach the vision defined in the purpose step, see Section 5.1.1. It is also recommended that two different roadmaps are created, one for the implementation process and another for the cultural change process see Section 4.4.6 . The cultural change should plan for which groups of people need extra attention so that the resistance towards the change, is as small as possible.

Planning should also include a plan for the required training, given the information gathered during the *Choose Alternative* step. The plan is first applied practically during the *training* step, see Section 5.1.6. It was also discovered during the literature review, see Section 4.5.2, that team composition is crucial when building the team who are going to create the initial models.

5.2 Framework Timeline

In the previous Section 5.1, the different steps were presented that were based on the case- and literature studies, all of which hinted towards an ordering or timeline. This section presents when each of the steps should be executed during the project and who should perform them.

Figure 5.2 shows how all the different steps presented in the framework overview, see Section 5.1, are connected, the figure also shows when each of the respective steps should start, and who are responsible for each. Figure 5.2 shows four different stakeholders, *company*, *project management*, *core-team* and *developer*, this section will further explain the different groups that conduct each step, and how the steps are connected.

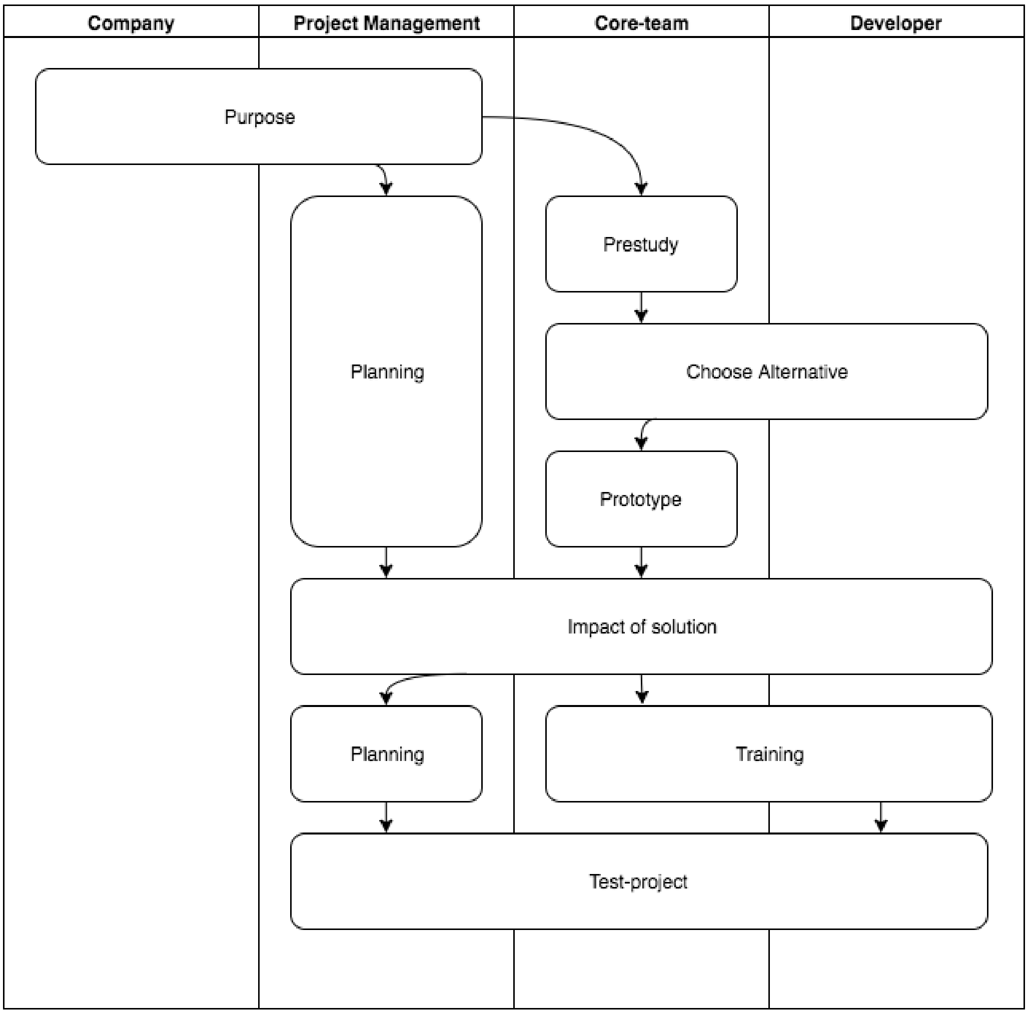

Figure 5.2: Framework on a timeline, presented in a swim-lane diagram

5.2.1 Company

The *company* in Figure 5.2 represents people within the organization that have the power to start new projects. These people either notice a problem within the current organization or have a problem presented to them, and they subsequently start a project in order to solve it, handing it over to the project managers.

After the company has handed over the problem to the project management it is essential that the company supports the project over a longer period with both human resources and funding; In Section 4.5.1 risks were discovered to be connected to the company not standing behind and supporting the project for long enough.

5.2.2 Project Management

The project management in Figure 5.2 are the people in charge of the project where modeling is going to be introduced. When the company has handed the purpose to the project management, see Section 5.2.1, the vision, problem, and goal need to be clearly defined. When the purpose is clearly defined the project management should conduct a stakeholder analysis.

The stakeholder analysis, Section 5.1.1, should be conducted to find people that can be a part in the core-team for the MBSE introduction. The core-team should consist of three to five people that all have a high level of domain knowledge, and are experts in their fields and have extensive modeling experience, see Section 4.5.2. If there is no modeling experience at the company consultants will have to be hired. The core-team will be responsible for the development of the MBSE methods and ways of working, while the project management has an administrative role planning for different milestones.

