MODEL-BASED SYSTEMS ENGINEERING MATURITY IMPROVEMENT IN INDUSTRY

Inaugural-Dissertation

zur

Erlangung des Doktorgrades

der Mathematisch-Naturwissenschaftlichen Fakultät

der Universität zu Köln

vorgelegt von

Tiago Luiz Buarque de Amorim

aus Recife, Brasilien

Mai 2021

Als Dissertation genehmigt von
der Mathematisch-Naturwissenschaftlichen Fakultät
der Universität zu Köln

1. Gutachter	Prof. Dr. Andreas Vogelsang, Universität zu Köln
2. Gutachter	Prof. Dr. Sjaak Brinkkemper, Universiteit Utrecht

Tag der mündlichen Prüfung: 29. Juli 2021

There is no knowledge
that is not power.

— Rudolph Waldo Emerson

ABSTRACT

Two things about change are valid in any scenario; it will happen and seldomly comes without challenges. That is also the case for organizations that must continuously change to tackle the increasing complexity of systems and market pressure. Model-based Systems Engineering (MBSE) is often suggested as a coping mechanism for the challenges of engineering contemporary complex systems. It advocates the integrated use of models throughout all phases of the system development life-cycle. Although many practitioners agree with the techniques' potential benefits, companies struggle to achieve MBSE adoption in a timely and effective way, diminishing investment returns. Additionally, MBSE comprehensive coverage is a double-edged sword, meaning teams struggle with selecting methods best aligned with their adoption goal, i.e., not all MBSE methods are fit to deliver the pursued benefits.

In this thesis, our contribution is four-fold. First, we perform a study to investigate the forces that prevent or impede the adoption of MBSE in embedded systems companies. We contrast the hindering forces with issues and challenges driving these companies towards introducing MBSE. Our results are based on 20 interviews with experts from 10 companies. Our findings show that forces preventing MBSE adoption relate to immature tooling, uncertainty about the return-on-investment, and fears on migrating existing data and processes. On the other hand, MBSE adoption also has strong drivers, and participants have high expectations mainly concerning managing complexity, easily adhering to new regulations, and reducing costs.

Second, we aim to discover the best practices and strategies to implement MBSE methods in embedded systems companies. Using an inductive-deductive research approach, we conducted 14 semi-structured interviews with experts from 10 companies. Moreover, we analyzed the data and drew conclusions validated by practitioners through an online questionnaire in a triangulation fashion. Our findings are summarized in an empirically validated list of 18 best practices for MBSE adoption and a prioritized list of the six most important best practices.

Third, we propose a method value ascription model to describe elements and relations that play a significant role when a development team considers which methods to adopt. The model comprises the classic cost-benefit appraisal and relevant aspects such as context, adoption goal, and method characteristics. All other thesis contributions have relations to the model elements, thus, unifying the whole work.

Last, we aim to relate goals driving MBSE method adoption and candidate solutions, thus providing suggestions that highest yield the expected benefits. For this means, we propose a goal-benefit model and respective operationalization method. The model relates benefits generated by MBSE methods with adoption goals and organization context. Our approach delivers a prioritized list of MBSE methods which can be transformed into an improvement roadmap. The approach was applied in six case studies in systems development teams located in Germany and Brazil. It was assessed positively by the case study participants, who confirmed that the approach indeed supports goal-oriented MBSE method selection and improvement process. We also provide a sensitivity analysis to validate the goal-benefit model.

We expect that our work brings new perspectives and venues for the MBSE and process improvement research community and helps practitioners in their never-ending quest to keep up with change.

ZUSAMMENFASSUNG

Zwei Dinge in Bezug auf Veränderungen sind in jedem Szenario gültig: Sie werden stattfinden und bringen zwangsläufig Herausforderungen mit sich. Dies gilt auch für Unternehmen und Organisationen, die zum einen die ständig steigende Komplexität von Softwaresystemen beherrschen müssen und zugleich dem Druck des Absatzmarktes unterliegen. Eine Methodik um die ingenieurtechnischen Herausforderungen die im Rahmen der Entwicklung moderner komplexer Softwaresysteme auftreten zu bewältigen, ist das Modellbasierte Systems Engineering (MBSE). MBSE zielt darauf ab formale Modelle durchgängig in alle Phasen des Softwareentwicklungsprozesses zu integrieren. Obwohl die potenziellen Vorteile der MBSE Techniken in der Praxis bekannt sind, haben Unternehmen Schwierigkeiten, MBSE zeitnah und effektiv einzuführen, was sich negativ auf das Kosten-Nutzen-Verhältnis auswirkt. Darüber hinaus ist der flächendeckende Einsatz von MBSE ein zweischneidiges Schwert, da die Teams zum einen mit der Auswahl der richtigen Methoden, die am besten zu ihrem Adoptionsziel passen, kämpfen, zum anderen aber nicht alle MBSE-Methoden geeignet sind, den angestrebten Nutzen zu liefern.

Der Beitrag dieser Arbeit lässt sich in vier Schritte aufteilen. Im ersten Schritt haben wir eine Studie durchgeführt um zu untersuchen welche übergeordneten Einflüsse die Einführung von MBSE in Unternehmen verhindern oder erschweren. Die erschwerenden Einflüsse stehen den Problemen und Herausforderungen in der Entwicklung gegenüber, die die Unternehmen motivieren MBSE zu adoptieren. Unsere Ergebnisse basieren auf 20 Interviews mit Expert:innen aus 10 verschiedenen Unternehmen die eingebettete Systeme entwickeln. Die Ergebnisse zeigen, dass unausgereifte Werkzeuge, Unsicherheiten in Bezug auf das Kosten-Nutzen-Verhältnis der MBSE Einführung und Angst vor der Migration bestehender Daten die Hauptgründe sind, MBSE Techniken nicht einzuführen. Andererseits konnten auch stark treibende Faktoren für die Einführung von MBSE identifiziert werden und die Teilnehmer formulieren hohe Erwartungen, vor allem in Bezug auf das Komplexitätsmanagement, das einfachere Einhalten neuer Regularien und die Reduzierung von Kosten.

Im zweiten Schritt extrahieren und beschreiben wir Strategien und Best Practices um Methoden des MBSE in Unternehmen, die sich auf eingebettete Systeme spezialisiert haben, zu implementieren. Unter Verwendung eines induktiv-deduktiven Forschungsansatzes führten wir 14 semistrukturierte Interviews mit Expert:innen aus 10 Unternehmen durch, analysierten die Daten und zogen Schlussfolgerungen, die wiederum mittels einer Triangulationsmethode von Praktiker:innen in Form eines Online-Fragebogens validiert wurden. Die Ergebnisse sind in einer empirisch validierten Liste von 18 Best Practices für die MBSE-Einführung und einer priorisierten Liste der sex wichtigsten Best Practices zusammengefasst.

Im dritten Schritt schlagen wir ein Modell für die Zuschreibung von Methodenwerten vor, um Elemente und Beziehungen zu beschreiben, die eine wichtige Rolle spielen, wenn ein Entwicklungsteam überlegt, welche Methoden es übernehmen soll. Das Modell umfasst die klassische Kosten-Nutzen-Abwägung und relevante Aspekte wie den Kontext, das Ziel der Einführung und die Eigenschaften der Methode. Alle anderen Beiträge der Dissertation haben Beziehungen zu den Modellelementen und vereinheitlichen so die gesamte Arbeit.

Schlussendlich setzen wir die Ziele, die die Einführung von MBSE-Methoden vorantreiben, mit den Lösungsvorschlägen in Beziehung, um die Vorschläge mit dem höchsten erwartbaren Nutzen zu extrahieren. Um dies zu erreichen präsentieren wir ein Ziel-Nutzen-Modell und eine entsprechende Operationalisierungsmethode. Das Modell setzt den durch MBSE-Methoden generierten Nutzen mit den Adoptionszielen und dem Organisationskontext in Beziehung. Unser Ansatz liefert eine priorisierte Liste von MBSE-Methoden, die in eine Roadmap zur Verbesserung der Entwicklungsprozess umgewandelt werden kann. Um den Ansatz zu evaluieren wurde er im Rahmen von sechs Fallstudien in verschiedenen Entwicklungsteams in Deutschland sowie in Brasilien angewendet. Der Ansatz wurde von allen Teilnehmer:innen durchweg positiv bewertet. Dies bestätigt, dass der entwickelte Ansatz einen gewinnbringenden Beitrag zur zielgerichteten MBSE-Methodenauswahl- und Verbesserung liefert. Zusätzlich wurde der Ansatz in Form einer Sensitivitätsanalyse validiert.

Wir erwarten, dass unsere Arbeit neue Perspektiven für die Wissenschaftscommunity im MBSE- und Prozessverbesserungs-Bereich bringt und Praktiker:innen unterstützt die ständig wandelnden Herausforderungen im dynamischen Umfeld der Softwareentwicklung zu meistern.

ACKNOWLEDGMENTS

First, I want to thank Prof. Dr. Andreas Vogelsang, who had faith in my capabilities when he decided to have me as a Doctor student back in 2017, and Prof. Dr. Sjaak Brinkkemper, who accepted the duty to be the second evaluator of this work.

I also want to thank:

- Prof. Dr. Edna Dias Canedo, whose support allowed me to do an exchange program in Brazil and put me in contact with many industrial partners.
- Dr. Martin Becker, who hired me when I finished my Master's degree back in 2013, kick-starting my career as a researcher.
- Industrial partners and study participants who willingly donate their time for this work.
- Alexandra Elbakyan for creating the Sci-Hub, "the first website in the world to provide mass & public access to research papers", and André Miede for creating the LaTeX template used in this thesis.
- My working colleagues at TU Berlin and Universität zu Köln.

I am thankful to my family, my friends, and my fiancée Blenda.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AHP	Analytic hierarchy process
MBSE	Model-based Systems Engineering
MBSE MM	Model-based Systems Engineering Maturity Model
BDM	Benefit Dependency Map

UML	Unified Modeling Language
OMG	Object Management Group
XP	eXtreme Programming
SE	Systems Engineering
MDD	Moldel-driven Development
MBE	Model-based Engineering
CMMI	Capability Maturity Model Integration

1

INTRODUCTION

The topic of this thesis is Model-based Systems Engineering (MBSE) maturity improvement in industry. In this chapter, we present the reasons why teams are impelled to adopt MBSE methods (Section 1.1), why adopting MBSE is not so trivial, and the value of its investigation (Section 1.2). In Section 1.3, we present the contributions of this thesis towards solving the presented problems. In Section 1.4, we present a summary of the research methods applied when investigating the aforementioned problems. Finally, this thesis outline is described in Section 1.5, and the material published during this thesis work is presented in Section 1.6.

1.1 CONTEXT

Many development teams face problems with the increasing complexity of software-intensive systems, their interdisciplinary development, and the vast amount of mainly text-based specifications. Additionally, as new functionalities are replicated by market competitors and soon become a commodity, teams are pushed to deliver in less time to gain competitive advantage, and nevertheless with top quality. These systems are usually safety-critical, and a small misshapen can compromise the whole organization's image. All this needs to be addressed at a global-dictated market-compatible cost which shrinks at every new development cycle. For instance, an European car manufacturer builds products having over 200 control units, delivering 400 functionalities (*e. g.,* breaking, accelerating, pedestrian collision mitigation, audio, GPS), built-in 2.000 software components working with 10.000 signals. These are added 30.000 mechanical parts allowing a combination of 1.000.000 different types of vehicles (*i. e.,* customization) which can be selected on their website (Batejat, 2019).

A solution to cope with the problems mentioned above is Model-based Systems Engineering (MBSE), whose methods are applied in the conceptual design, continues throughout development and later life cycle phases (INCOSE, 2007). MBSE is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities. In this approach, models (as opposed to documents) serve as blueprints for developers to write code, provide formalization, tackle complexity, and enhance the system's understanding. Specialized tools automate much of the non-creative work (which translates to gains in productivity and quality) and generate artifacts (*e.g.*, code) based on the models. MBSE foster artifact reuse, improves product quality, and shortens time to market (Carroll and Malins, 2016).

However, MBSE is complex (as the problem types it proposes to solve), making teams struggle to sow the benefits of the newly adopted methods. Additionally, teams are resistant to change, and managers are not sure which methods to adopt.

1.1.1 *Model-based Systems Engineering*

Model-based Systems Engineering (MBSE) is a term that predicates the use of modeling to analyze and document critical aspects of systems engineering. It is broad in scope, spanning the whole development life-cycle, and covers levels from system-of-systems to individual components. MBSE is a model-centric approach providing a single point of truth reflected in a set of living artifacts.

Compared to document-centric development, MBSE supports engineers with automation capabilities (*e.g.*, code generation, document derivation) and enhanced analysis capabilities (*e.g.*, behavioral analysis, performance analysis, simulation). The use of models (and associated tools) brings benefits for product quality and process quality (Chaudron, Heijstek, and Nugroho, 2012).

UML and SysML are standardized graphical modeling languages for MBSE to define different models, processes, procedures, and operations. While UML is predominantly used for software development, SysML also encompasses the physical aspects of a system. Prominent tool vendors, such as IBM, Oracle, Microsoft, or the Eclipse Foundation offer tooling solutions for MBSE. More information on MBSE is provided in Section 2.2.

1.2 PROBLEM STATEMENT

An increasing number of software-intensive cyber-physical systems development teams recognize the complexity of integrating components of heterogeneous domains, disciplines, and vendors (Vogelsang et al., 2017). Additionally, fierce market competition pushes teams to reinvent their processes to, at least, keep up with the never-ending increasing demands.

MBSE has been proposed as a solution for aligning the engineering process of these systems by using well-defined and agreed-on models consistently through all development phases and domains. Although many practitioners agree with the arguments on MBSE methods' potential benefits, companies struggle with their adoption. Process changes are required in all system life-cycle phases, including a shift in the development paradigm (i.e., abstract thinking (Hutchinson, Whittle, and Rouncefield, 2014)) and new tools application. Projects are not likely to meet their cost and delivery targets when adoption is carried out poorly.

In any case, introducing process change in an already running and established development team is not friction-less (Conner, 1993; Hammer, 2007). Micromanagement is required to tackle hurdles and psychological barriers, and this becomes more critical when it comes to such a complex change as MBSE adoption (Vogelsang et al., 2017). However, other methodologies, such as agile practices, have been adopted much faster. So, what are the reasons and factors that prevent or impede companies from adopting MBSE?

PROBLEM STATEMENT 1: We need to understand how to efficiently introduce new MBSE methods in development teams.

Development teams adopt MBSE methods for many reasons. Sometimes their goal is to enhance the product's properties (*e.g.*, quality, desirable functionalities). Perhaps they seek to increase the development process efficiency (*e.g.*, cost, lead time). Others might need to tackle problems related to systems' complexity (*e.g.*, interdisciplinary development, the vast amount of text-based specifications) (Vogelsang et al., 2017).

MBSE offers methods for the whole system development life-cycle. Specific MBSE methods and techniques may not be suitable in all situations, and with so many possibilities, it is hard to know which methods to adopt. Managers might be tempted to go for a *low-hanging fruit* strategy, which is cheap, easy, and has high implementation success rate. However, considering the adoption goal, will the team harvest the expected benefits? If the answer is no, the initiative might feel like a waste of time (*i.e.*, new processes but same issues). Thus, there is a need to assess which MBSE methods should be implemented according to the team's context, and considering associated costs and benefits (Chaudron, Heijstek, and Nugroho, 2012).

The alignment importance with the team's adoption goal is emphasized in a report released by the Project Management Institute (PMI)[1], projects and programs that are aligned with a team's strategy are completed successfully more often than projects that are misaligned (77% vs. 56%). At the same time, only 60% of strategic initiatives meet

---

[1] https://www.pmi.org/

their original goals and business intent. The report states that most executives admit a disconnection between strategy formulation and implementation (PMI, 2017).

The Model-based Systems Engineering Maturity Model (MBSE MM) is a possible solution for the aforementioned issue. The MBSE MM is a descriptive and prescriptive (Pöppelbuß and Röglinger, 2011) focus area maturity model (van Steenbergen et al., 2010, 2013). It is descriptive because it can assess and describe the team's current MBSE capabilities. It is prescriptive because it provides a maturity path with requirements; thus, one can see the possible next possible capabilities to be adopted. Additionally, the complexity and pervasiveness of MBSE are reflected in the six engineering functions and fifteen focus areas of the MBSE MM. However, the MBSE MM does not provide any systematic method to select improvement measures *i. e.,* it lacks built-in contingency to guide tailoring (Conboy and Fitzgerald, 2010) such as a decision calculus (Little, 1970; Peterson, 2017; Pöppelbuß and Röglinger, 2011) After assessing the maturity profile of a team, there are still many possibilities to progress; therefore, the problem is only partially addressed.

PROBLEM STATEMENT 2: We need a method to provide decision support when selecting the MBSE methods that are most appropriate for each team considering respective contextual characteristics and adoption goal.

1.3 CONTRIBUTIONS OF THIS THESIS

This thesis goal can be summarized into investigating MBSE adoption by development teams. We combined the specific study goals from each chapter with the following three goals, which we address in this thesis.

g1: Understanding the forces that drive MBSE adoption.

g2: Identifying strategies and best practices to increase the success rate of MBSE adoption.

g3: Designing and evaluating an approach for selecting MBSE methods that yield the maximum benefit for a team according to adoption goals and contextual characteristics.

In the following two subsections, we describe the contributions of this thesis in detail. We differentiate between two types of contributions, namely *Major contributions* and *Minor contributions*. The former is strictly related to the main goals driving the research, whilst the latter are contributions required to achieve the former, but they do not have enough substance to be categorized as the major contributions. In this thesis, we provide the following solutions for the stated problems:

1.3.1 *Major contributions*

In this subsection, we describe the major contributions of this thesis.

M1: FORCES THAT DRIVE OR PREVENT MODEL-BASED SYSTEMS ENGINEERING ADOPTION IN THE EMBEDDED SYSTEMS INDUSTRY. We investigate the forces that prevent or impede the adoption of MBSE in companies that develop embedded software systems. We contrast these hindering forces with issues and challenges that drive these companies towards introducing MBSE. Our results are based on 20 interviews with experts from 10 organizations in Germany. We analyze the results utilizing thematic coding and categorize the identified forces into inertia and anxiety forces, which prevent MBSE adoption, as well as push and pull forces, which drive the companies towards MBSE adoption. We frame the results with the coding of what the interviewees considered as MBSE. Forces that prevent MBSE adoption relate mainly to immature tooling, uncertainty about the return-on-investment, and fears on migrating existing data and processes. On the other hand, MBSE adoption also has strong drivers, and participants have high expectations mainly concerning managing complexity, adhering to new regulations, and detecting bugs earlier. We observed that the hindering forces are much more concrete and MBSE-specific compared with the fostering forces, which are very generic (e.g., increase in product quality, managing complexity, supporting reuse). From this, we conclude that bad experiences and frustration about MBSE adoption originate from misleading or too optimistic expectations. Nevertheless, companies should not underestimate the necessary efforts for convincing employees and addressing their anxiety. This major contribution is presented in Chapter 3.

M2: STRATEGIES AND BEST PRACTICES FOR MODEL-BASED SYSTEMS ENGINEERING ADOPTION IN EMBEDDED SYSTEMS INDUSTRY. Our goal was to find out what was tried, what worked, what did not work, how the problems were solved, what can be recommended, and what should be avoided when adopting MBSE in organizations that develop embedded systems. For this purpose, we conducted 14 semi-structured interviews with experts from embedded systems organizations. From these interviews, we extracted 18 best practices fitted for tackling MBSE adoption challenges. Sequentially, we validated and prioritized the best practices with an online questionnaire answered by MBSE practitioners. Our findings provide input for planning MBSE adoption based on the knowledge of practitioners that went through the experience of implementing MBSE in already established embedded systems development organizations. This contribution is presented in Chapter 4.