When the purpose step is complete the project management starts planning for the different steps during the project, see Figure 5.2. During the entire planning process, the project management continuously communicates with the core-team and incorporate their findings into the planning. When a new step of the project is started the core-team can also rely upon and adopt the planning.

After the core-team creates a prototype they are going to perform a new stakeholder analysis in order to find who is affected by the proposed model. In parallel the stakeholder analysis, project management should gather information about how the current way of working needs to change to include the new methods suggested from the core-team. Both the literature review and case study concluded that the way of working and modeling practices go hand-in-hand, see Chapter 3 and 4. Once the impact of solution step is done the project management needs to plan for how the MBSE approach created by the core-team is going to be introduced into the test-project. This is most suitably done via the use of a roadmap as discussed in Section 4.5.2.

Note that the planning stage is parallel to the other steps and relies on information obtained in them. The other steps, in turn, rely on the information obtained and created during the planning stage.

5.2.3 Core-team

The core-team is a group of people that work well together and consists of the domain and/or model experts. Using a core-team reduces many of the risks associated with the introduction of modeling, see Section 4.5.2. The core-team is responsible for many of the steps presented in Section 5.1 as can be seen in Figure 5.2

The core-team starts by conducting a *prestudy*, see Section 5.1.2, that is based on the purpose that was constructed by the project management and company, see Section 5.2.2 and 5.2.1. The aim of the prestudy to find how the goal, problem, and vision presented by the purpose should be solved or achieved. The result of the prestudy should be at least two different alternatives should have been created. When the prestudy is finished one of the alternatives will be chosen.

In the next step *choose alternative* the core-team need to be able to commit to one of the prestudy alternatives. Since the developers are later going to use the models, and know how the system currently works, they are consulted during interviews. Section 5.1.3 presents the three main points that should be analyzed during the interviews; current modeling knowledge, if the proposed prestudy solves the problem, and which of the prestudy alternatives has the largest impact. The developers modeling knowledge is then communicated to the project management, and the selected prestudy alternative is used to create the prototype.

The core-team creates the prototype that will act as a guideline or demonstration that can be presented and discussed with both developers and other core-team members. The prototype will act as a blueprint for how the modeling will work in the test project, the only difference is that changes can happen quickly as only the core-team are using it. Once the prototype is complete the impact of the solution will be analyzed during the *impact of solution step.*

The aim of *impact of solution* is for the core-team to be able to determine who will be affected by the new modeling method proposed by the prototype, see Section 5.1.5. The core-team will work closely with the project management during this phase so that the new method and its associated way of working can be planned for by the *project management* team. The developers also need to be asked if they feel the proposed way of working indeed solves the problem that it is designed to solve, while at the same time are being satisfactory for them to use.

The training stage is initiated by the core-team, using plans created by the *project management*. The core-team should be involved in the training process since they will be a core part of the development team during the test-project. Their main focus will be on the improvements of the modeling methods rather than development. After the training process, the core-team and *project-management* start the test-project migration, something that the project management have planned for a long period of time.

5.2.4 Developer

The developers are the people that are going to develop and utilize the models. They are only involved during three steps: *choose alternative*, *impact of solution*, and *test-project*, the reason for this is respecting the time of the developers as they continue working as usual.

In the *choose alternative* stage, the developers are only partially involved, since the core-team need to interview some of the developers to get their opinions. When the core-team is analyzing the impact of their prototype the developers inspect that the proposed solution actually solve the problem, and how they are supposed to use the solution.

The only large undertaking the development team is going to have is the training and test-project, where they have to start working using the new modeling techniques and methods. During the *choose alternative* and *impact of solution* stages, see Section 5.1.3 and 5.1.5 the developers are only having a small role; they are interviewed and questioned about their opinions of the current state or prototype.

6

Conclusion

The main goal of this study was to create a framework that provides recommendations for introducing MBSE. The research consisted of a systematic and additional literature review, and a case study. The systematic literature review (SLR) provided an unbiased foundation on which the framework was built, while the additional literature review (ALR) added surrounding information about the domain that was out of scope for the SLR. After the literature review, a case study was conducted.

The case study consisted of interviews and aimed to answer what risks Saab Järfälla and Linköping found during their introduction of models. The interviews focused on key personnel that was involved during the introduction process, and some developers that currently utilized the MBSE approach. The combination of the SLR, ALR and case study lead to a number of interesting findings:

The purpose needs to be clearly defined when introducing MBSE, having an unclear purpose will undoubtedly lead to problems further on in the project. Assemble a core-team that investigates how the modeling introduction should be performed, they will also make sure that the purpose is fulfilled. The core team should create prototypes and conduct prestudies among other things, but most importantly allow for the introduction process to be centralized around a core group. Introducing changes into a project take time, planning for the changes, and providing them with enough time will allow for new ways of working to be created and for the new methods to flourish.

The way of working is found to be closely connected to the methods that are created to support the models, even so, to successfully introduce an MBSE approach the different processes surrounding how the project work need to be changed as well, i.e; the modeling methods and way of working go hand-in-hand. To understand the new methods and ways of working, sufficient training needs to be provided to all affected stakeholders. The stakeholders are obtained by the core-team and by analyzing three different categories; *know*, *understand*, *create*.

During the thesis, a framework containing the aforementioned findings and more was created. The framework answers what different areas of the introduction process that can be tailored, and in some occasions provide examples about how to do so. The framework is also presented on a timeline to make it easier for practitioners to initiate the introduction, and provide help for whom should work on what.

6.1 Future Work

One of the most prominent limitation lies in the fact that the recommended framework has not been tried in a real industrial project, it is, therefore, hard to determine what positive aspects the framework provides in practice. The framework should be tested in a real project where the process can be further improved.

The topic of model or MBSE introduction is interesting, that has not yet been fully extensively described in the literature. Further research should be undertaken in the form of additional case studies or experience reports, to confirm the reported results.

Bibliography

Alhalabi, F., Maranzana, M., & Sourrouille, J.-L. (2008). A uml based methodology to ease the modeling of a set of related systems. In *Software engineering advances, 2008. icsea'08. the third international conference on* (pp. 51–57).

Andersson, H., Herzog, E., Johansson, G., & Johansson, O. (2010). Experience from introducing unified modeling language/systems modeling language at saab aerosystems. *Systems Engineering*, *13*(4), 369–380.