M3: A METHOD VALUE ASCRIPTION MODEL. We propose a model for method value ascription. MBSE encompasses many methods, and deciding which ones are the most interesting for a team requires proper value appraisal. Thus, understanding the mechanics of method value ascription helps to deliver better decision-making for an effective MBSE adoption. Additionally, we relate the other contributions of this thesis to the model elements. This contribution is presented in Chapter 5.

M4: A GOAL-BENEFIT DECISION CALCULUS APPROACH TO SUPPORT MBSE METHODS SELECTION. We propose a decision calculus approach to prioritize candidate capabilities from the MBSE MM according to the development team adoption goal and context. The approach is composed of a model and an instantiation method. The model breaks down the adoption goal into so-called competitive priorities related to benefits afforded by capabilities upon their implementation. The instantiation method guides the weight assignment of the model relations according to the relevance of the elements towards their neighbors. According to the assigned weights, points are distributed along the resulting graph, from the adoption goal to the competitive priorities, benefits, and candidate capabilities. As a result, candidate capabilities are prioritized according to the overall points they collect, reflecting their relevance towards the adoption goal and the team's context. The approach was applied in six development teams in Brazil and Germany, whose members evaluated the approach helpful. This contribution can be broken down into:

- A goal-benefit decision calculus approach to support MBSE methods selection. The approach is composed of:
  - A goal-benefit meta-model that relates MBSE methods with benefits, competitive priorities, and finally with the adoption goal.
  - A method for instantiating the meta-model and assigning quantitative impact measures to the model's relations, which is used to derive a prioritized list of MBSE process improvement candidates.

- Qualitative and quantitative evaluations of the approach based on six case studies and an analysis of the model.

We want to support teams when choosing which MBSE techniques to adopt with this approach. This contribution is described in detail in Chapter 6.

1.3.2 *Minor contributions*

This subsection describes contributions that support the main contributions but could be considered less relevant or novel.

M1: ADAPTATION OF THE FORCES MODEL TO MBSE METHOD ADOPTION. The forces model (Klement, 2018) was initially devised to describe the forces that affect buyer decision towards new products. These are forces to have in mind when devising and promoting products. We extended it and use it to describe the forces towards method adoption, in special MBSE adoption. The re-purposing of this framework in Chapter 3 is a minor contribution of this thesis.

M2: ELICITATION OF MODEL-BASED SYSTEMS ENGINEERING BENEFITS. As part of the design of the MBSE goal-benefit meta-model mentioned in the previous subsection, we needed to investigate in the literature the tangible benefits that a team would enjoy upon the MBSE method adoption. As a result, we elicited ten benefits which are thoroughly described in Chapter 2.2.2.

M3: EXPERT ASSESSMENT OF MBSE MM CAPABILITIES AFFORDANCES. As part of the approach creation from Chapter 6, we asked five MBSE experts to assess the relationship between the capabilities from the MBSE MM and the elicited MBSE benefits. The results can be seen in Table 6.3 This assessment enriches the MBSE MM, which can be bundle up with more information. Also, it sheds light on the capabilities that can yield the highest benefits in general.

1.4 RESEARCH METHODOLOGY

The presented work was performed with the industry's stark collaboration since our research questions explored gaps relevant to practitioners. During our research, we employed some methods for acquiring empirical evidence. These methods are briefly described in the following and are described in depth in the respective chapters. A summary of methods and subjects can be seen in Table 1.1.

1.4.1 *Interviews*

Interviews were used as the primary source of empirical evidence in major contributions *M1: Forces that drive and prevent MBSE adoption in the embedded systems industry* (cf. Chapter 3) and *M2: Strategies and best practices for model-based systems engineering adoption in embedded systems industry.* (cf. Chapter 4). The method provides insights into the examined topic and gives essential information to understand the phenomenon in its real context (Dresch, Lacerda, and Antunes, 2015; Runeson and Höst, 2008). In both cases, we had an inductive and exploratory goal (Shields and Rangarjan, 2013). The interviews were conducted in a semi-structured fashion using a guide structured along a funnel model (Bryman, 2015; Runeson and Höst, 2008). They started with general questions about the participant's context, understanding of MBSE concepts, and afterward going into detail about specific topics such as employee training, MBSE integration, or experiences in the past.

The interviews were analyzed using qualitative coding (Neuman, 2010) and managed using the qualitative data analysis tool ATLAS.ti[2]. The analysis started with the involved researchers working on the same small set of interviews. The results were discussed and merged in a meeting to homogenize the codes understanding (Weston et al., 2001) (i.e., what/how to look for). The remaining interviews were tackled in a cross-analysis fashion.

1.4.2 *Surveys*

We used surveys in major contributions *M2: Strategies and best practices for Model-based systems engineering adoption in embedded systems industry.* (cf. Chapter 4) and *M4: A goal-benefit decision calculus approach to support MBSE methods selection* (cf. Chapter 6). They are shortly described in the following paragraphs.

For contribution *M2: Strategies and best practices for Model-based systems engineering adoption in embedded systems industry.*, we surveyed MBSE practitioners to validate the elicited best practices in a deductive triangulation fashion. In the survey, we asked subjects to evaluate their agreement towards the best practices. The survey was conducted

---

[2] `http://atlasti.com`

using Google Forms[3] and the respondents were screened from two research projects mailing list, CrESt[4] and SPEDiT[5], and two MBSE discussion groups from a business and employment-oriented social network[6]. Our anonymous questionnaire had three parts. In the first part, we asked for demographic data (team size, industry sector) and added a yes/no question whether the respondent has ever participated or observed some endeavor to introduce Model-based Engineering in a team. We used this question to exclude respondents without proper experience. In the second part, we asked the respondents to express their agreement with each best practice using a Likert scale (Likert, 1932) with four levels: *Strongly Disagree, Disagree, Agree, Strongly Agree*. Lastly, we asked the respondents to select up to five best practices they considered most important. More details about this survey is disclosed in Chapter 4.

For contribution *M4: A goal-benefit decision calculus approach to support MBSE methods selection*, we performed a survey with MBSE specialists to assess which benefits adopted MBSE capabilities could afford. In a single-round survey (Pan et al., 1996) fashion, five MBSE experts were sent through e-mail a spreadsheet file containing a capability for each row and benefits for columns. They were asked to state whether the capability could produce the benefit and to which intensity. We collected their answers and aggregated the values using the median. More details about this survey can be seen in Chapter 6.

1.4.3 *Case studies*

We conduct six case studies as part of contribution *M4: A goal-benefit decision calculus approach to support MBSE methods selection* (cf. Chapter 6) to corroborate the fulfillment of the proposed approach design requirements. In these case studies, we applied the approach in development teams and collected their feedback about the delivered results and the approach itself. The case studies were carried out in Brazil and Germany and were selected from the author's research colleagues' contacts. The participants were debriefed over the model and the method in a pitch meeting to present the approach. Once all tasks were performed, we held a meeting to present the results and collect the participants' impressions of the approach.

---

[3] `https://www.google.com/forms/about/`
[4] `https://crest.in.tum.de/`
[5] `https://spedit.in.tum.de/`
[6] `https://www.linkedin.com/`

Table 1.1: Data collection methods used in this thesis

Chapter	Methods	Data Sources
Chapter 3	Stakeholder Interviews	20 participants
Chapter 4	Stakeholder Interviews	14 participants
	Survey	40 participants
Chapter 6	Case study	6 development teams
	Survey	5 participants

1.4.4 *Categorization of research methods*

We classify our studies according to dimensions proposed by (Runeson and Höst, 2008) and (Stol and Fitzgerald, 2015), which aim to classify research methods in software engineering and are explained in the following:

- **Obtrusiveness (O)**
  This dimension describes the researchers control level within a study setting (*e. g.*, variables, confounding factors).

- **Generalization (G)**
  This dimension describes the results' applicability, either *"universal"* or *"particular"*.

- **Research setting (RS)**
  This dimension classifies the study environment regarding its realness. There are four categories to choose from, *"natural"* for field studies, *"contrived"* for a more artificial, somewhat a research lab environment, *"no empirical setting"* (*e. g.*, derivation of a conceptual framework from literature), and *"setting independent"* which is fit for surveys with large sample size.

- **Focus (F)**
  This dimension put emphasis on the most generalizable feature of the study and it is divided in three categories, 'actor' (*e. g.*, surveys), *"context"* (*e. g.*, field studies), and *"behavior"* (*e. g.*, experimental studies).

- **Purpose (P)**
  The study purpose can be classified in four types, *"exploratory"*, *"descriptive"*, *"explanatory"*, and *"improving"*.

The studies performed in this thesis can be classified in the presented framework as described in Table 1.2.

Table 1.2: Categorization of the studies included in this thesis

	(O)	(G)	(RS)	(F)	(P)
Chapter 3					
Interviews	unobtrusive	particular	natural	behavior	exploratory
Chapter 4					
Interviews	unobtrusive	universal	natural	behavior	improving
Survey	unobtrusive	universal	natural	behavior	descriptive
Chapter 6					
Case study	unobtrusive	particular	natural	context	explanatory
Survey	unobtrusive	particular	natural	context	descriptive

1.5 OUTLINE

This thesis is organized as explained in the following:

Chapter 2 (Foundational theory) presents the knowledge needed to understand concepts and terminologies used throughout this thesis. We explain the required concepts of MBSE, give an introduction on Focus Area Maturity Models, and explain the MBSE MM.

Chapter 3 (Forces that drive or prevent Model-based Systems Engineering adoption in the embedded systems industry) presents an empirical study on the societal forces that hinder or foster MBSE adoption. The study was based on 20 interviews made with practitioners and seasoned researchers.

Chapter 4 (Strategies and best practices for Model-based Systems Engineering adoption in embedded systems industry) presents 18 best practices that practitioners should adhere when adopting MBSE for increased efficiency and efficacy.

Chapter 5 (Method value ascription) introduces a model to described how teams ascribe value to adoption candidate methods according to adoption goal, methods intrinsic characteristics, context and associated costs.

Chapter 6 (Goal-benefit decision calculus for MBSE method selection) introduces a goal-benefit decision calculus method for selecting MBSE methods based on the development team adoption goal and context.

Chapter 7 (State of the art) describes the current state of the art structured along with the contributions of this thesis considering empirical work on MBSE adoption, context modeling, goal modeling, method selection, and Benefits Management.

Chapter 8 (Conclusions and outlook) summarizes this thesis by presenting its contributions, limitations and directions for future work.

1.6 PUBLICATIONS

The work presented in this dissertation was developed between May 2017 and July 2020 at the Technical University of Berlin (Daimler Center for Automotive Information Technology Innovations) and between August 2020 and May 2021 at the University of Cologne. In this section, we present the work published during this time.

1.6.1 *Core publications*

Some ideas and representations have appeared in the publications listed below:

- T. Amorim et al. (2019). “Strategies and Best Practices for Model-based Systems Engineering Adoption in Embedded Systems Industry.” In: *Proceedings of the 41st ACM/IEEE International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP’19)*

  My contributions: As the first author, I wrote the whole paper text, performed the interviews thematic coding together with two other authors, elicited the best practices from the codes and verbatim, and designed, conducted, and analyzed the survey data. This paper was republished later in another conference: T. Amorim et al. (2020). “Strategies and Best Practices for MBSE Adoption in Embedded Systems Industry.” In: *Proceedings of the Software Engineering 2020, Fachtagung des GI-Fachbereichs Softwaretechnik*

- A. Vogelsang et al. (2017). “Should I Stay or Should I Go? On Forces that Drive and Prevent MBSE Adoption in the Embedded Systems Industry.” In: *Proceedings of the 18th International Conference of Product-Focused Software Process Improvement (PROFES'17)*

  My contributions: As the second author, I performed the interviews thematic coding with two other authors and wrote a significant part of the text, contributing with ideas for the research methodology (*e.g.*, cross-analysis of the interviews) and data representation, such as diagrams and tables.

1.6.2 *Additional publications*

Following publications were developed during the aforementioned period, but their topic differs from this thesis topic.

- A. Boll et al. (2021). “Characteristics, Potentials, and Limitations of Open Source Simulink Projects for Empirical Research.” In: *Software and Systems Modeling*

- H. Martin et al. (2020). “Combined automotive safety and security pattern engineering approach.” In: *Reliability Engineering and System Safety* 198

- T. Amorim et al. (2017). “Systematic Pattern Approach for Safety and Security Co-engineering in the Automotive Domain.” In: *Proceedings of the 36th International Conference on Computer Safety, Reliability, and Security (SAFECOMP'17)*

1.6.3 *Supervised works*

As part of this work, the following master theses were supervised by the author and successfully completed:

- D. Büch, *Unternehmensorientierte Optimierung des Entwicklungsprozesses anhand eines Reifegradmodells fur Modell-basiertes Systems Engineering* (Company-oriented optimization of the development process using a maturity model for model-based systems engineering), Technische Universität Berlin, 2018

- K. Miroshnichenko, *Erweiterung eines MBSE Reifegradmodells um Methoden zur Modellierung von Produktlinien* (Extension of an MBSE maturity model to include methods for modeling of product lines), Technische Universität Berlin, 2018

2

FOUNDATIONAL THEORY

This chapter presents the knowledge required to understand the work presented in this thesis. In Section 2.1 we clarify terms that can have many interpretations. In Section 2.2, we define MBSE, its characteristics, and shed light on important aspects. In Section 2.3 we explain what a maturity model is, discourse about different types and their characteristics. In Section 2.4 we introduce the MBSE MM, which is part of the approach presented in Chapter 6.

2.1 TERMINOLOGY

In this section, we clarify some terms used throughout this thesis.

ORGANIZATION companies, governments, non-governmental organizations, or development teams.

DEVELOPMENT TEAM any systems or software development team. This term can also be referred to as only *team*.

PROCESS is a series of actions undertaken to achieve a result (Cambridge University Press, 2020). In the context of this thesis, we use the term process as in the development process, which is the set of steps required to develop software or systems. A process encompasses methods that are applied in tasks. A process can contain sub-processes. For example Verification and Validation process is contained in the System development process.

PROCESS IMPROVEMENT is the modification of the development team's current process to achieve more efficiency and effectiveness. Improvement can be sometimes hard to measure locally, some processes are labor-intensive, and at first, one could think that there was no improvement, but this extra effort pays off in other phases of development (*i. e.,* extra analysis on early phases represents more effort which compensates for less effort with design issues further on).

MATURITY indicates the degree of development from an initial state to a more skillful and capable state (Mettler and Winter, 2010). It defines how complex a process is in process maturity, described on a maturity model, in an ordinal capability scale.

MATURITY IMPROVEMENT is the adoption of methods that afford a development team with more sophisticated capabilities.

METHOD are techniques and procedures applied in the execution of processes. An approach performing a software/systems development project, based on a specific way of thinking, consisting of guidelines, rules, and heuristics, structured systematically in terms of development activities, with related development work products and developer roles (played by humans or automated tools) (Brinkkemper, Saeki, and Harmsen, 2001).

Method adoption is the adoption of a method by a development team to improve its processes. Legacy methods with similar objectives are dropped.

CAPABILITY is the ability to achieve a specific goal, making use of the available resources (Bharadwaj, 2000).

MODEL is the description or specification of a system and its respective environment. A model is created to represent a specific category of information (Miller and Mukerji, 2003) and is a simplification of reality (Booch, Rumbaugh, and Jacobson, 2005). A model can be textual (*e.g.,* programming language), purely graphical (*e.g.,* electronic diagram) or have both elements (*e.g.,* UML diagram). The models considered in this thesis are composed of drawings and text, which can be in natural language or use a modeling language.

BENEFIT is a value created as a result of the successful completion of a project (PMI, 2017) (e.g., process adoption). A benefit is: (1) always positive, (2) the goal of method adoption, and (3) contributes to meeting at least one competitive priority. The benefits of MBSE adoption are described further (cf. Section 2.2.2) and plays a pivotal role in our approach (cf. Chapter 6).

ADOPTION GOAL is an envisioned development team's future state achieved through method adoption. It can be about changing (e.g., developing new lines of business, exploring new markets) or maintaining its current position relative to its market and competition (OMG, 2010).

2.2 MODEL-BASED SYSTEMS ENGINEERING

Model-based Systems Engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases (INCOSE, 2007). MBSE is a term that predicates modeling to analyze and document critical aspects of the systems engineering lifecycle. It is broad in scope, spanning the SE lifecycle, and covers levels from system-of-systems to individual components. MBSE is a model-centric approach providing a single point of truth reflected in a set of living artifacts.

In traditional document-centric development, systems engineers focus on the development of textual specifications and design documentation. MBSE fosters a coherent system model development containing requirements, design, analysis, and verification information. The model serves as a *single-source-of-truth* and is the primary artifact produced by systems engineering tasks. Using the same model elements, the system model holds discipline-specific views of the system (*e. g.,* system behavior, software, hardware, safety, security) and creates a common standards-based approach to design that can be programmatically validated to remove inconsistencies. This feature improves system analysis reducing the number of defect types commonly injected in a traditional document-based approach. Additionally, the system model provides consistent propagation of corrections and incorporation of new information and design decisions. Documentation, when needed, is generated from the system model (Kaslow et al., 2017).

The use of models to support systems engineering started with the *Model Driven Architecture*, which encompasses a set of Object Management Group (OMG) standards. Later, the term Moldel-driven Development (MDD) was coined for describing an approach for software development that includes standards and methods to make the development process faster, less prone to errors, and easier to understand. Further, the *Model-driven Engineering (MDE)* was created to include engineering steps into the MDD. Finally, the *Model-based Engineering (MBE)* was created to define any engineering performed using models.

The difference between model-based and model-driven is that artifacts are generated from models (*i. e.,* like an automated step, implies

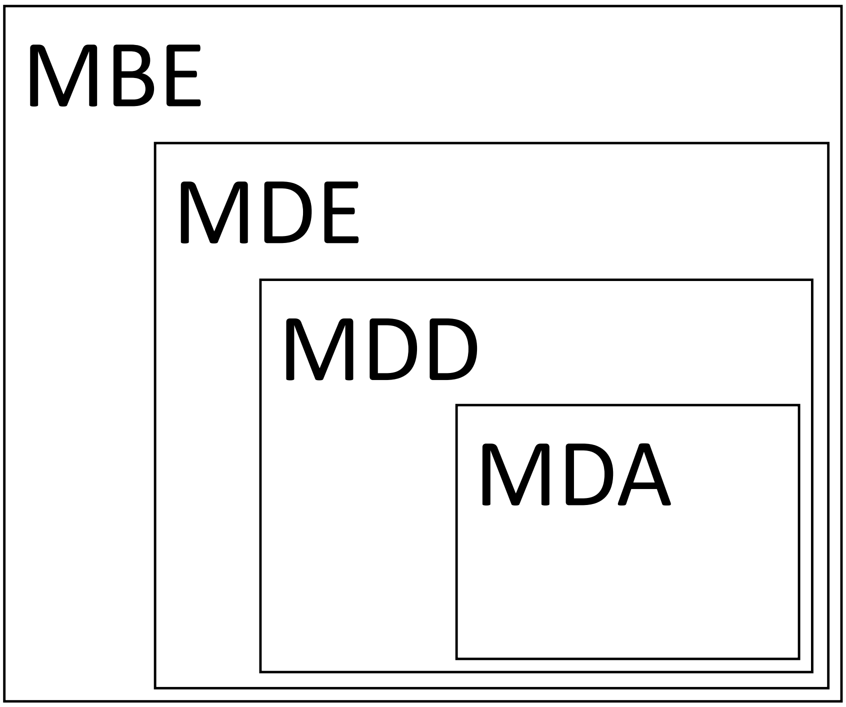


Figure 2.1: Representation of modeling initiatives (Ameller, 2009).

automation) in the latter. In the former, models can also be used as blueprints for developing the system. For instance, in model-driven testing, the tests are generated from models. Thus, Model-based Engineering (MBE) subsumes all other terms (cf. Figure 2.1).

Additionally, MBSE supports engineers with automation capabilities (e.g., code generation, document derivation) and enhanced analysis capabilities (e.g., behavioral analysis, performance analysis, simulation). MBSE is widely used in some application domains as an integral part of development (Bone and Cloutier, 2010). Prominent tool vendors, such as Matlab, IBM, Oracle, Microsoft, or the Eclipse Foundation, offer tooling solutions for MBSE.

2.2.1 *MBSE modeling languages*

MBSE encompasses many modeling languages, being UML, SysML, and Simulink the most popular ones. In the following paragraphs, we describe their most significant characteristics.

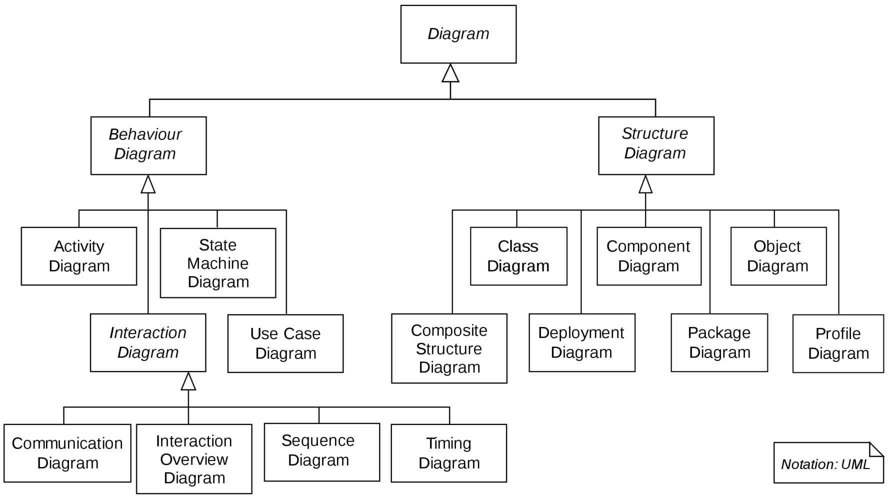


Figure 2.2: Hierarchy of UML 2.2 diagrams.

THE UNIFIED MODELING LANGUAGE is a graphical modeling language for designing software. It may be used to visualize, specify, construct, and document the artifacts of software-intensive systems (Booch, Rumbaugh, and Jacobson, 2005). Unified Modeling Language (UML) is appropriate for modeling different types of systems, ranging from enterprise information systems to distributed Web-based applications and even to hard real-time embedded systems. It addresses the views needed to develop and then deploy systems. The UML diagrams are divided into two types, namely structural and behavioral. They are depicted in Figure 2.2

SYSML (short for "Systems Modeling Language") is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities (Fosse and Bayer, 2016). It is a profile of UML 2 (*i. e.,* a generic extension mechanism for customizing UML models for particular domains and platforms). It was developed by the SysML Partners' SysML Open Source Specification Project in 2003 and was adapted and adopted by the OMG in 2006. While UML is predominantly used for software development, SysML also encompasses physical aspects of a system and allows the integration of heterogeneous domains in a unified model at a high abstraction level. The languages' graphical models are intended to cover all development phases of a system. At the writing of this thesis, the SysML version is 1.6.

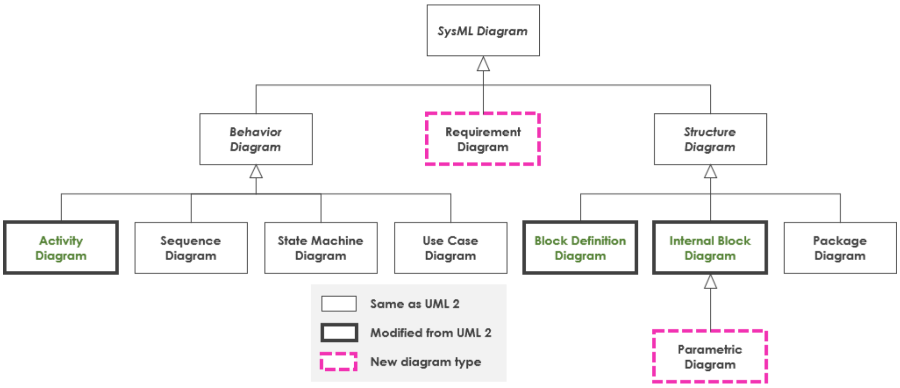


Figure 2.3: SysML diagram types, shown as a class diagram (*MBSE and SysML*).

SIMULINK is a Matlab-based graphical programming environment for modeling, simulating, and analyzing multi-domain dynamical systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. Different kinds of blocks can be connected via ports to transmit outputs and receive inputs, yielding a data flow-oriented model. Subsystems are special blocks that contain another Simulink diagram, thus enabling hierarchical modeling. Figure 2.4 shows an example of a Simulink diagram (taken from Minh, Moustafa, and Tamre, 2017). The model shows a dual-clutch control of an automatic transmission system of a vehicle with two separate clutches. Blocks of various types are connected via signal lines. The four smaller blocks on the left side are inport blocks, transporting input values from the model's context. One of them is the car's current speed (*VehSpd*), which is further processed to compute the next gear shift. Also, three outport blocks (same symbol as inports but with incoming signal lines) transport output values of the model to its context. The four rectangular blocks shaded in gray are subsystems. The subsystems are part of the model, and the contained behavior can be displayed on request. The other shapes represent basic blocks (i.e., non-composite blocks). The pentagon at the top (*trq_dem*) is a *goto* block that transports its signal to some other part of the model (to a point deeper in one of the subsystems). The triangle (*Tmax*) is a *gain* block, which multiplies a signal with a constant. The black bar is a multiplexer block, which combines inputs with the same data type and complexity into a vector output. The rectangle with the label "[0,1]" is a saturation block, which produces an output signal with the input signal's value bounded to some upper and lower values.

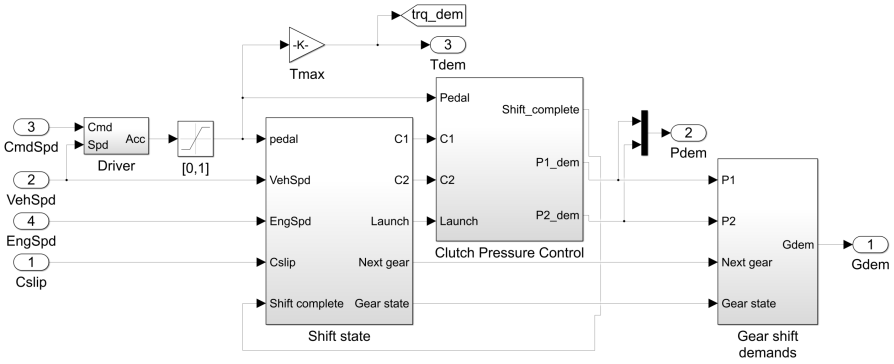


Figure 2.4: Example of a Simulink block diagram modeling the dual-clutch control of an automatic transmission system of a vehicle with two separate clutches (Minh, Moustafa, and Tamre, 2017).

2.2.2 *MBSE benefits*

MBSE is vast; thus, one can speak of many benefits that can be harvested upon the adoption of its methods. Nevertheless, at the same time, not all methods deliver the same benefits. Thus we see as a fundamental part of understanding MBSE also understanding the perceived associated benefits, and in a second moment, associate methods to these benefits. For instance, *Easier handling of complexity* (Asan, Albrecht, and Bilgen, 2014) is a benefit generated by MBSE models and tools that support impact, coverage, and consistency analysis. The benefits associated with MBSE adoption considered in this thesis were gathered through an extensive MBSE literature survey. The collection and presentation of this information is a minor contribution of this thesis and are listed below:

- *Easier reuse of existing artifacts (Salimi and Salimi, 2017):* Models are known to improve reuse. Within the MBSE paradigm, functions are encapsulated, have well-defined interfaces, and are traced to diagrams that explain their usage. Reusing good quality artifacts improves the system's quality by diminishing the code subject to failure.

- *Better communication (Hutchinson et al., 2011):* Models enhance stakeholders understanding. The many types of diagrams used in MBSE represent different views of the systems and are less ambiguous than the natural language used in document-based artifacts. The models from different engineering areas are linked, providing a better overview of the whole system. These characteristics improve communication among stakeholders.

- *Better understanding of problem domain (Boehm, Gray, and Seewaldt, 1984):* Using models, engineers can better grasp the problem at hand, which leads to less "gold plating", fewer missing features, fewer validation issues, and requirements change.

- *Better understanding of solution space (Harvey and Liddy, 2014):* Models can represent more information to the eyes at once compared to document-based specification, which requires thorough skimming. Thus domains that developed reliable modeling options did not go back to document-based (e.g., circuit boards). Product functions can be better selected, become market competitive, and are eventually copied by competitors.

- *Better estimates (cost, the impact of req. changes) (Dabkowski et al., 2013):* Compared to the document-centric approach, MBSE shifts part of the effort from development to design tasks; thus, engineers get a good grasp of systems at early stages of the lifecycle. This benefit allows more precise estimates (i.e., effort estimation deviation gets lower while the deadline meeting rate gets higher). Additionally, MBSE eases integration which contributes to adherence to the estimates.

- *Less defects (McConnell, 2004):* Employing MBSE can diminish defects in various ways: test cases can be automatically generated, defects due to inconsistency and test coverage are tackled by traceability of elements, and reuse guarantees defect-free code.

- *Easier handling of complexity (Asan, Albrecht, and Bilgen, 2014):* Current systems are built with more control units than the previous generation. Additionally, aggressive competition requires engineers to combine functionalities to create new features, which adds to the complexity (Vogelsang, 2019). MBSE addresses this issue through traceability which allows automated checks, saves information-seek effort, and decreases inconsistencies between artifacts.

- *Improved verification & validation (McConnell, 2004):* MBSE models can represent different views of a system, thus helping to communicate better what is understood from stakeholders and the environment. The traceability between the models enables automatic verification. Validation is enhanced with better ways to represent the requirements as well as simulations.

- *Improved quality of specification (Fosse and Bayer, 2016):* This benefit is achieved through specialized tools that can perform many types of analysis in models, *i. e.*, incomplete and inconsistent requirements can be automatically detected. Additionally, the myriad of views provided by MBSE supports and amplifies human thinking, reasoning, and cooperation (Bubenko and Kirikova, 1999).

- *Efficient certification (Helle, 2012):* Models ease the process of generating (e.g., simulations), gathering (i.e., traceability) and reusing evidence. The generation of documents for this means can be automated.

2.3 MATURITY MODELS

Maturity models are used to assess the process maturity of a development team and to guide further improvement. Two approaches for implementing maturity models exist.

With a top-down approach, a fixed number of maturity stages or levels is specified first and further corroborated with characteristics (typically in the form of specific assessment items) that support the initial assumptions about how maturity evolves. Prominent examples of are CMMI (Kneuper, 2008) or SPICE (ISO/IEC, 2003).

When using a bottom-up approach, distinct characteristics or assessment items are determined first and clustered in a second step into maturity levels to induce a more general view of the different steps of maturity evolution. They are distinguished from fixed-level maturity models, such as CMMI, in that they are suited to the incremental improvement of functional domains. One class of these bottom-up maturity models is the Focus Area type (van Steenbergen et al., 2010, 2013), where capabilities are defined for different focus areas and arranged in a progressing order that can be mapped to maturity levels (cf. Figure 2.5). This type of maturity model effectively defines a domain and provides development teams with implementable practices and processes. Maturity models from several application areas such as Software Product Management or Enterprise Architecture employ focus area design (see (Sanchez-Puchol and Pastor-Collado, 2017) for a comprehensive survey).

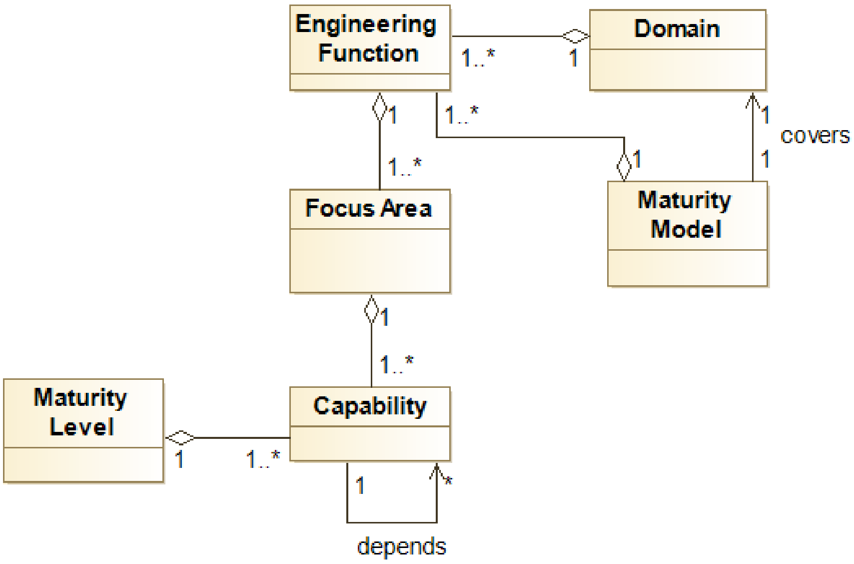


Figure 2.5: Focus area maturity model meta-model.

2.4 MODEL-BASED SYSTEMS ENGINEERING MATURITY MODEL

The MBSE MM was created during the project SPES2020[1]. It is a focus area maturity model which uses the SPES methodology (Broy et al., 2012; Pohl et al., 2016) as a reference framework. In the MBSE MM, functional domains are named *engineering function* to emphasize the relation to engineering phases in the development process (cf. Figure 2.6). The MBSE MM consists of 6 engineering functions that group a total of 15 focus areas addressing the different activities that models can support.

Each focus area has a set of capabilities, indexed by A – G, which describes a piece of an artifact or a capacity to perform methods (e.g., automatic analysis, simulate) onto these artifacts at different stages of modeling support. The capabilities are positioned against each other in a maturity matrix (cf. Figure 2.7). For example, the less mature capability (*i. e.,* A) in the focus area Goal Modeling is "Stakeholder goals regarding the function and quality of the system under consideration have been identified and documented". The following capabilities of the focus area characterize increasing use and analysis of models up to F: "The goal models are analyzed automatically in terms of consistency, satisfiability, and completeness".

---

[1] http://spes2020.informatik.tu-muenchen.de/spes_xt-home.html

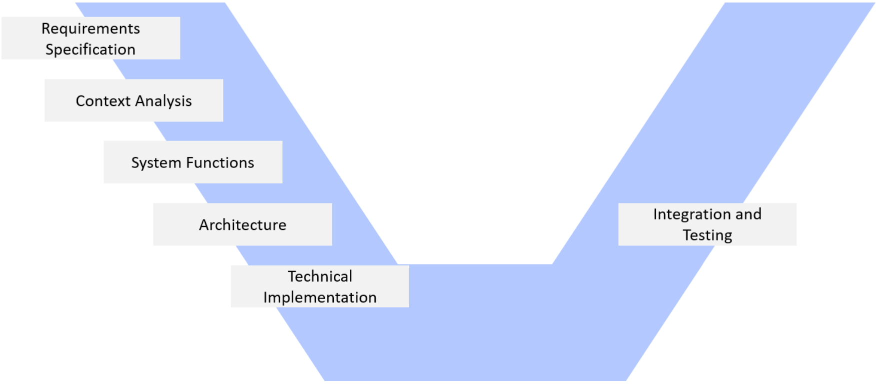


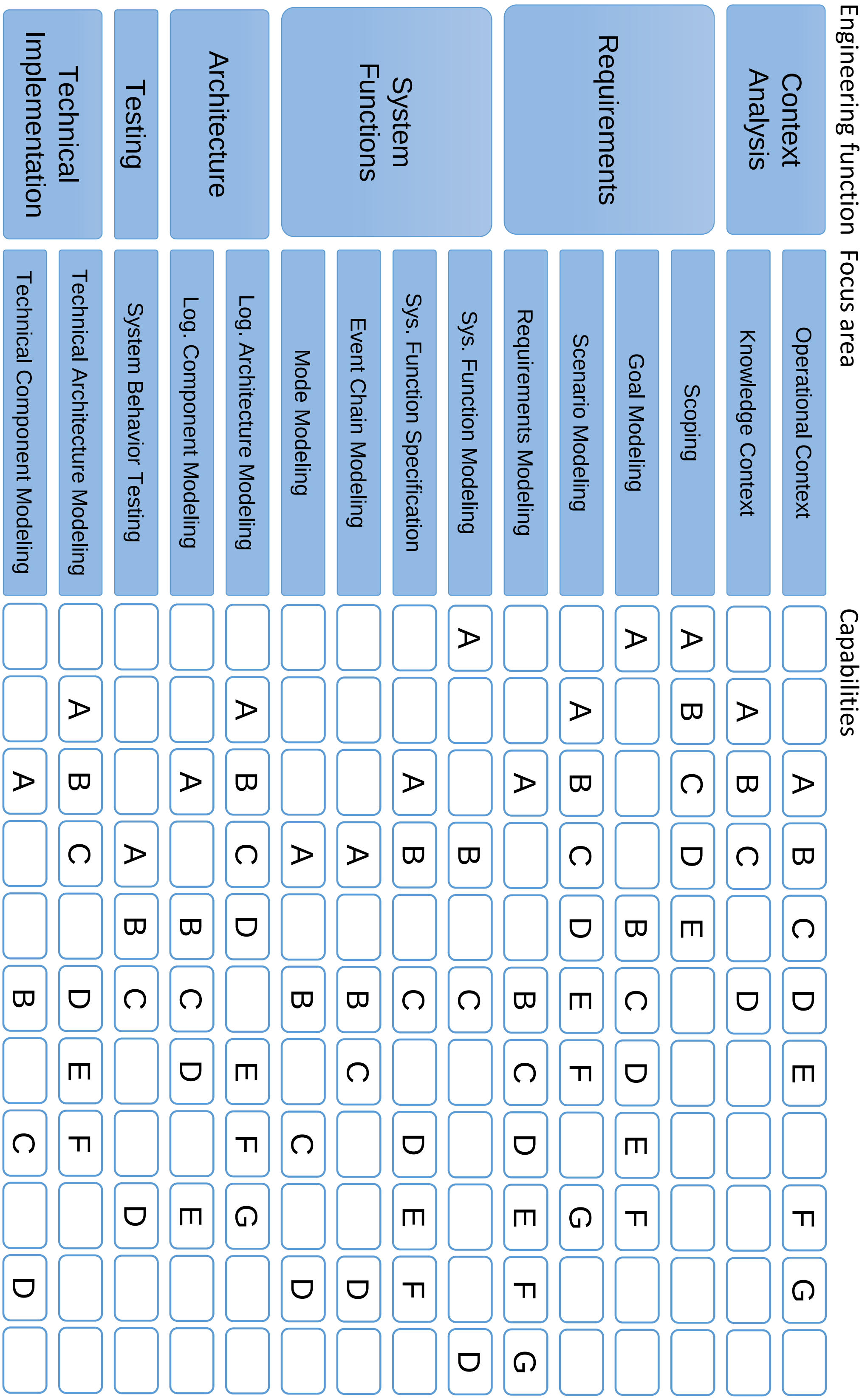
Figure 2.6: MBSE MM engineering functions represented in the V-model.

Capabilities located in higher levels of maturity may depend on capabilities on lower levels of maturity. Capabilities at the same level have no dependencies among each other. The maturity level describes how well MBSE practices are implemented. Improvement actions are associated with the capabilities to guide the development team in the incremental development of the engineering function. An overview of the MBSE MM can be seen in Figure 2.7.

Regarding its possible use, maturity models can be descriptive, prescriptive, or comparative (de Bruin et al., 2005). A maturity model is said to be descriptive when used for assessing the development team's current state. If it suggests improvements, the model is prescriptive, and when it can be used to compare the processes of different organizations, it is also comparative. The MBSE MM is entirely descriptive and partially prescriptive. Partially prescriptive because the model lacks guidance on selecting the subsequent capabilities to implement. It is not comparative because the MBSE MM was created to fit many development contexts, and there is no need to achieve maturity levels. Thus, some organizations might lack maturity in some focus areas because of business goal misalignment.