Baruzzo, A., & Comini, M. (2008). A methodology for uml models v&v. In *Software testing, verification, and validation, 2008 1st international conference on* (pp. 513–516).

Bassi, L., Secchi, C., Bonfe, M., & Fantuzzi, C. (2011). A sysml-based methodology for manufacturing machinery modeling and design. *IEEE/ASME Transactions on Mechatronics*, *16*(6), 1049–1062.

Bousse, E., Mentré, D., Combemale, B., Baudry, B., & Katsuragi, T. (2012). Aligning sysml with the b method to provide v&v for systems engineering. In *Proceedings of the workshop on model-driven engineering, verification and validation* (pp. 11–16).

Bryman, A., Bell, E., & Nilsson, B. (2005). *Företagsekonomiska forskningsmetoder*. Liber ekonomi.

Chabrol, M., & Sarramia, D. (2000). Object oriented methodology based on uml for urban traffic system modeling. In *International conference on the unified modeling language* (pp. 425–439).

Chaudron, M. R., Heijstek, W., & Nugroho, A. (2012). How effective is uml modeling? *Software & Systems Modeling*, *11*(4), 571–580.

Clarke, P., & O'Connor, R. V. (2012). The situational factors that affect the software development process: Towards a comprehensive reference framework. *Information and Software Technology*, *54*(5), 433–447.

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

Cruzes, D. S., & Dyba, T. (2011). Recommended steps for thematic synthesis in software engineering. In *Empirical software engineering and measurement (esem), 2011 international symposium on* (pp. 275–284).

Cruzes, D. S., Dybå, T., Runeson, P., & Höst, M. (2015). Case studies synthesis: a thematic, cross-case, and narrative synthesis worked example. *Empirical Software Engineering*, *20*(6), 1634–1665.

Estévez, E., Marcos, M., Sarachaga, I., & Orive, D. (2007). A methodology for multidisciplinary modeling of industrial control systems using uml. In *Industrial informatics, 2007 5th ieee international conference on* (Vol. 1, pp. 171–176).

Forward, A., & Lethbridge, T. C. (2002). The relevance of software documentation, tools and technologies: a survey. In *Proceedings of the 2002 acm symposium on document* *engineering* (pp. 26–33).

Gérard, S., Terrier, F., Voros, N. S., & Koulamas, C. (2001). Efficient system modeling of complex real-time industrial networks using the accord/uml methodology. In *Architecture and design of distributed embedded systems* (pp. 11–21). Springer.

Haselton, M. G., Nettle, D., & Murray, D. R. (2005). The evolution of cognitive bias. *The handbook of evolutionary psychology.*

Hebig, R., & Bendraou, R. (2014). On the need to study the impact of model driven engineering on software processes. In *Proceedings of the 2014 international conference on software and system process* (pp. 164–168).

Hebig, R., Schmied, A. I., & Weisemöller, I. (2016). Lessons learned from co-evolution of software process and model-driven engineering. In *Managing software process evolution* (pp. 257–280). Springer.

Heldal, R., Pelliccione, P., Eliasson, U., Lantz, J., Derehag, J., & Whittle, J. (2016). Descriptive vs prescriptive models in industry. In *Proceedings of the acm/ieee 19th international conference on model driven engineering languages and systems* (pp. 216–226).

Herrera, F., Posadas, H., Penil, P., Villar, E., Ferrero, F., Valencia, R., & Palermo, G. (2014). The complex methodology for uml/marte modeling and design space exploration of embedded systems. *Journal of Systems Architecture*, *60*(1), 55–78.

Herzog, E., Andersson, H., & Hallonquist, J. (2010). 4.5. 3 experience from introducing sysml into a large project organisation. In *Incose international symposium* (Vol. 20, pp. 595–604).

Iqbal, M. Z., Arcuri, A., & Briand, L. (2010). Environment modeling with uml/marte to support black-box system testing for real-time embedded systems: methodology and industrial case studies. In *International conference on model driven engineering languages and systems* (pp. 286–300).