2.4.1 *Engineering functions and focus areas*

The MBSE MM comprises six engineering functions with 15 focus areas. The focus areas have different capabilities amount, as modeling is not equally distributed over different engineering functions. For instance, Requirements modeling has many practices while System Testing and Integration has only a tiny amount. The engineering functions and focus areas are depicted in Figure 2.7 and are described in the following paragraphs:



Engineering function	Focus area	Capabilities										
Context Analysis	Operational Context			A	B	C	D	E		F	G	
	Knowledge Context		A	B	C		D					
Requirements	Scoping	A	B	C	D	E						
	Goal Modeling	A				B	C	D	E	F		
	Scenario Modeling		A	B	C	D	E	F		G		
	Requirements Modeling			A			B	C	D	E	F	G
System Functions	Sys. Function Modeling	A			B		C					D
	Sys. Function Specification			A	B		C		D	E	F	
	Event Chain Modeling				A		B	C			D	
	Mode Modeling				A		B		C		D	
Architecture	Log. Architecture Modeling		A	B	C	D		E	F	G		
	Log. Component Modeling			A		B	C	D		E		
Testing	System Behavior Testing				A	B	C			D		
Technical Implementation	Technical Architecture Modeling		A	B	C		D	E	F			
	Technical Component Modeling			A			B		C		D	

Figure 2.7: Overview of the Model-based Systems Engineering Maturity Model.

CONTEXT ANALYSIS This engineering function groups capabilities related to modeling and context analysis of the system-under-development. The model differentiates two types of context:

- *Operational Context* encompasses actors and external systems that interact with the system at runtime.
- *Knowledge Context* describes relevant sources of information for the development of the system (e.g., stakeholders, standards, laws).

REQUIREMENTS This engineering function encompasses capabilities related to modeling and requirement analysis of the system-under-development from different perspectives.

- *Scoping* contains capabilities related to understanding the responsibilities and features of the system.
- *Goal Modeling* refers to stakeholder goals and their interdependencies.
- *Scenario Modeling* addresses desired interaction patterns between the system and its context entities.
- *Requirements Specification* contains capabilities related to specific desired properties of the system at its interface.

SYSTEM FUNCTIONS Capabilities of this engineering function are related to modeling and analyzing the system-under-development's user-observable functions (or features).

- *System Function Modeling* contains capabilities to provide an overview of all functions and their interplay.
- *System Function Specification* refers to the maturity of a single function specification.
- *Event Chain Modeling* addresses the necessary flow of information within the system to implement a function.
- *Mode Modeling* contains capabilities related to describing operational states of the system and their transitions.

ARCHITECTURE Capabilities related to modeling and analyzing the internal architecture of the system-under-development in terms of interacting (logical) components.

- *Logical Architecture Modeling* contains capabilities providing an overview over all components and their interplay.
- *Logical Component Modeling* refers to the specification maturity of components.

TECHNICAL IMPLEMENTATION Capabilities related to modeling and analyzing the technical architecture of the system-under-development in terms of physical devices and their communication.

- *Technical Architecture Modeling* contains capabilities to provide an overview of all technical components and their communication (e.g., electronic computing units (ECUs), bus systems, sensors, actuators).
- *Technical Component Modeling* refers to the specification maturity of technical components.

INTEGRATION AND TESTING Capabilities related to modeling and analyzing testing and integration activities.

- *System Behavior Testing* contains capabilities to test desired behavior of the system-under-development.
- *System Quality Testing* refers to testing desired qualities (a.k.a. non-function properties) of the system-under-development.

The capabilities of the MBSE MM are listed in Table A.2 and the positioning of the capabilities according to maturity levels is depicted in Table A.1.

2.4.2 *Capabilities*

The engineering functions have one or more focus areas, which are sliced into several capability levels. The capability levels are described using letters (e.g,. ‘A’, ‘B’, ‘C’), and the alphabetic order defines the maturity of the capability, being the ones that appear earlier of lower maturity (i.e., refinement, formalism, automation) compared to ones that appear later. The granularity of the capabilities was defined to consider small development steps towards the most mature focus area capability (the longest being eight maturity levels). The description of the capabilities defines either (1) how the information is stored (documents or models) or (2) how the information can be manipulated (static or dynamic analysis). The following capability types are described in the model.

- *identified and documented:* this capability type is usually performed using natural language and non-MBSE oriented tools. Such tasks encompass eliciting information from external sources as well as refinement and further development of existing artifacts. Although these capabilities do not employ models or specialized tools, the information acquired is needed for more mature ones. For instance: *"The vision of the system under consideration is identified and documented"* - Scoping SCO A.

- *modeled:* information is modeled using computer-aided design tools and formal languages. For instance: *"Objectives, sub-goals and their interrelationships are modeled"* - Goal Modeling GOM E.

- *is executable:* this activity allows the design tool to execute the models. For instance: *"The behavior of the logical components is described so that it is executable"* - Logical Component Modeling LCM E.

- *analyzed automatically:* properties of the system (e.g., inconsistency, incompleteness) are analyzed automatically using tool support. For instance: *"The behavior of individual groups of logical components is analyzed automatically"* - Logical Architecture Modeling LAM F.

- *generated automatically:* based on models; artifacts are automatically generated using tool support (e.g., code, documents, other models). For instance: *"The mapping of SW elements (resource-consuming elements) to the given execution platform (resource-providing elements) is generated automatically (deployment model)"* - Deployment DEP C.

- *can be simulated:* capabilities of this type enable model simulation, allowing engineers to identify design errors. For instance: *"The model of the system functions can be simulated together with a description of the context"* - System Function Modeling SFM D.

All capabilities, when adopted, add new procedures to the set of existing ones, except for capabilities of the *modeled* type, which some will replace existing tasks. A description of all MBSE MM capabilities can be seen in Table A.2.

3

FORCES THAT DRIVE OR PREVENT MODEL-BASED SYSTEMS ENGINEERING ADOPTION IN EMBEDDED SYSTEMS INDUSTRY

MBSE comprises a set of models and techniques that is often suggested as a solution to cope with the challenges of engineering complex systems. Although many practitioners agree on the techniques' potential benefits, companies struggle with the adoption of MBSE. In this chapter, we investigate the forces that prevent or impede the adoption of MBSE in companies that develop embedded software systems. We contrast the hindering forces with issues and challenges that drive these companies towards introducing MBSE. Our results are based on 20 interviews with experts from 10 organizations. Through exploratory research, we analyze the results employing thematic coding. Forces that prevent MBSE adoption relate mainly to immature tooling, uncertainty about the return-on-investment, and fears on migrating existing data and processes. On the other hand, MBSE adoption also has strong drivers, and participants have high expectations mainly concerning managing complexity, adhering to new regulations, and reducing costs. We conclude that bad experiences and frustration about MBSE adoption originate from false or too high expectations. Nevertheless, companies should not underestimate the necessary efforts for convincing employees and addressing their anxiety.

This chapter is based on previous publication (Vogelsang et al., 2017).

3.1 INTRODUCTION

Model-based Systems Engineering (MBSE) describes models and model-based techniques to develop complex systems, which are mainly driven by software (Broy et al., 2012). MBSE tackles those systems' complexity through an interrelated set of models, which connects all development activities and provides comprehensive analyses. Many companies face problems with the increasing complexity of software-intensive systems, their interdisciplinary development, and the massive amount of mainly text-based specifications. Such hurdles are especially true for embedded software systems (Broy, 2006). Model-based techniques offer a solution to managing these problems, and companies are attracted to its benefits.

Despite the envisioned MBSE benefits, companies are struggling with implementing their methods within development teams. Of course, organizational change is never easy (Conner, 1993; Hammer, 2007); however other methodologies, such as agile practices, have been adopted much faster. So, what are the reasons and factors that prevent or impede companies from adopting MBSE?

In this chapter, we investigate the forces that prevent or impede the adoption of MBSE in companies that develop embedded systems. We contrast forces that hinder its adoption with forces that drive companies towards introducing MBSE.

Our results are based on 20 interviews with experts from 10 organizations in Germany. We analyze the results through thematic coding and categorize the identified forces into inertia and anxiety forces, which prevent MBSE adoption, and push and pull forces, which drive the companies towards MBSE adoption. We frame the results with the coding of what the interviewees considered as MBSE. Our scientific contributions are the following:

- We present a set of hindering and fostering forces on MBSE adoption in the industry. These results were extracted from interviews with 20 experts from 10 organizations located in Germany.

- We analyze these forces to differentiate between MBSE specific forces and forces inherent to any kind of methodological change.

Forces that prevent MBSE adoption relate mainly to immature tooling, uncertainty about the return-on-investment, and fears on migrating existing data and processes. On the other hand, MBSE adoption also has strong drivers, and participants have high expectations mainly concerning managing complexity, adhering to new regulations, and detecting bugs earlier. We observed that the hindering forces are much more concrete and MBSE-specific than the fostering forces, which are often very generic (*e. g.,* increase in product quality, managing complexity, supporting reuse). Frequently, the interviewees could not even tell why or which part of MBSE contributes to the expected benefits.

From this, we conclude that bad experiences and frustration about MBSE adoption originate from false or too high expectations. Nevertheless, companies should not underestimate the necessary efforts for convincing employees and addressing their anxiety.

3.2 FORCES ON MBSE ADOPTION

In studies of customer demand and motivation, there is a model called Forces of Progress (Klement, 2018), which is used to define the emotional forces that generate and shape customers' high-level demand for a product. In this model, there are two groups of opposing forces. One group, which works to generate demand, is composed of two forces, namely *push* and *pull*. The other group is responsible for reducing and blocking demand and is also composed of two forces, namely *inertia* and *anxiety*. These forces are felt by customers differently while searching, choosing, and using a new product.

Likewise, when development teams face process change, their associates experience similar forces towards the endeavor, which is no different for MBSE adoption. In the following, we explain how these forces play when considering organizational change and their relation to the topic of this chapter.

3.2.1 *Push force*

Change does not take place while the current solution works fine. When this is not the case, the push force comes into play. The push force is generated by feeling that things currently are not sufficient anymore, either because the environment changed or the shortcomings of the current methods are not satisfactory. Then development teams feel impelled to change, to seek something else that might work better. Push can be internal or external:

- *Internal Push* this force emanates from frustration with the current way of doing things. Rather, they experienced a combination of circumstances that made them think, *"I don't like how things are; I want to make a change"*. In a systems engineering development team, bottlenecks in the current process are phenomena delivering push force.

- *External Push* the outside world is forcing development teams to change. Something in the environment changed, and the old way of solving their problems needs to change. New requirements from stakeholders are examples of how this force is triggered (*e.g.*, compliance with regulations).

3.2.2 *Pull force*

While push force starts the change movement, the pull force steers and directs it. There are two kinds of pull forces:

- *An idea of better life* Development teams do not change for the sake of changing; they do to help make their lives better. When they have the right methods for their problem, they can do things they could not before. The idea of this better life is what pulls them to take action. For instance, managers hear about the benefits of MBSE thus seeking to adopt it to keep competitive within its market.

- *Preference for a particular solution* Self-improvement motivates development teams to begin searching for and using a new solution; however, many factors play a role while choosing one. The context of the pull force shapes the desire to change, thus affecting the choosing criteria. Considering MBSE context, this force will steer an organization to select some specific method or tool.

3.2.3 *Anxiety force*

The anxiety force is associated with *how* or *if* a novel solution can deliver the expected benefits. It can happen in two situations, first, before deciding which solution should the selected, and later while familiarity with the product requires time. These two types are described in the following:

- *Anxiety-in-choice*, which boils up to feelings of uncertainty about whether a particular novel solution will deliver the expected benefits and help us achieve our goals. After the choice is made, the anxiety-in-choice largely vanishes, and the second type of anxiety might kick in. Managers feel this force due to unsureness of the efficacy of the new methods and if the methods are truly delivering the expected benefits.

- *Anxiety-in-use*. This form of anxiety takes place, as the name says, during the use of the solution and is generated by some properties of the new solution that we are not completely comfortable with it. Engineers used to the old way of doing things are prone to feel this force with the new methods when they get frustrated in some activity.

3.2.4 *Inertia force*

Inertia, as the name says, is the tendency to do nothing or to remain unchanged. Inertia is experienced in two different ways:

- *Habits-in-choice* are the forces that prevent development teams from switching the current way of doing things to a novel way. This force is experienced at the decision time for the new solution. The possible cumbersome needed translation of file formats between new and old tools is an example of a situation that triggers habits-in-choice.

- *Habits-in-use* is a force that takes place after the solution is implemented. Individuals start to use the solutions as planned, but sometimes they start to go slowly back to the old ways of doing things. These phenomena happen because keeping their old habits is easier. For instance, engineers might still use documents in parallel to the models to store information. In order to overcome this force, engineers should be helped to drop old habits and develop new ones (*e. g.,* training).

We created a quadrant diagram that groups the forces according to triggers and forces type (cf. Figure 3.2). On the vertical axis, we categorize the forces in two types, either hindering or fostering MBSE adoption. On the horizontal axis, we describe the triggers of these forces, either due to events in the present (*i. e., Current Situation*) or due to events in the future (*i. e., Envisioned Solution*). Additionally, we placed the forces in a timeline that displays the event of organizational change to depict when each force kicks in (cf. 3.1). Two events are depicted in this timeline, namely *Decision to change* and *New solution in place*.

3.3 STUDY APPROACH

In this section, we describe how the study was designed and conducted.

3.3.1 *Research questions*

We structure our research by two research questions that focus on hindering and fostering forces of MBSE adoption.

- RQ0: What means model-based engineering for practitioners?

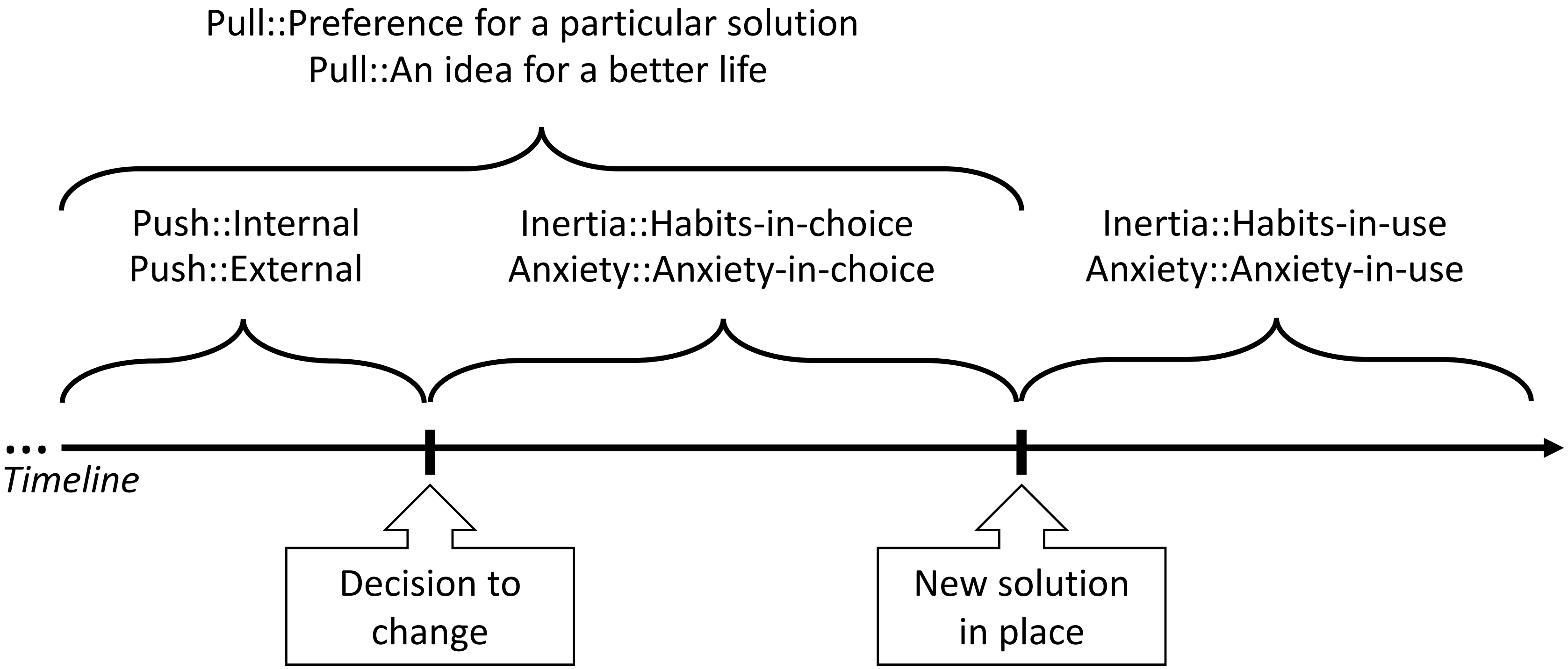
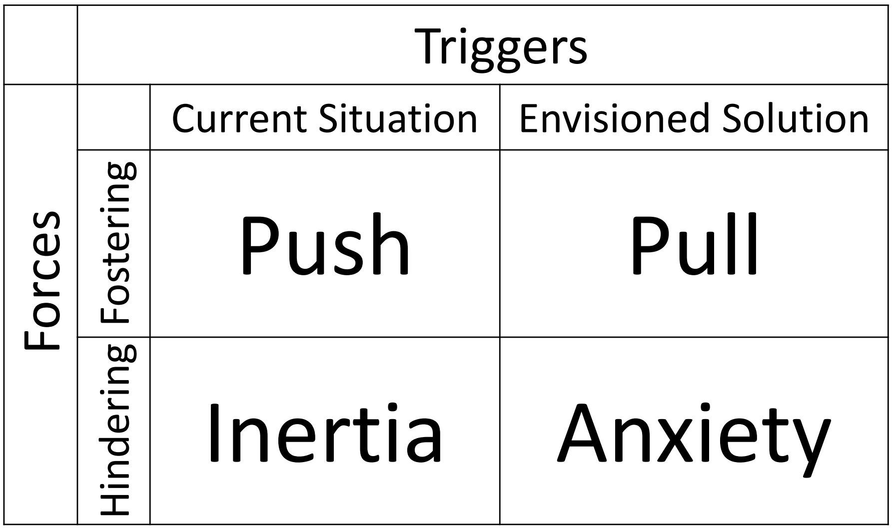


Figure 3.1: Timeline of change depicting when each force is triggered.



		Triggers	
		Current Situation	Envisioned Solution
Forces	Fostering	Push	Pull
	Hindering	Inertia	Anxiety

Figure 3.2: MBSE adoption forces diagram.

- RQ1: What are perceived forces that prevent MBSE adoption in the industry?
    - RQ1.1: What are habits and inertia that prevent MBSE adoption?
    - RQ1.2: What are anxiety factors that prevent MBSE adoption?
- RQ2: What are the perceived forces that foster MBSE adoption in the industry?
    - RQ2.1: What are the perceived issues that push the industry towards MBSE?
    - RQ2.2: What MBSE benefits are perceived as most attractive?

3.3.2 *Research design*

This work is an *exploratory research* (Shields and Rangarjan, 2013) based on semi-structured interviews. The method provides insights into the examined topic and gives essential information to understand the phenomenon in its real context (Dresch, Lacerda, and Antunes, 2015; Runeson and Höst, 2008). For performing the interviews, we developed an interview guide (Bryman, 2015). The interview guide was structured along a funnel model (Runeson and Höst, 2008) starting with general questions about the participant's context, understanding of MBSE concepts, and afterward going into detail about specific topics such as employee training, MBSE integration, or experiences in the past.

3.3.3 *Study participants*

The interview participants were selected from personal contacts of the authors and industrial partners that participate in a German research project[1] that has a focus on MBSE adoption in practice. The interviewee selection was based on two criteria: First, the interviewee should have several years of work experience. Second, the interviewee should work in an environment where MBSE adoption is a realistic option. In our case, we, therefore, restricted the group of interviewees to people working on embedded systems or in the context of embedded systems. Interviewees did not need to have adopted MBSE in their context; however, 13 of the 20 interviewees stated that they already have experiences in adopting MBSE. Table 3.1 provides an overview of the participants and their context. The interviews were conducted by two of the authors from May to December 2016.

3.3.4 *Interviews*

There were 20 face-to-face interviews. Every interview took around one hour. In consent with the interviewee, the interviewer took notes for detailed analysis. All interview notes were managed using the qualitative data analysis tool ATLAS.ti[2]. The interview guideline can be seen in Appendix B.

3.3.5 *Analysis*

Three researchers analyzed the interviews using *qualitative coding* (Neuman, 2010). Neither of them participated in the interview phase. The study was framed using the framework of *Forces on MBSE Adoption* (see Section 3.2) with the following codes: {Push, Pull, Inertia, Anxiety}. The analysis started with all three researchers working on the same five interviews. The results were later discussed and merged in a meeting. The discussions helped to homogenize the codes among the researchers (Weston et al., 2001) (*i. e.,* what/how to look for on each force). The remaining 15 interviews were tackled in a cross-

---

[1] `https://spedit.in.tum.de/`

[2] `http://atlasti.com`

Table 3.1: Study participants

ID	**Industry Sector**	**Type of Company**	**Role of Participant**	**MBSE Attitude**
P1	Tool vendor	OEM	Technical Sales	neutral
P2	Tool vendor	Academic	Professor	neutral
P3	R&D services	SME	Manager	neutral
P4	Automotive	OEM	Head of Development	positive
P5	Automotive	OEM	Systems Engineer	neutral
P6	Medical	SME	Head of SW Development	positive
P7	Medical	SME	Head of QA	positive
P8	Automotive	Supplier	Function Architect	negative
P9	Automotive	OEM	SW Architect	neutral
P10	Automotive	OEM	Function Architect	positive
P11	Research	Academic	Professor	negative
P12	Avionics	Supplier	Technical Project Manager	neutral
P13	Automotive	Supplier	Developer	positive
P14	Avionics	OEM	SW Developer	neutral
P15	Avionics	Supplier	SW Developer	negative
P16	Avionics	OEM	Team Lead	neutral
P17	Electronics	OEM	Head of SW Development	neutral
P18	Avionics	SME	Head of System Engineering	negative
P19	Robotics	OEM	Team Lead	positive
P20	Automotive	OEM	Research and Development	negative

analysis fashion (cf. Figure 3.3). The interviews were divided equally into three groups (A, B, C), and each researcher coded the interview transcripts of two groups (*i. e.,* AB, BC, or AC) individually the same way as before. Then, each researcher merged the results and judged existing conflicts of the group he did not work on (a researcher coding interviews of groups AB merged the results of interviews in group C). In a round with all three researchers, the unresolved conflicts were ironed out. Finally, the codes were divided into three groups {Pull, Inertia, (Anxiety, Push)} and each researcher worked on the quotations of codes of a group individually, performing open coding to create second-level codes (cf. Figure 3.4). We present the results in Section 3.4 by reporting the codes with the number of related quotations and the number of interviews in which the code appeared. The quotations' number indicates the code's significance overall interviews, and the number of interviews indicates the pervasiveness of the code within the interviews. The codebook can be seen in Appendix D.

3.3.6 *Threats to validity*

The validity of our results are subject to the following threats:

Subject selection bias. Since this is an exploratory study, we selected a convenience sample of project partners and friends as study subjects. Although we selected study participants from a broad spectrum of companies and industrial domains, the results may be influenced by the fact that all study participants work in Germany. Additionally, the interviewees were selected from an environment where MBSE adoption is a realistic option.

Researcher bias. Our study was carried out in the context of a project on transferring MBSE into practice, which means that the authors have a positive attitude towards MBSE in general. Additionally, some interviewees are also partners in this project; however, we interviewed people from companies not involved in the project. In order to reduce researcher bias, the interviews were conducted by two researchers who took notes independently.

Reactivity refers to how much of the observations are caused because the interviewers are present. Investigators can impact both the environment and the people being observed. Getting rid of reactivity threats is impossible; however, the investigator should be aware of it and how it influences what is observed. Thus, validity is threatened by the possibility of misunderstandings between interviewees and the researchers. In order to minimize this risk, the study goal was explained to the participants before the interview. Steps taken to improve the reliability of the interview guide included a review and a pilot test.

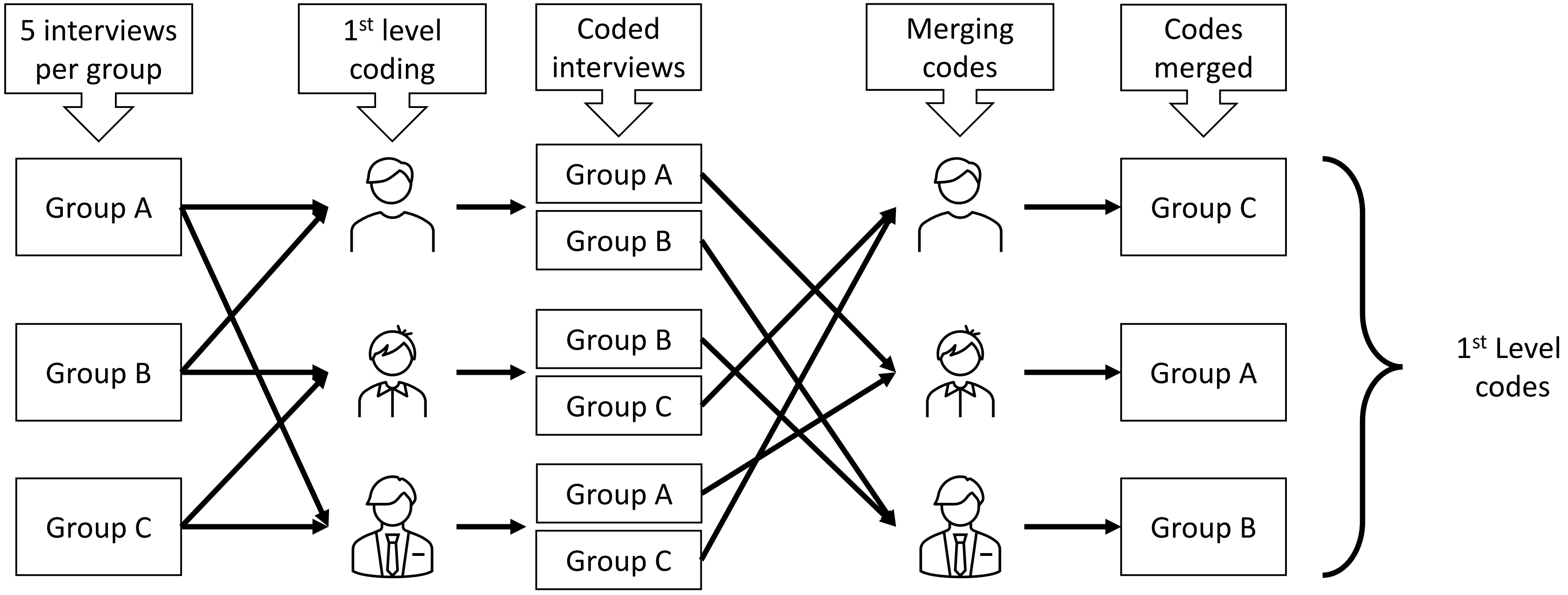


Figure 3.3: Analysis process to retrieve the first level codes from the interviews.

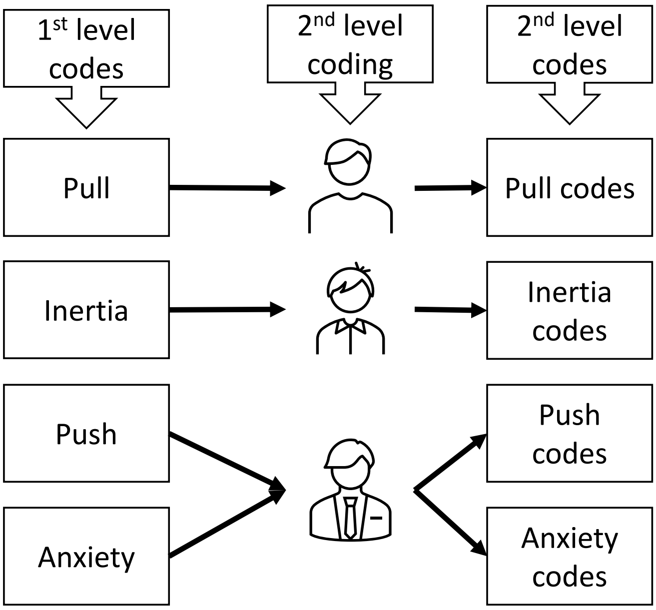


Figure 3.4: Analysis process to retrieve the second level codes from the interviews.

We followed several strategies proposed by Maxwell (Maxwell, 2012) to mitigate the threats. The interviews were conducted as part of a larger project, where we established a *long-term involvement* of the study subjects. As part of this, we presented our study in the project context, where the project partners reviewed the results. We substantiate our assertions by providing *quasi-statistics* on the frequency of code occurrences in the interview data. To further validate our results, we *compared* them with existing studies on development methodology adoption.

External validity. We expect that our results represent the German embedded systems industry; however, we cannot generalize the results to other countries or other types of systems engineering.

3.4 RESULTS

In this section, we present the results of our study. In the following subsection, we present an overview of the results and the definition of MBSE according to the study's participants. Further, we describe how each force influences the adoption of MBSE supported by the verbatim of the interviews.

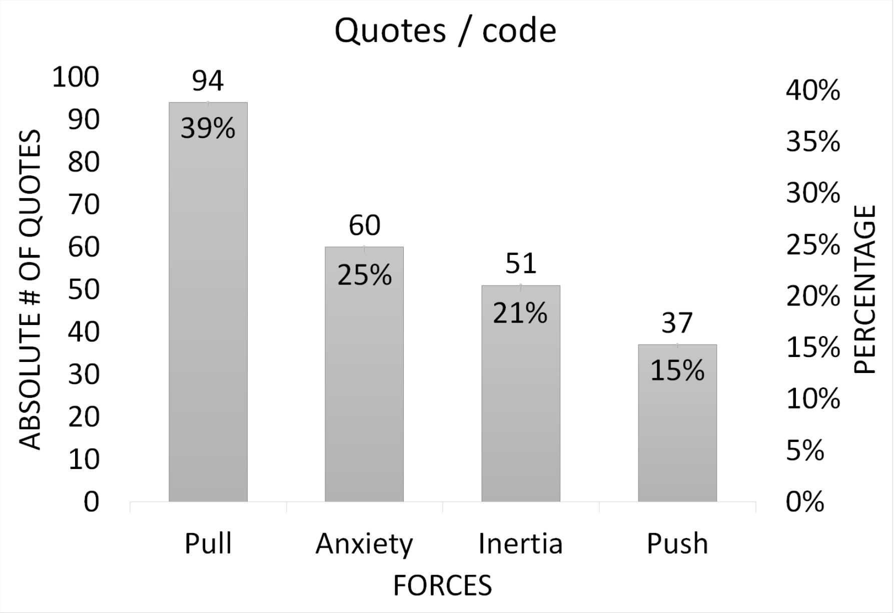


Figure 3.5: Number of quotations related to MBSE adoption forces.

3.4.1 *Overview*

In total, we coded 242 quotations. Their distribution between the forces can be seen in Figure 3.5. The fostering (131 times) and hindering (111) forces were mentioned to a similar amount. Quotations categorized as pull (94) are almost triple of push (37). These numbers can be compared to the quotations on inertia (51) and anxiety (58). Overall, pull forces were coded most, representing 39% of all quotations.

To analyze the general attitude of a participant towards MBSE adoption, we divided the number of coded quotations related to fostering forces (push and pull) by the total number of quotations coded for that participant. We considered a participant to have a positive attitude when fostering forces was higher than 60%, a neutral attitude for ratios between 60% and 40%, and a negative attitude for a ratio smaller than 40%. As depicted in Table 3.1, we had a relatively balanced set of participants concerning MBSE attitude. For 9 out of the 20 interviews, we coded a similar number of fostering and hindering forces (*i. e.,* neutral attitude). In 6 interviews, the fostering forces dominated (*i. e.,* positive attitude), and in 5 interviews, the hindering forces dominated (*i. e.,* negative attitude).

The results of the last step of the coding process generated similar codes in different categories (*e. g.,* Tooling Shortcomings from Anxiety category and Immature tooling or Incompatibility with existing tools, both from Inertia category). Although similar names, these codes encompass disjoint characteristics, and their coexistence serves a purpose. All codes created during the analysis can be seen in Figure 3.6.

3.4.2 *Definition of MBSE*

In this section, we try to answer *RQ0: What means model-based engineering for practitioners?*. In the interviews, we did not refer to any specific MBSE approach. We did this on purpose to identify forces independent from any concrete technique or tooling. Additionally, comparing the results would have been much harder due to the large variety of MBSE approaches and flavors. Nevertheless, we asked the interviewees to define MBSE. The result can be seen in Figure 3.7, where a word cloud representation of terms mentioned more than two times is depicted.

The word cloud shows the close association of MBSE with graphical models. Especially graphic descriptions of architectures and processes were mentioned several times. However, some interviewees mentioned that *"graphical representation is only a part of MBSE, not everything" (P12)* and others pointed out that MBSE should not be deformed to *graphical programming*. The only reference to a specific instance of MBSE in the word cloud is given by *Simulink*. Simulink[3] is a widely used tool in the embedded systems domain for modeling, simulating, and analyzing dynamic systems. Interestingly, the interviewees mentioned that using Simulink is *not* considered as doing MBSE (*e. g.*, P4:*"Pure implementation with Simulink is graphical programming, not MBSE."*, P16:*"Simulink is model-based engineering but not model-based* systems *engineering"*). UML/SysML, which we expected to appear more often in the characterization of MBSE, was only mentioned rarely; however, *notation* was mentioned several times. The term *information model* was used a few times as an essential part of an MBSE approach. P7: *"A core topic of MBSE is the information model that specifies and relates all development artifacts."* Apart from that, the interviewees frequently mentioned several well-known properties related to MBSE, such as *abstraction, formalization, and comprehension*. In summary, the results show that our interviewees were not biased by a specific MBSE flavor or approach that they previously had in mind when answering our questions. However, the variety of answers also shows that MBSE is still far away from common understanding.

---

[3] `https://de.mathworks.com/products/simulink.html`

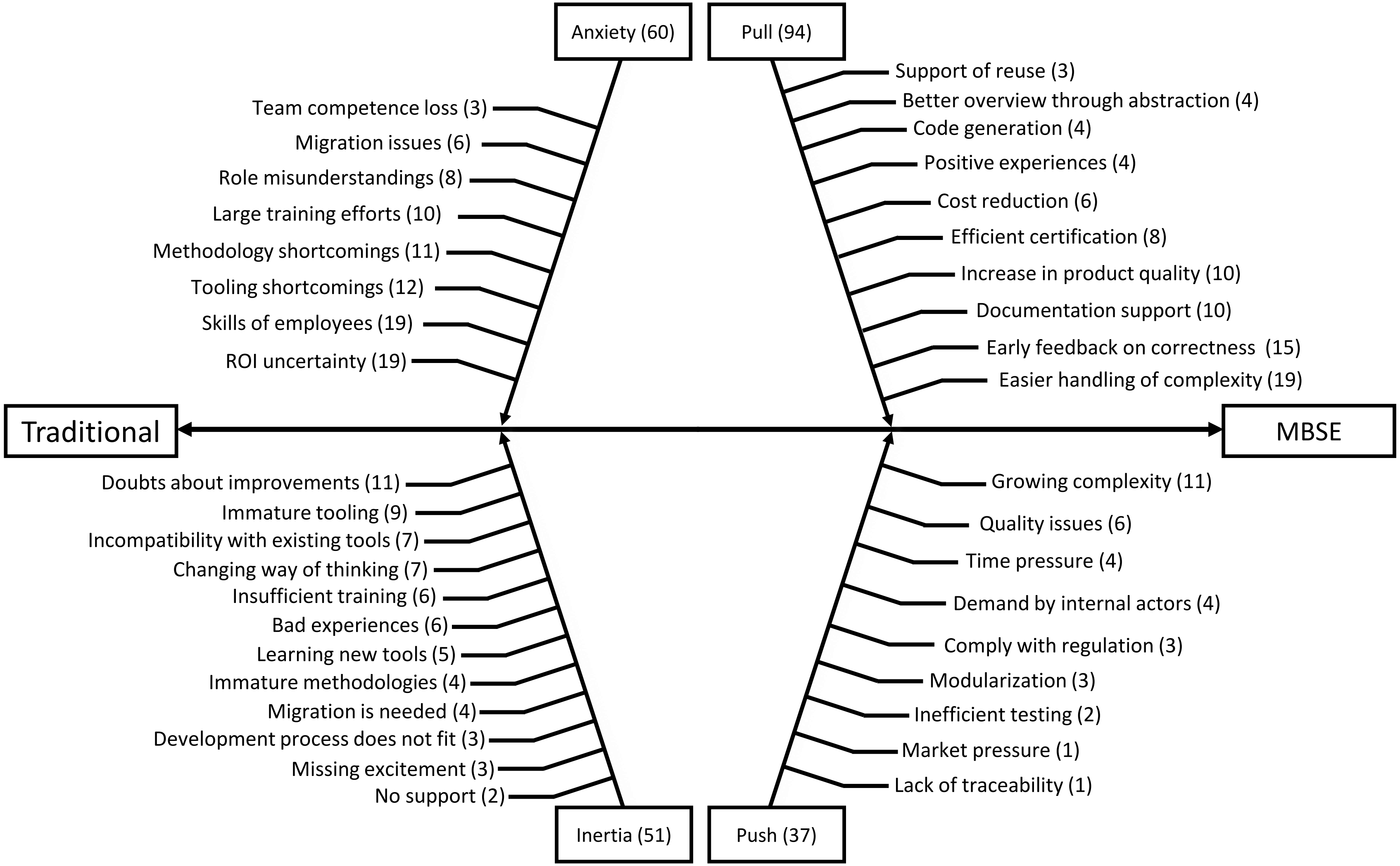


Figure 3.6: Fishbone diagram overview of MBSE adoption forces.

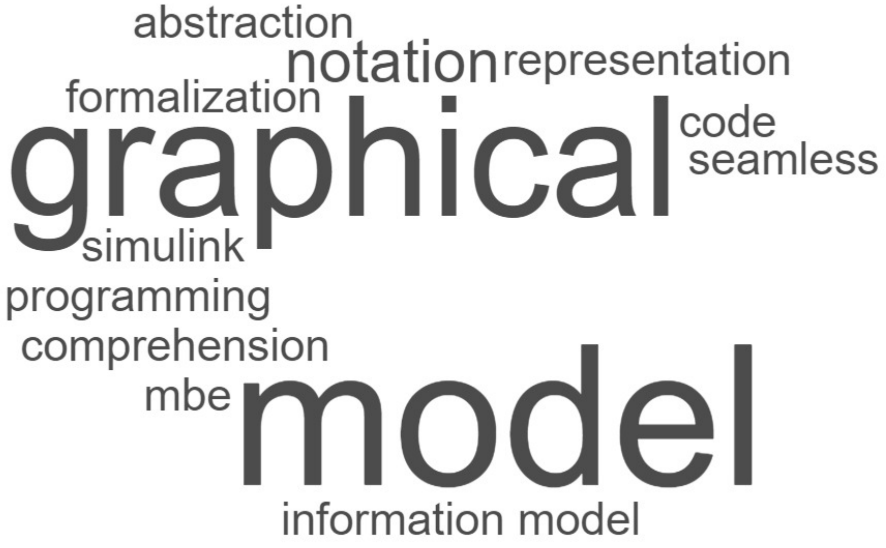


Figure 3.7: Word cloud of MBSE descriptions.