Kalus, G., & Kuhrmann, M. (2013). Criteria for software process tailoring: a systematic review. In *Proceedings of the 2013 international conference on software and system process* (pp. 171–180).

Keele, S. (2007a). Guidelines for performing systematic literature reviews in software engineering. In *Technical report, ver. 2.3 ebse technical report. ebse.* sn.

Keele, S. (2007b). Guidelines for performing systematic literature reviews in software engineering. In *Technical report, ver. 2.3 ebse technical report. ebse.* sn.

Łabiak, G., Adamski, M., Doligalski, M., Tkacz, J., & Bukowiec, A. (2012). Uml modelling in rigorous design methodology for discrete controllers. *International Journal of Electronics and Telecommunications*, *58*(1), 27–34.

Lane, J. A., & Bohn, T. (2013). Using sysml modeling to understand and evolve systems of systems. *Systems Engineering*, *16*(1), 87–98.

Li, L., Wang, N., Ma, L., & Yang, Q. (2011). Modeling method of military aircraft support process based sysml. In *Reliability, maintainability and safety (icrms), 2011 9th international conference on* (pp. 1247–1251).

Liebel, G., Marko, N., Tichy, M., Leitner, A., & Hansson, J. (2016). Model-based engineering in the embedded systems domain: an industrial survey on the state-of-practice. *Software & Systems Modeling*, 1–23. Retrieved from `http://dx.doi.org/10.1007/s10270-016-0523-3` doi: 10.1007/s10270-016-0523-3

Liu, X., Ren, Y., Wang, Z., & Liu, L. (2013). Modeling method of sysml-based reliability block diagram. In *Mechatronic sciences, electric engineering and computer (mec), proceedings 2013 international conference on* (pp. 206–209).

Louise Barriball, K., & While, A. (1994). Collecting data using a semi-structured interview: a discussion paper. *Journal of advanced nursing*, *19*(2), 328–335.

MacCormack, A., & Verganti, R. (2003). Managing the sources of uncertainty: Matching process and context in software development. *Journal of Product Innovation Management*, *20*(3), 217–232.

Maisenbacher, S., Kernschmidt, K., Kasperek, D., Vogel-Heuser, B., & Maurer, M. (2013). Using dsm and mdm methodologies to analyze structural sysml models. In *Industrial engineering and engineering management (ieem), 2013 ieee international conference on* (pp. 351–355).

Mattsson, A., Fitzgerald, B., Lundell, B., & Lings, B. (2012). An approach for modeling architectural design rules in uml and its application to embedded software. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, *21*(2), 10.

Object-Management-Group. (2015a). *Omg systems modeling language (omg sysml™).* http://www.omg.org/spec/SysML/1.4/PDF.

Object-Management-Group. (2015b). *Omg unified modeling language™(omg uml).* http://www.omg.org/spec/UML/2.5/PDF.

Ortiz Zaragoza, F. J., Sala, M., Santos, A., Álvarez Torres, M. B., Iborra García, A. J., Fernández Meroño, J. M., et al. (2002). Modelling a software architecture for robots control using uml and comet architectural design method.

Osmundson, J. S., & Huynh, T. V. (2006). A systems engineering methodology for analyzing systems of systems using the systems modeling language (sysml).

Pelliccione, P., Knauss, E., Heldal, R., Ågren, M., Mallozzi, P., Alminger, A., & Borgentun, D. (2016). *Automotive architecture framework: the experience of volvo cars.*

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

Saldaña, J. (2015). *The coding manual for qualitative researchers.* Sage.

Scanniello, G., Gravino, C., Genero, M., Cruz-Lemus, J., & Tortora, G. (2014). On the impact of uml analysis models on source-code comprehensibility and modifiability. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, *23*(2), 13.

Schönherr, O., & Rose, O. (2010). Important components for modeling production systems with sysml. In *Iie annual conference. proceedings* (p. 1).

Subramanian, G. H., Klein, G., Jiang, J. J., & Chan, C.-L. (2009). Balancing four factors in system development projects. *Communications of the ACM*, *52*(10), 118–121.