In the following, the forces found in the study are subsequently described and explained using the information from the interview transcripts and the interpretations from coding and analysis.

3.4.3 *Hindering Force: Inertia*

With 51 distinct quotations, *inertia* forces were mentioned fewer times than forces related to anxiety (60 quotations). We structured the inertia-related quotations concerning four inertia topics.

TOOLING INERTIA (21 CODED QUOTATIONS FROM 15 INTERVIEWS) With 21 quotations, tooling inertia was the most frequently mentioned inertia force. Tooling inertia describes phenomena of the current in-house tooling environment that made our participants refrain from adopting MBSE. Tooling inertia includes resistance against *learning new tools* and potential *incompatibilities of MBSE tools* with current tools.

"People preferred using Excel instead of the new MBSE tool" (P8)

"Especially elderly employees who are used to textual specifications have difficulties with drawing tools" (P15)

"It is not possible to connect/trace the models with artifacts in other tools." (P5)

Apart from learning and integrating new tools, our participants also reported on employees' resistance if *MBSE tools are immature*. Especially tools with poor user experience, low stability, and missing essential features are significant factors in why employees resist MBSE adoption.

"Tool is not user friendly. Things are distributed over several menus; you have to look for everything." (P5)

"We are working in teams. That is why we need a tool with fine-grained access rights and control." (P10)

We classified *immature tooling* as inertia force because the expectation of missing essential features makes the current situation look not so bad. We will see that tooling issues were also mentioned in the context of anxiety. In these cases, interviewees mentioned fears that the currently available tools cannot fulfill the promises of MBSE.

CONTEXT INERTIA (18 CODED QUOTATIONS FROM 13 INTERVIEWS) A second inertia force mentioned quite often was context inertia, which describes people refraining from MBSE adoption because they believe it does not fit their current business situation. The most mentioned in this category was doubts about whether MBSE would improve the current situation.

"It needs a huge emergency to justify the costs of introducing an MBSE tool." (P7)

"Currently, problems are not so urgent yet. Therefore, there is not much willingness to act." (P20)

Another aspect that makes people refrain from MBSE adoption is the potential need to migrate old data or legacy systems or when it seems that the current development process does not fit MBSE techniques.

"Legacy problems are a huge hurdle because, in general, the old way of working must further be maintained and supported." (P20)

"MBSE adoption would have caused changes in our development process. Therefore, we did not do it." (P2)

PERSONAL INERTIA (16 CODED QUOTATIONS FROM 9 INTERVIEWS) Personal inertia captures forces related to an individual's personality and experiences that hold him/her back from adopting MBSE. In our study, these forces were lead by the resistance against learning a new way of thinking.

"MBSE is not just about changing the notation; it is about changing the way of how I think about systems" (P2)

"Abstractions in MBSE are not easy to comprehend." (P12)

Similarly, if people had negative experiences with MBSE or related techniques, they have a personal reluctance against adopting MBSE in their current situation.

MATURITY INERTIA (12 CODED QUOTATIONS FROM 8 INTERVIEWS) Maturity inertia was the least mentioned in our interviews. Participants were critical about a potential MBSE adoption if they had the impression that: The MBSE methodology is not mature enough. There has not been sufficient training before. There is no support from experts.

"We first need a common terminology between employees of different departments" (P7)

"The support for debugging problems is very limited" (P9)

3.4.4 *Hindering Force: Anxiety*

Anxiety is a force related to expectations and fears that make MBSE adoption less appealing. These expectations originate from uncertainties that are still to be clarified or a false perception of reality. We structured the anxiety-related quotations into the following topics:

ROI UNCERTAINTY (19 CODED QUOTATIONS FROM 12 INTERVIEWS) Return on investment (ROI) is the benefit resulting from an investment. Any development team introducing MBSE will incur cost spread in several factors such as training, tooling, migration, or lower productivity. Many interviewees were concerned that the investments in introducing MBSE will not pay off.

"[It will costs us] A large sum in the million range" (P7)

"Coaching on the job is very important, but it costs a lot" (P2)

SKILLS OF EMPLOYEES (19 CODED QUOTATIONS FROM 11 INTERVIEWS) Some interviewees fear that (some of) the employees in their company may lack the necessary skills to adopt MBSE efficiently. This shortcoming can negatively influence the introduction of MBSE in two different ways: Either those employees do not adopt MBSE, or they misapply them.

"Mechanical engineers know CAD modeling but do not know modeling of behavior" (P1)

"Modeling should not be an end in itself" (P16)

TOOLING SHORTCOMINGS (12 CODED QUOTATIONS FROM 8 INTERVIEWS) The interviewees perceived problems with tooling as a reason for not introducing MBSE. The interviewees fear that current tool solutions do not address a significant part of the development process and the envisioned benefits of MBSE. Thus, extra work would be necessary to fill the gaps (*e.g.,* migration of data between MBSE tools and current tools).

"Everything in one tool? Nobody wants that" (P5)

"Performance of the tools [is a challenge for introducing MBSE]" (P7)

METHODOLOGY SHORTCOMINGS (11 CODED QUOTATIONS FROM 6 INTERVIEWS) Many interviewees emphasized the lack of maturity on the current MBSE methodology. This category can be interpreted in two ways. Either the methodology is incomplete, or the knowledge of practitioners is immature. Besides, concerns about the lack of tailored approaches for MBSE introduction were pointed out.

"A consistent methodology is lacking, resulting in uncertainties" (P1)

"There are no process models that integrate MBSE properly." (P11)

LARGE TRAINING EFFORTS (10 CODED QUOTATIONS FROM 5 INTERVIEWS) This category grouped perceived potential problems related to training the team on using MBSE and its respective tools. Some of the codes were related to training costs and had intersections with *ROI uncertainty*. Other codes were related to the fear of unsuccessful training.

"Training is necessary: How do I bring my employees to the same level as the experts?" (P7)

"Employees will not accept MBSE if no training is provided before." (P7)

Besides these categories, interviewees also mentioned potential *team competence loss* (3 times coded from 3 interviews) and new responsibilities in the team that could cause *role misunderstandings* (8 times coded from 5 interviews). The interviewees perceived *migration issues* (6 times coded from 6 interviews) of projects that started with traditional development method to MBSE.

3.4.5 *Fostering Force: Push*

With 37 distinct quotations, *push* was the force with the smallest number of quotations. We structured push forces within three categories:

PRODUCT PUSH (20 CODED QUOTATIONS FROM 10 INTERVIEWS) We grouped here codes related to product-oriented push forces. *Growing complexity* (11 coded / 8 interviews) of the software was the code with most quotations within the push forces. As systems become more software-intensive, tackling the growing complexity is currently a real challenge; thus, development teams feel the need to shift to better solutions.

"Increasing complexity of products [pushes us towards MBSE]" (P1)

"Complex software, especially with concurrency [pushes us towards MBSE]" (P3)

Other codes were *quality issues* (6/3) within the product or its specification and the need for *modularization* (3/3) in order to make certification and reuse more efficient.

STAKEHOLDER ENFORCEMENT (8 CODED QUOTATIONS FROM 4 INTERVIEWS) Some interviewees mentioned that they are forced or pushed towards MBSE by recommendations or stakeholders' requests. *Demands by internal actors* (4/3) such as developers or management push companies towards MBSE adoption as well as legal requirements to *comply with regulations* (3/1). *Market pressure* (1) was mentioned concerning issues with acquiring talented employees:

"We have to be modern; otherwise we will not get good people anymore" (P2)

PROCESS PUSH (7 CODED QUOTATIONS FROM 4 INTERVIEWS) Deficiencies of the current process were only mentioned a few times as forces that push companies towards MBSE. The codes were *time pressure* (4/3), *inefficient testing* (2/2), and *lack of traceability* (1).

"We have no idea what happens when something changes" (P5)

"[We have] Large amounts of requirements; how can the tester handle this?" (P5)

In summary, interviewees provided more push forces related to issues with the product instead of issues with the process.

3.4.6 *Fostering Force: Pull*

We identified several factors of envisioned benefits that drive companies towards MBSE adoption. A majority of the interviewees' responses are related to envisioned improvements of the development process, which is interesting since process issues were only mentioned a few times as push factors.

EASIER HANDLING OF COMPLEXITY (19 CODED QUOTATIONS FROM 12 INTERVIEWS) With each new function to integrate, the complexity of software increases. Managing the different software components gets more and more complicated. The interviewees see great opportunities in MBSE to support this challenge. Due to many possible variants of products, the complexity of software increases in many companies.

"[MBSE will help us to] understand highly complex issues or illustrate something" (P15)

"[MBSE will support the] management of product line and variability" (P1)

EARLY FEEDBACK ON CORRECTNESS (15 CODED QUOTATIONS FROM 10 INTERVIEWS) The desire for early feedback and front-loading was also a stark pull factor. Early verification on higher levels of development was specially mentioned to improve the development process and, finally the product.

"Early verification and simulation saves time in the end" (P7)

"[MBSE will provide] better quality due to early fault detection" (P4)

"[MBSE will] Enable automatic verification" (P6)

DOCUMENTATION SUPPORT (10 CODED QUOTATIONS FROM 7 INTERVIEWS) The interviewees expect support to create and manage documentation. The increasing complexity of software development has complicated the management of System Requirements documents.

"[MBSE will provide] better documentation" (P13)

"[MBSE will] generate documentation and code" (P12)

INCREASE IN PRODUCT QUALITY (10 CODED QUOTATIONS FROM 5 INTERVIEWS) The interviewees expected to deliver better products by introducing MBSE, which includes the final product and intermediate development artifacts.

"[MBSE will] improve the quality of requirement documents" (P10)

EFFICIENT CERTIFICATION (8 CODED QUOTATIONS FROM 5 INTERVIEWS) Some interviewees envision that MBSE will make it easier to certify software-intensive products. Some interviewees specifically mentioned that MBSE would enable a modular certification, where only parts of the product are certified and not the entire product.

"[MBSE is] necessary to comply with regulatory requirements" (P6)

"[MBSE will enable] modular certification and parallel development" (P6)

Additional, less frequently mentioned, pull factors include *cost reduction* (6 coded quotations), *positive experiences* (4), *code generation* (4), *better overview through abstraction* (4), and *support of reuse* (3).

3.5 DISCUSSION

The results show that people from the industry have high hopes and expectations for MBSE. However, several hurdles need to be addressed when adopting MBSE, some of which are very generic. These problems are sometimes even part of human nature and its natural resistance to change in general.

3.5.1 *Relation to existing evidence*

When comparing our results to related studies on forces of adopting development methodologies in the industry, we can identify some general patterns. Hohl et al. (Hohl et al., 2016) report on forces preventing the adoption of agile development in the automotive domain. They also report on inertia and anxiety forces resisting a necessary change of mindset or limited organizational restructuring acceptance. Additionally, the current development process was perceived as good enough. The same forces also appeared in our study. Riungu-Kalliosaari et al. (Riungu-Kalliosaari et al., 2016) performed a case study on the adoption of DevOps in industry, where they identified five high-level adoption challenges. Three of these challenges were also mentioned as inertia or anxiety factors in our study, namely *deep-seated company culture*, *industry constraints and feasibility*, and *unclear methodology*. Parallels can also be found in the work of Bauer and Vetrò (Bauer and Vetrò, 2016) concerning the adoption of structured reuse approaches in the industry.

Similarly, we also found common and generic goals (*i. e.*, pull forces) that focus on many process improvement activities. Schmitt and Diebold (Schmitt and Diebold, 2016) have analyzed common improvement goals that are usually considered when improving the development process. The pull factors that we extracted in our study are part of the main goals elicited by Schmitt and Diebold (especially quality and time-to-market)

When focusing on the forces specific to MBSE that did not appear (so strongly) in the related studies, some factors remain. The incompatibility of MBSE tools with existing tools is a specific inertia force that prevents MBSE adoption. The second force of inertia specifically reported for MBSE adoption is adopting a new way of thinking, especially concerning abstractions. The anxiety forces that we identified were rather generic such that we did not identify any MBSE specific anxiety forces. Interestingly, loss of competencies or loss of power, a typical anxiety factor, was not often mentioned.

3.5.2 *Impact for industry*

MBSE streamlines the activities in all phases of the software lifecycle. It replaces document-based systems engineering and automates several tasks (*e. g.,* code generation). A development team transitioning from document-based to model-based will require changes in all software development stages, including tools, processes, artifacts, and developing paradigms.

Our interviewees focused more on push forces related to the product and not so much on the process. One might infer that engineers recognize their products' growing complexity, but they cannot link it to the current processes' shortcomings. Perhaps they believe the processes are okay, since it has been functioning correctly until now, and the problem is the product, which is getting more challenging to develop.

The results support decision-making and are an initial step towards efficiently introducing MBSE in companies. Implementing change is always a hassle; therefore, companies should manage expectations by setting concrete improvement goals, relating them to concrete MBSE techniques, and making changes step-by-step. Many interviewees mentioned that MBSE adoption should best be piloted in small projects with a clear scope. For ordinary organizational hindering forces, the state-of-the-art provides many methods. They are also relevant and are also affected by the MBSE specific ones.

We placed the forces in a timeline of change depicting when they are triggered (cf. 3.1). The goal is to help practitioners be attentive when the forces are expected to kick in and mitigate the hindering force towards adoption.

3.5.3 *Impact for academia*

MBSE complexity raises uncertainties towards the effort and success of its introduction. These uncertainties can be mitigated by knowledge building. Misunderstandings of MBSE, its tools, and processes were quoted often, which means research is not correctly reaching practitioners. This problem is not limited to the MBSE domain but to research in general. With a clear idea of the forces fostering and hindering MBSE introduction, the next step is to understand how to manage those factors, mitigate them when necessary, or strengthen those that contribute to successful MBSE introduction. The results provide promising research directions based on real industry needs.

4

STRATEGIES AND BEST PRACTICES FOR MODEL-BASED SYSTEMS ENGINEERING ADOPTION IN EMBEDDED SYSTEMS INDUSTRY

MBSE advocates the integrated use of models throughout all development phases of a system development life-cycle. It is also often suggested as a solution to cope with the challenges of engineering complex systems. However, MBSE adoption is no trivial task, and companies, especially large ones, struggle to achieve it in a timely and effective way. We aim to discover the best practices and strategies to implement MBSE in companies that develop embedded software systems. Using an inductive-deductive research approach, we conducted 14 semi-structured interviews with experts from10 companies. Further, we analyzed the data and drew some conclusions validated by an online questionnaire in a triangulation fashion. Our findings are summarized in an empirically validated list of 18 best practices for MBSE adoption and a prioritized list of the six most important best practices. Raising engineers' awareness regarding MBSE advantages and acquiring experience through small projects are considered the most critical practices to increase MBSEadoption's success.

This chapter is partly based on previous publication (Amorim et al., 2019).

4.1 INTRODUCTION

Model-based Systems Engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities. It begins in the conceptual design phase and continues throughout development and later life cycle phases (INCOSE, 2007). MBSE is part of a long-term trend towards model-centric approaches adopted by other engineering disciplines, including mechanical, electrical, and software.

In this approach, models (as opposed to document-centric approaches) serve as blueprints for developers to write code, provide formalization, tackle complexity, and enhance the system's understanding. Specialized tools automate much of the non-creative work (which translates to gains in productivity and quality) and generate code based on the models. MBSE fosters artifact reuse, improves product quality, and shortens time to market (Carroll and Malins, 2016).

Despite the aforementioned benefits, adopting MBSE is a complex task, especially for large and established companies (Vogelsang et al., 2017). Process changes are required in all system life-cycle phases and a shift in the development paradigm (i.e., abstract thinking (Hutchinson, Whittle, and Rouncefield, 2014)), and the application of new tools. Projects are not likely to meet their cost and delivery target when adoption is carried out poorly.

Our goal was to find out what was tried, what worked, what did not work, how the problems were solved, what can be recommended, and what should be avoided when adopting MBSE in organizations that develop embedded systems. For this purpose, we conducted 14 semi-structured interviews with experts from embedded systems organizations. From these interviews, we extracted 18 best practices fitted for tackling MBSE adoption challenges. Sequentially, we validated and prioritized the best practices with the help of an on-line questionnaire, which MBSE practitioners answered.

Our findings provide input for planning MBSE adoption based on practitioners' knowledge that went through the experience of implementing MBSE in already established embedded systems development organizations.

This chapter presents the following contributions:

- A granular set of MBSE adoption strategies and best practices derived from real field experience validated by practitioners through a questionnaire.
- A prioritized list of the six most important best practices.
- The findings shed light on MBSE adoption issues and how practitioners overcame them.

We expect that this knowledge can help organizations to adopt MBSE in a cost-effective and timely manner.

4.2 STUDY APPROACH

In this section we describe how the study was designed.

4.2.1 *Research objective*

The research objective is summarized in terms of the Goal-Question-Metric (Basili, 1992) (a.k.a. GQM) template in Table 4.1.

Table 4.1: Research objective.

Analyze	MBSE adoption
for the purpose of	increasing efficiency (i.e., cost and time)
with respect to	adoption strategies and best practices
from the viewpoint of	engineers that participated or observed some endeavor to introduce Modelbased Engineering in a organization
in the context of	embedded systems industry.

4.2.2 *Research design*

For this study, we devised an inductive-deductive research approach in a triangulation fashion (Denzin, 2006) composed of three main activities as depicted in Figure 4.1. Inductive reasoning is used to build hypotheses and theories. However, inductive reasoning allows for the conclusion to be false (Copi, 2003), thus we applied deductive reasoning to our findings. Such measure also addresses some threats to validity (cf. Section 4.2.3). The activities are described in detail in the following paragraphs.

4.2.2.1 *Inductive reasoning*

At the inductive phase, we followed an exploratory research approach (Shields and Rangarjan, 2013) to identify MBSE adoption strategies and best practices from the experience of experts. The method provides insights into the examined topic and gives essential information to understand the phenomenon in its real context (Dresch, Lacerda, and Antunes, 2015; Runeson and Höst, 2008). Semi-structured interviews were used to collect the data, qualitative coding analysis (Neuman, 2010) was used to analyze the data.

INTERVIEW WITH PRACTITIONERS. We conducted 14 face-to-face interviews, each taking around 60 minutes. We developed an interview guide (Bryman, 2015) that was structured along a funnel model (Runeson and Höst, 2008). It started with general questions about the participant's context and the understanding of MBSE. Afterward, detail about specific topics such as employee training, process and tooling selection, and adoption experiences. The interviewee selection was based on two criteria: First, the interviewee should have work

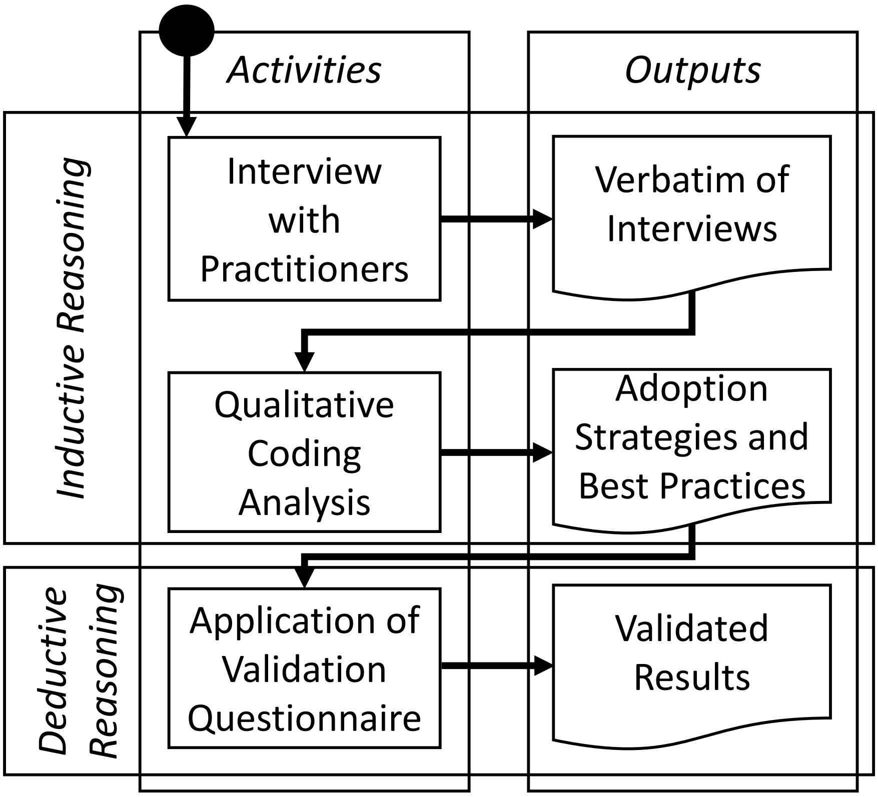


Figure 4.1: Research workflow.

Table 4.2: Interview participants.

ID	Industry Sector	Type of Company	Role of Participant
P1	Tool vendor	OEM	Technical Sales
P2	Tool vendor	Academic	Professor
P3	R&D services	SME	Manager
P4	Automotive	OEM	Head of Development
P5	Automotive	OEM	Systems Engineer
P6	Medical	SME	Head of SW Development
P7	Automotive	Supplier	Function Architect
P8	Automotive	OEM	SW Architect
P9	Research	Academic	Professor
P10	Automotive	Supplier	Developer
P11	Avionics	OEM	Team Lead
P12	Electronics	OEM	Head of SW Development
P13	Avionics	SME	Head of System Engineering
P14	Robotics	OEM	Team Lead

experience of several years. Second, the interviewee should work in an environment where MBSE adoption is a realistic option. Therefore, we restricted the group of interviewees to people working on embedded systems or within this context. The participants were screened from work contacts of the authors and industrial partners that participated in a research project[1] with a focus on MBSE adoption in practice. Table 4.2 provides an overview of the participants and respective demographic data. In consent with the interviewee, the interviewer took notes for detailed analysis. The outcome of this activity is the *Verbatim of Interviews* (cf. Figure 4.1), which is input for the next activity.

QUALITATIVE CODING ANALYSIS. Three researchers analyzed the interviews using qualitative coding (Neuman, 2010) and managed using the qualitative data analysis tool ATLAS.ti[2]. Neither of them participated in the interview phase. The analysis started with all three

---

[1] `https://spedit.in.tum.de/`

[2] `http://atlasti.com`

researchers working on the same five interviews. The results were later discussed and merged in a meeting. We used the discussions to homogenize the understanding of the codes among the researchers (Weston et al., 2001) (i.e., what/how to look for). The remaining interviews were tackled in a cross-analysis fashion. In a round with all three researchers, the unresolved conflicts were ironed out. The outcome of this activity was a list of *Adoption Strategies and Best Practices* for MBSE adoption (cf. Figure 4.1).

4.2.2.2 *Deductive reasoning*

Rather than obtaining new information, our goal in this phase was to validate our previous finding through triangulation (Denzin, 2006). For this mean, a questionnaire was built based on the previous activity findings and distributed among practitioners.

APPLICATION OF VALIDATION QUESTIONNAIRE. The questionnaire respondents were screened from two research projects mailing list, CrESt[3] and SPEDiT[4], and two MBSE discussion groups from a business and employment-oriented social network[5]. Our anonymous questionnaire had three parts. In the first part, we asked for demographic data (*i. e.*, organization size (cf. Table 4.3), industry sector (cf. Table 4.4)) and added a yes/no question whether the respondent has ever participated or observed some endeavor to introduce Model-based Engineering in an organization. We used this question to exclude respondents without proper experience. We had 40 respondents in total; three were excluded.

Table 4.3: Organization size of the questionnaire respondents (i.e., number of employees).

Organization size	**Percentages**
1-20	13,5%
21-100	13,5%
101-1000	10,8%
>1000	62,2%

---

[3] `https://crest.in.tum.de/`

[4] `https://spedit.in.tum.de/`

[5] `https://www.linkedin.com`

Table 4.4: Organization size of the questionnaire respondents (i.e., number of employees).

Industry sector	Percentages
Automotive	35,1%
Others	29,7%
Academia	21,6%
Energy	8,1%
Manufacturing	5,4%

In the second part, we asked the respondents to express their agreement with each best practice (BP) using a Likert scale (Likert, 1932) with four levels: *Strongly Disagree, Disagree, Agree, Strongly Agree*. In the last part, we asked the respondents to select up to 5 BPs they considered most important.

Section 4.3 contains the output of all activities and respective discussions.

4.2.3 *Threats to validity*

The validity of our results is subject to the following threats:

Sampling bias. All interview subjects work in Germany (omission bias). They were selected from a convenience sample of project partners and professional contacts (inclusion bias). To mitigate these issues, we created the questionnaire and distributed it to a broader audience (i.e., triangulation).

Researcher bias. To mitigate this threat, the interviews were conducted by two researchers who took notes independently. Further, the interpretation presented in this chapter was validated with MBSE practitioners through a questionnaire.

Research method. To minimize misunderstandings between interviewees and the researchers, the study goal was explained to the participants before the interview. Steps taken to improve the reliability of the interview guide included a review and a pilot test. We followed several strategies as proposed by Maxwell (Maxwell, 2012) to mitigate threats.

External validity. We expect that our results represent the embedded systems industry; thus, we cannot generalize the results to other types of industry.

4.2.4 *Availability of data*

Due to the unreasonable effort necessary for anonymizing the interview transcripts, we do not disclose them. However, we disclose the interview guideline (cf. Appendix B), and the questionnaire (cf. Appendix C).

4.3 RESULTS AND DISCUSSION

In this section, we present and discuss our findings, which are shaped in the form of Best Practices and are classified into four groups:

- Piloting - what to care about at the first attempts
- Tool and process - selection and adoption decisions
- Knowledge building - development of team competence
- Management - human resources motivation and support

This section is divided as follows. The next subsection presents all best practices; the following four subsections describe the groups and associated best practices. We present quotes as evidence and the result of the questionnaire. We also discuss related existing work. In the last subsection, we present the Best Practices list with the top six most selected best practices.

4.3.1 *Summary of Best Practices*

The identified best practices and their respective groups are summarized below:

Piloting

BP01: The organization should start adopting MBSE with new projects.

BP02: The pilot project should create real value for the organization (i.e., no didactic project).

BP03: The pilot project should have enough budget and time allocated to bear the overhead of adoption.

BP04: No translation of old artifacts except for reusable artifacts.

BP05: Start small in terms of the project and team size in order to acquire some experience.

Tools and Process

BP06: Tools with open interfaces and homogeneous work-flow are preferred.

BP07: All engineers should have access to the tools.

BP08: Tool acquisition is very costly therefore should be thoroughly planned.

BP09: Have the new MBSE processes well documented so you better understand what tool you will need.

Knowledge building

BP10: All engineers should get, at least, basic training in MBSE.

BP11: Using examples that are familiar to the domain of the organization eases the understanding. Model some existing artifacts for use as examples.

BP12: Many strategies can be used to build knowledge of an organization, the context should be taken into consideration.

BP13: There should be a planned form of later evaluation to fill eventual gaps.

Management

BP14: Make the advantages of MBSE clear.

BP15: Have technically prepared people to support your engineers (i.e., not sales personnel).

BP16: Bring everyone to adoption (i.e., avoid creating castes).

BP17: If you have good engineers, let them do the work for you, it is cheaper, and they will engage more (i.e., empowering).

BP18: Management should unify all employees towards adoption.

4.3.2 *Piloting*

This subsection describes best practices related to the initial attempt to implement MBSE and how to harvest benefits from it. The questionnaire results for this group's best practices are depicted in Figure 4.2. The respondents agreed with most of the BPs. BP03 got the most agreement rate (95%) among its peers. On the other hand, 38% of the participants disagreed with BP04. We hypothesize that respondents that have no reuse culture in their organizations are likely to disagree with this BP.

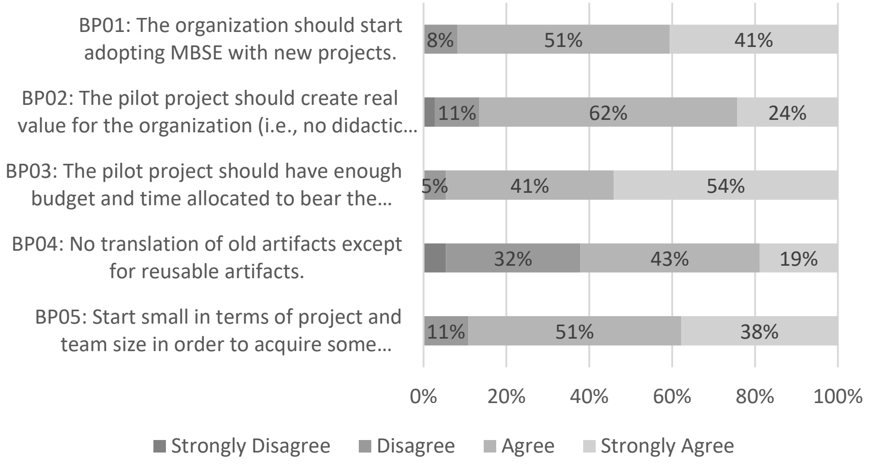


Figure 4.2: Questionnaire results for Piloting group.[6]

BP01: The organization should start adopting MBSE with new projects.

"New projects with MBSE" (P14)

"Do not recycle the past" (P9)

"No migration of old projects" (P8)

"New start in new project preferred" (P1)

"Set up own project for method development instead of accompanying from series development" (P3)

"Remove [team] from the existing organization, set up your project, the rest continues as before. Within a few years, new product generation has higher market share." (P10)

Introducing MBSE through new projects was a good practice pointed out by many interviewees. By doing so, the effort would not be wasted with the re-work of existing artifacts that might just be archived soon. Starting from scratch also prevents developers from shortcutting the workflow with information already documented, thus compromising learning the method (i.e., there is no document version for referencing the *possible* incomplete information). Additionally, already running projects have fixed budgets and deadlines that do not consider such overhead. Meeting those constraints gets more challenging when the engineers need to concentrate on learning MBSE.

"Legacy stay where they are" (P14)

---

[6] BP02 and BP04 received 3% and 6% respectively of 'Strongly Disagree'.

"Existing projects remained in old surroundings and were gradually archived" (P1)

"Backwards compatibility, so that only changes need to be redesigned (includes not only code, but also specs, tests, test infrastructure)" (P10)

As MBSE became more pervasive in the organizations, the old document-based projects were naturally phased out. Exceptions for this are Product Line intensive organizations or organizations with high reuse levels. In this case, translating the existing document-based artifacts that are meant to be reused to model-based artifacts is a recommended practice.

BP02: The pilot project should create real value for the organization (i.e., no didactic project).

"Start with a concrete project" (P9)

Learning MBSE through a real-life project with real deadlines and milestones is reportedly better than learning through mock-up projects and generic examples. The real-life setting boosts engineers' motivation since their work is already producing something useful. Additionally, real projects are related to the domain of the organization.

BP03: The pilot project should have enough budget and time allocated to bear the overhead of adoption.

"The first projects must be able to bear the burden." (P9)

"Business case because of investment hurdle in the beginning" (P12)

The effort needed by engineers to learn should be considered when deciding for the pilot project, which must bear the burden, i.e., managers should avoid using critical (time-wise, business-wise, technology-wise) projects. By doing so, managers can mitigate the risk of a project failing to meet its constraints or MBSE being not adequately implemented.

BP04: No translation of old artifacts except for reusable artifacts.

"Parts with reuse are taken over" (P14)

"It started with reverse engineering. The existing system was clustered according to the information model" (P7)

Despite the support for using new projects, converting the existing document-based artifacts prone to reuse was also a strategy presented by some respondents. Such artifacts are well detailed, allowing the developer to concentrate on the methodology rather than the content. Moreover, the newly created models can be compared to the existing document-based version to verify and validate them. It is crucial to choose artifacts reused in future MBSE projects; otherwise, the effort will be used for learning purposes.

BP05: Start small in terms of the project and team size in order to acquire some experience.

"Only 1-2 people, then the rest of the team" (P14)

"Start in some areas that you can better oversee [...] Best to start in individual areas according to company guidelines" (P6)

"Often pilot installation to gain experience, then expansion companies-wide" (P1)

"It is advisable to start with a partial introduction first, later through a consistent MBE. - needed breath" (P2)

"Introduction as 'submarine project' by a dedicated developer, then application in own department, now overarching roll-out." (P5)

"If necessary implementation strategy with 3-4 months observation of what the developer does, and then decide how to improve the process with MBE" (P2)

Starting with a small and highly motivated group of employees in a grassroots strategy fashion is likely to increase success. Such practice allows the implementation in an easily controlled setting and identifies characteristics exclusive to the organization (current processes and auxiliary tools) and its domain (development of DSL). Thus, issues can be identified with more precision and addressed promptly. Once the small group develops acquaintance with the new tasks and tools, the implementation should be rolled out to the rest of the team.

Related work. Investments in training, tools, and modeling environments should be considered in the projects selected for initially applying MBSE. Expectations should be handled, and proper time is given for the effort to pay dividends (Fosse and Bayer, 2016). The experience of Fosse and Bayer (Fosse and Bayer, 2016) made the authors support the "learning by doing" strategy in real-life projects by keeping the focus on project deliverables and modeling as needed. According to the authors, the pressure to deliver real engineering products forces discovery and resolution of problems not likely encountered in a didactic-only project. The authors stated that it is possible to yield communication and understanding improvement at an earlier stage by focusing on the description first and then analyzing it. Hutchinson et al. (Hutchinson, Whittle, and Rouncefield, 2014) state that "starting small" when adopting MDE and then growing, thus committing more resources on a broader scale, is a helpful response towards MDE adoption. Hammer (Hammer, 2007) states that one unit's pioneering positive experiences can energize an entire organization.

4.3.3 *Tools and process*

Tools are an essential cornerstone of MBSE. Traceability, simulation, automatization of tasks are among the things that are only possible with the use of specialized tools. However, tool licenses are expensive, and new tools require learning investment. Rolling back tool acquisition is very costly; thus, wise decisions can save time and resources. Proper tool selection requires identifying what capabilities the organization would like to develop by implementing MBSE and acquiring the tools to help it fulfill its goal.

Figure 4.3 shows the survey results for Tools and process best practices. We can see a reasonable agreement rate among the respondents and very few disagreement responses.

BP06: Tools with open interfaces and homogeneous work-flow are preferred.

"Customers want homogeneous toolchain but open interfaces" (P1)

"Often the entire toolchain is replaced" (P1)

Because MBSE is complex and encompasses the complete system development life-cycle, supporting toolchains are often equally problematic. Several tools are currently required to do the job because existing ones are not developed enough to cover the whole life cycle. For this matter, import and export features must allow the work to continue in further design phases without much ado; thus, interoperability through open interfaces becomes desirable. By seeking open interface tools, the organization protects itself from many problems using exclusive proprietary tools.

BP07: All engineers should have access to the tools.

"The worker level must also get the tools" (P9)

"Tool usage for all developers who use the tools" (P6)

MBSE tools should be widely available for all engineers. Failing to do so can create two classes of engineers, and this could jeopardize the adoption. However, some engineers are mainly producing models while others are mostly consuming them for some activity. Therefore, understanding how the tools and models are going to be used can lower acquisition costs.

BP08: Tool acquisition is very costly therefore should be thoroughly planned.

"Develop an adoption strategy" (P8)

"Development of a holistic approach/methodology" (P1)

Defining the new processes well and in advance is highly recommended. Planning should be done considering the full implementation of the processes and tools. This measure avoids realizing at a later point in time that the decisions taken when planning the MBSE adoption (tools to be acquired, processes to adopt, training contents) are not fit for further intended process maturity. This mishappen's consequences can be manifold: from extra effort to work out tool inter-interoperability, until costs with redundant tool acquirement and waste of time with training.

BP09:Have the new MBSE processes well documented so you better understand what tool you will need.

"If the standard process is well documented, the MBE implementation will work easier." (P1)

Organizations that have their current development process described and documented as it is performed have less trouble when implementing MBSE. The tasks and artifacts to be replaced can be straightforwardly identified, and the consequences of the change are easily identified.

"First the process was set, then the tool decision" (P7)

In MBSE, tools and processes are highly intertwined. It is not uncommon that tool manufacturers also provide the process workflow specially designed to be used with their tool. Thus, tool selection is also a contributing factor when choosing which MBSE process to adopt.

"Tool support for automation of work" (P12)

Automation of work is one of the best features provided by MBSE. The formality of models allows tools to automatically verify many properties that otherwise would require intensive human labor. While some capabilities are indispensable, others might be just nice to have. Understanding which features bring the best benefits to the organization helps achieve a better cost-benefit relation when deciding on tool adoption.

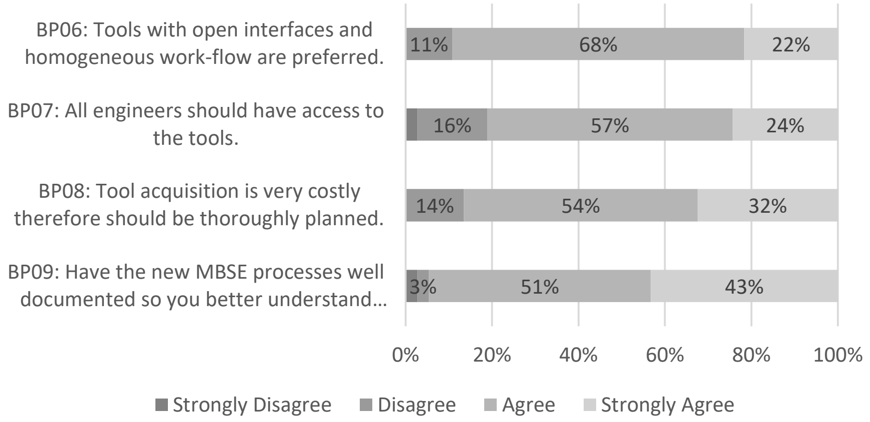


Figure 4.3: Questionnaire results for Tools and process group.[7]

Related work. In the Jet Propulsion Laboratory of NASA (Fosse and Bayer, 2016), MBSE teams were organized in a 3-tier fashion, with a small set of core modelers, a large set of modeling-savvy SEs, within a large set of project personnel. In this sort of team arrangement, everyone needs some training, but not to the same depth. The authors of (Carroll and Malins, 2016) cite three points that must be taken into account: (1) the existing process should be well documented and should cover the whole life-cycle otherwise is quite complicated to understand what needs to be adapted, (2) the MBSE model management processes (i.e., creation, update, and maintenance of models) should be defined thereto derive engineering artifacts, (3) investment in full-scale MBSE tool (i.e., not RE management tools) that are accessible to all team members is necessary. Whittle et al. (Whittle et al., 2017) stress the importance of social and organizational factors for the selection and development of tools. They found in their study that it is better to "Match tools to people, not the other way around" and to "more focus on processes, less on tools". Our interviews' provided answers reflect the opinion that the exchange of information (i.e., models) between tools is a problem that must be addressed by proper import/export mechanisms and open interfaces. In a previous study, we have seen that tool incompatibility is one of the significant forces that hinder companies from adopting MBSE solutions (Vogelsang et al., 2017). Research has developed alternatives that may fit the MBSE paradigm better. Seamless MBSE may be supported by multiple tools manipulating models in a common model repository instead of importing/exporting models (cf. *projectional editors* (Voelter et al., 2014)).

---

[7] BP07 and BP09 received 3% of 'Strongly Disagree' each.