Suochang, Y., et al. (2010). Modeling method for information model of fault tree diagnosis based on uml. In *Computer science and information technology (iccsit), 2010 3rd ieee international conference on* (Vol. 7, pp. 238–241).

Voirin, J.-L., Bonnet, S., Normand, V., & Exertier, D. (2015). From initial investigations up to large-scale rollout of an mbse method and its supporting workbench: the thales experience. In *Incose international symposium* (Vol. 25, pp. 325–340).

Whittle, J., Hutchinson, J., & Rouncefield, M. (2014). The state of practice in model-driven engineering. *IEEE software*, *31*(3), 79–85.

Xin, W., Chao, L., Weiren, X., & Ying, L. (2015). A failure time series prediction method based on uml model. In *Computer science and network technology (iccsnt), 2015 4th international conference on* (Vol. 1, pp. 62–70).

Zaki, M. Z. M., & Jawawi, D. N. A. (2011). Model-based methodology for implementing marte in embedded real-time software. In *Computers & informatics (isci), 2011 ieee symposium on* (pp. 536–541).

Zhou, S., Sun, Q., & Jiao, J. (2014). A safety modeling method based on sysml. In *Reliability, maintainability and safety (icrms), 2014 international conference on* (pp.

1180–1185).

Zhu, H. T., Pan, W. L., & Liu, C. (2010). Study of product modeling method based on uml. In *Key engineering materials* (Vol. 419, pp. 765–768).

A

Appendix 1

A.1 Interview Protocol MBSE

Provide an overview of our purpose, our intended uses for the interview data, and the measures you've taken to protect confidentiality and anonymity. Also, discuss and get permission for tape recording or note-taking.

A.1.1 Introduction

- Introduce yourselves and your project
- Ask permission for recording
- Mention your supervisor at SAAB
- Agree on time for the interview
- Explain confidentiality of interview

A.1.2 A few background questions

- What is your job title?
- What are your responsibilities?
- What is your background, what have you been working with?
  - Time within the organization?
  - Modelling

A.1.3 Project Questions

- What project are you currently working on?
  - Can you provide a general overview of the project?
- How long have you worked on the project?
- What is your current position in the project?
  - Tasks / Expertise
  - Do you use models in your current project?
  - Have the models been useful

- Please show
  - System
  - Method
  - What models are used?
  - What models do you like/dislike and why?
  - What is Cumbersome?
  - Tools? Happy with tools?

A.1.4 Current Project Context

- What aspects of your current project do you feel could improve?
  - Why?
  - Could you give provide an example?
- What aspect do you think could improve between different branches in the company?
  - Why?
- What area of the project do you perceive as the most problematic?
  - Why do you perceive it as the most problematic?
  - How would you like to improve it?

A.1.5 Communication (Probes)

- How often do meetings between SAAB branches occur?
- How often do you consult the documentation?
- Explain how design decisions are made.
- How do you think that communication between team members is going?
- How do you think that communication between the different parts of the company is currently?
- How are requirements communicated to the team?
- How well did you understand the project when you started?

A.1.6 Documentation (Probes)

- How has the documentation been done previously?
- How do you feel about the documentation in its current state?

A.1.7 Models (Probes)

- How familiar are you with the concepts of models in software development ?
- Have you used models?

  - If so, which modeling languages have you used?
  - If so, for which purpose(s) have you used modeling ?
- Do you have an opinion on models in sw and system development?
- How should models (not) be used?
- What do you think would be risks or hurdles in using models effectively?
- Are you using code generation?
- What models do you use, which are redundant, what is missing?
- Traceability issues?
- Is it easy to modify the models?
- What was the biggest mistake?

A.1.8 Process (Probes)

- How is the way of working, agile, waterfall?
- How are changes to the process handled?
- How are future plans for the project specified?
- How are future plans for the project communicated?
- Has the modelling process changed during your time in the project?
- What were the changes?
  - How do you feel about the changes?

A.1.9 End questions

- Any questions?
- Is there anyone else you think we should talk to?
- Show framework and ask for feedback