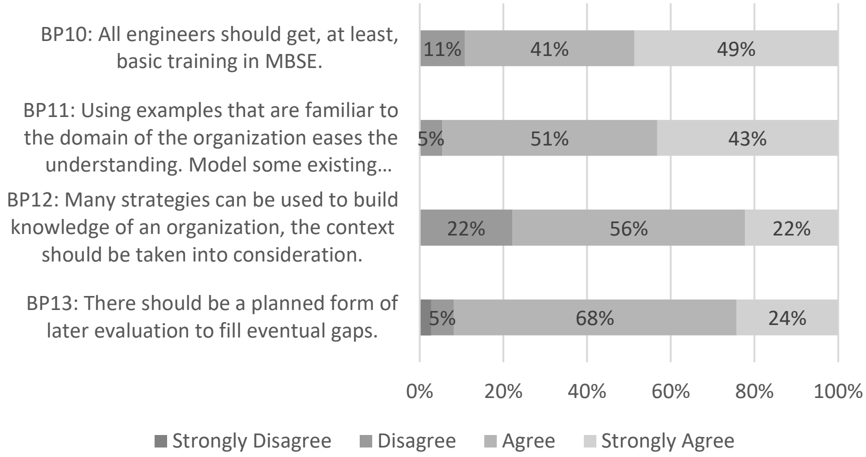


Figure 4.4: Questionnaire results for Knowledge building group.[8]

4.3.4 *Knowledge building*

Training can be carried out in many ways. In this subsection, we will present different views and strategies, which sometimes can also be contradictory. This phenomenon does not invalidate them, as the reader should realize that the organization's context influences the best approach. The questionnaire answers had an excellent agreement rate with the best practices, the minor being 78% rate of 'Agree'. The exception is BP12, which received 22% 'Disagree's. (cf. Figure 4.4).

BP10: All engineers should get, at least, basic training in MBSE.

"Training, training, training!" (P1)

"Basic training for all users of MBSE" (P6)

"Everyone who uses MBSE should be trained in the methodology" (P8)

"Broad basic training of all employees - Everyone should have the same understanding" (P9)

Basic training should be provided to all employees. Nevertheless, the required knowledge will vary among team members (i.e., not everyone will require the same type of knowledge or in the same depth). For instance, developing complete models requires much more in-depth knowledge than only understanding them.

---

[8] BP13 received 3% of 'Strongly Disagree'.

BP11: Using examples that are familiar to the domain of the organization eases the understanding. Model some existing artifacts for use as examples.

"Examples from similar industry help" (P9)

"Catching up with experiences/feedback from the network (from the same industry)" (P13)

"[Training team] has modeled examples of clients and then presented them [sic] to the client (we will show you how we do it)" (P2)

The overall understanding can be enhanced by modeling some requirements that are familiar to the developers. The effort required to understand what is being modeled (i.e., domain) is diminished; thus, trainees can concentrate solely on the modeling concepts. If an organization has no prior knowledge in MBSE or modeling in general, it helps when some experts model small parts of existing systems from the organization as examples to transfer and demonstrate the idea of MBSE.

BP12: Many strategies can be used to build knowledge of an organization, the context should be taken into consideration.

"F2F, except perhaps for basics" (P6)

"Face-to-Face Seminar with exercises" (P7)

"Deeper training of individual employee, who then pass on their knowledge in the team" (P9)

Depending on the team dynamics, it is possible to concentrate the training on a small group of participants, and later those people will be responsible for disseminating the knowledge to the bigger group. This strategy fits very well when the piloting is also performed with a small part of the team. Other interviewees said that training should be done with face-to-face seminars followed by exercises. As we can see, both approaches seem to work; therefore, the context should drive such type of decision.

"Per tool training e.g. Doors[9]" (P7)

"Model should be the reference. To do this, connect other tools (e.g., importing Doors[9] for initial filling, then only update exports back to Doors, generated code frames force interface fidelity)" (P5)

Tool and process are very intertwined in MBSE; thus, training should mind the tools used. The training should emphasize the model, showing there should not be co-existing two artifacts with the same purpose, namely a document-based version and a model-based ver-

---

[9] https://www.ibm.com/us-en/marketplace/rational-doors

sion. The verbatim above gives the example of a requirements modeling tool (Doors).

"Strategic cooperation, introductory consulting [...] Coaching during the introduction has proven very successful" (P1)

Training provided by external consultancy guarantees a minimal homogeneous level of knowledge throughout all employees. Experienced consultants are very efficient and can transmit knowledge very efficiently. However, one should have in mind that participant P1 works with technical sales.

BP13: There should be a planned form of later evaluation to fill eventual gaps.

"Continuous explanation of the methodology" (P1)

"Accompanying technical monitoring of work results; is not happening systematically." (P5)

Training is an ongoing process, which includes returning to training concepts at a further point in time and assessing the engineers' knowledge (e.g., through the quality of the artifacts being produced). This procedure is necessary to be put into practice and is not happening systematically.

Related work. The training activities should use real examples since they are much more effective at conveying understanding and building support (Fosse and Bayer, 2016). According to (Carroll and Malins, 2016), the organization should nurture a cadre of trained systems engineers with at least moderate skill in employing MBSE tools and techniques and whose MBSE roles are delineated from the more traditional roles. For the engineering staff a basic level of training in the MBSE processes (so that they understand the value of the models and what to expect from the systems engineers) and how to read MBSE artifacts (so that they can interpret the information provided from the MBSE processes). Although the awareness towards training, 62% of "MBSE active users" surveyed by (Motamedian, 2013) never received official training from their organization, and just 2% had a complete technical course sponsored by their organization. In line with the reported best practices, several works exist that report on the efficient transfer of MBSE methodologies by remodeling parts of existing systems from companies (e.g., (Böhm et al., 2014)). Tool vendors usually provide training for the methodology and workflow to be used with their products (Hoffmann, 2012).

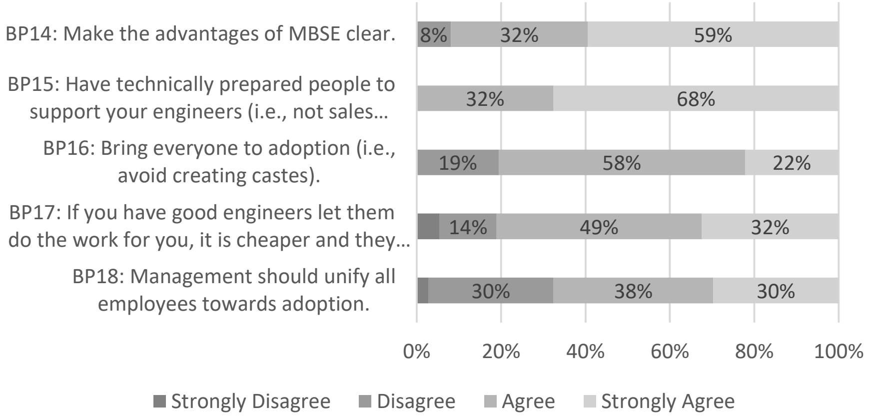


Figure 4.5: Questionnaire results for Management group.[10]

4.3.5 *Management*

Although the current state of the practice has achieved some degree of automatization in systems engineering, its tasks are still human-intensive. Thus, introducing process change in an already running and established organization is not frictionless (Conner, 1993; Hammer, 2007). In an ideal world, everyone would jump in the adoption boat without hesitating and be fully motivated; the reality says that management should care for the team vibe towards the adoption. Micromanagement is required to tackle hurdles and psychological barriers, especially when it comes to such a complex change as MBSE adoption (Vogelsang et al., 2017). This section is about strategies to manage the overall mood and expectations of the ones affected by MBSE adoption.

The questionnaire results showed that BP16, B17, and BP18 received many disagreement votes. Although most of the respondents still agreed with them (i.e., at least 68% of agreement rate). One hypothesis is that they are context-dependent. On the other hand, BP14 and BP15 got almost no disagreeing votes. (cf. Figure 4.5)

BP14: Make the advantages of MBSE clear.

"Advantages of simulation were made visible" (P11)

"Good examples show: project for new methods tools, then show that it works (utility and acceptance)." (P13)

---

[10] BP17 and BP18 received 5% and 2% respectively of 'Strongly Disagree' each.

"Representation as a means of unique selling proposition. Works, but each developer must be convinced individually." (P10)

The way people perceive the MBSE introduction can become an obstacle; thus, it is imperative to manage the mood of those involved, especially in the very beginning, when everything is experimental. The advantages of the approach should be made clear, also emphasizing benefits for the employees. By doing so, organizations can foster engineers' acceptance and collaboration towards MBSE adoption.

"Every few years new product generation, e.g., because of Platform changes through technical advances (multicore ...) or new system approaches that affect many functions. Then there is a break in the system concepts; many have to be taken in hand because you can introduce new methods." (P10)

As technology evolves, new paradigms need to be adopted to keep up with the market. Pairing such changes with MBSE adoption improves its acceptance by surfing on the already needed change atmosphere. However, a paradigm shift may be time-consuming; thus, this strategy is not fit for all situations.

BP15: Have technically prepared people to support your engineers (i.e., not sales personnel).

"Tool sellers sometimes only send sales personnel, can not answer technical questions, does not make a good impression." (P13)

The engineers will have specific technical doubts about the modeling environment. Therefore, having prepared people to support them is fundamental. Sales personnel are usually not prepared enough for this task, and failing to do so might raise skepticism towards the adoption.

BP16: Bring everyone to adoption (i.e., avoid creating castes).

"Avoid living apart, everyone has to go!!!" (P9)

Other best practices enforced that all employees should receive access to the tools (BP07) and basic training (BP10). The problem of not doing so is creating two classes of engineers (i.e., those working with MBSE and those that are not), which can jeopardize their motivation. Albeit, it is not uncommon for teams working in different technologies in the same organization. Another exception is when the strategy trains few engineers, which will pass the knowledge to the others. This point should be considered with reservations.

BP17: If you have good engineers let them do the work for you, it is cheaper, and they will engage more (i.e., empowering).

"Information model developed itself" (P7)

"[The knowledge was created by the] developers available or self-built (i.e., only UML and the DSLs)" (P4)

Some organizations had their developers learning about models autonomously (i.e., the knowledge was organically developed). That is not so surprising since, in technological areas, engineers need to be learning new technologies all the time.

BP18: Management should unify all employees towards adoption.

"Guidance from management important to bring everyone on the same platform" (P6)

Implementing MBSE involves engineers from different phases of software development and different fields of engineering. These professionals are usually not working together on an everyday basis. Filling this gap is the duty of management, thus uniting the whole team towards MBSE adoption.

Related work. In a previous work (Vogelsang et al., 2017), we identified forces that work towards *hindering* or *fostering* the adoption of MBSE and their origin. Hindering forces can be classified as either Inertia or Anxiety. The former is triggered by the feeling that the current solution is "good enough" and habits that keep people from trying out something new. The latter is triggered by fears that MBSE adoption will not pay off, mainly caused by uncertainties and perception flaws (Vogelsang et al., 2017). Motamedian (Motamedian, 2013) had in his survey questioned subjects about barriers using MBSE. The second and third reasons most voted were "the lack of perceived value of MBSE" and "resistance to change", the need to micromanage the engineers' expectations and motivation toward the process of adoption. Some ideas are just standard process-changing measures with MBSE flavor, which was something also found by Hutchinson et al. (Hutchinson, Whittle, and Rouncefield, 2014) Nevertheless, the emphasis and alignment with other phenomena increase their relevance thus are worth mentioning. The challenges that arise (Vogelsang et al., 2017) are due to the nature of the current context, the shift to MBSE (e.g., compile development skills vs. abstraction skills), and the human-in-the-loop. Hammer (Hammer, 2007) pointed out that the adoption initiative has to have an owner (*e. g.*, a senior executive) who has the responsibility and authority to ensure that the process delivers results. He also said that efforts that lack the backing of senior executives are likely to fail.

4.3.6 *Most important Best Practices*

Besides validating our interview findings, we used the questionnaire to discover which best practices are considered the most important. For this means, we asked the respondents to select among all BPs the five most important when adopting MBSE. The results can be seen in Figure 4.6. Of the six most selected, three were from the Piloting group, one from the Knowledge building group, and two from the Management group. The BPs are discussed in the following under this light:

bp14 ***Make the advantages of MBSE clear.*** The most important BP is related to increasing the engineers' motivation towards MBSE adoption by making them understand its benefits. Engineers are less likely to withstand the hurdles of adoption if they cannot perceive its benefits (Vogelsang et al., 2017). Systems Engineering is a human-intensive activity; thus, the engineers' collaboration is crucial for adoption success.

bp05 ***Start small in terms of the project and team size in order to acquire some experience.*** This best practice is about using a small setting to understand the best way to introduce MBSE considering the organization context. Through experimenting, managers can understand which tools, languages, and styles are best fitted for the organization and its respective domains and current processes.

bp02 ***The pilot project should create real value for the organization (i.e., no didactic project).*** This best practice is about making the adoption efforts meaningful and relevant right from the beginning. By working into something used in a production setting, the employees have to learn and employ MBSE. If it is just for learning, there are no consequences if the project is incomplete or not well done, making the learning incomplete. It also gives room for procrastination (i.e., learning later when it is necessary).

bp03 ***The pilot project should have enough budget and time allocated to bear the overhead of adoption.*** This BP relates to the effort required by the engineers to develop their MBSE skills. The project selected to pilot the adoption should have its budget planned to cover the learning curve costs, and the delivery of its artifacts should be planned accordingly. Time-critical projects should be avoided. If the project selected is not fit, the engineers will drop the MBSE techniques in favor of already established

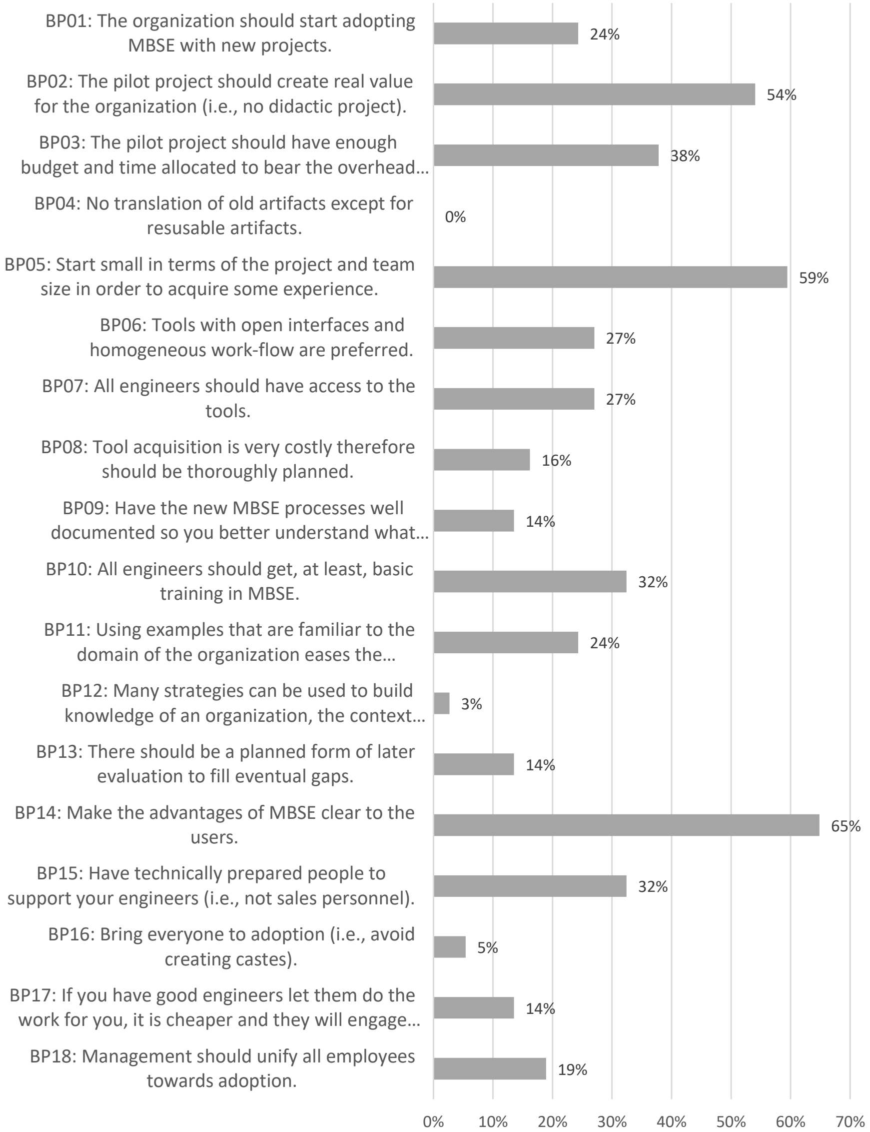


Figure 4.6: Percentage of respondents rating each best practice as one of the "five most important."

development methods to achieve celerity gains and meet deadlines. The adoption is very likely to fail.

bp10 ***All engineers should get, at least, basic training in MBSE.*** In a system development project, different skill sets are mastered by different professionals who specialize in specific parts of the life-cycle (e.g., development, testing, design). Despite their skill focus, all professionals have an overall system development knowledge, which helps them understand the big picture, and the same should happen with MBSE. Although not all engineers require excellent modeling skills, everyone should be able to, at least, read and understand the models.

bp15 ***Have technically prepared people to support your engineers (i.e., not sales personnel).*** Providing appropriate support to engineers helps to speed up adoption. This practice also hinders possible frustrations during learning, thus keeping their morale high and transmits the message that they are not on their own.

As for the least important best practices, *BP04: No translation of old artifacts except for reusable artifacts* was not selected as "most important" by any of the respondents. *BP12: Many strategies can be used to build knowledge of an organization, the context should be taken into consideration* was selected only once.

4.4 USING BEST PRACTICES TO MITIGATE MBSE ADOPTION HINDERING FORCES

This section relates the strategies and best practices presented in this chapter with the hindering forces towards MBSE adoption from Chapter 3. Our goal is to propose mitigation strategies for the hindering forces using the elicited best practices.

The interview participants from this chapter study are a subset of the Chapter 3 study. We correlate the issues (*i. e.,* hindering forces) with the solutions (*i. e.,* best practices) when the respective codes emerge from the same participants during the interviews. We assume that the person experiencing and relating the issues have better knowledge to propose solutions. For instance, interview participant P11 provided the following statement: *"The first projects must be able to bear the burden"*, which was used as evidence for the Best Practice *BP03: The pilot project should have enough budget and time allocated to bear the overhead of adoption.*. The same participant stated: *"It is a mistake to believe that an advantage will be felt in the first project"*, which was used to determine the anxiety hindering force *ROI uncertainty*. Thus,

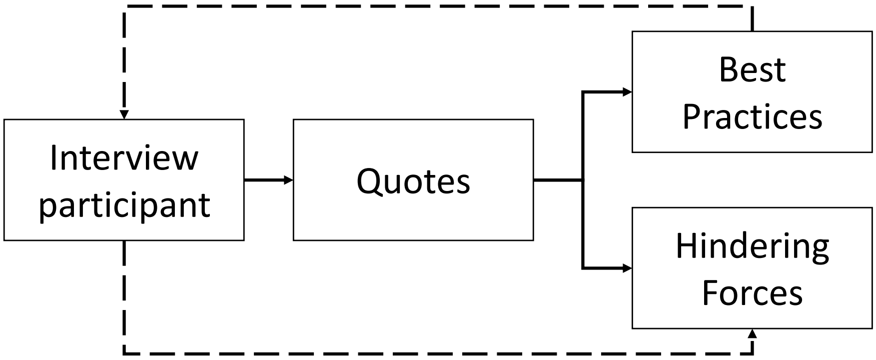


Figure 4.7: Relation between hindering force and mitigating best practice.

we claim that BP03 can be applied to mitigate the anxiety hindering force *ROI uncertainty*. The relation between hindering force, best practice, and interview participant is depicted in Figure 4.7. The relations that could be drawn this way between best practices and hindering forces are listed in Table 4.5. Not all best practices are present in the aforementioned table because we could not find evidence in the interviews relating them to hindering forces. From the 18 identified best practices we could only find evidence to relate 13. We also did not find any evidence for a best practice addressing the *No support* and *Bad experiences* inertia forces. Moreover, the relations presented in this table are not comprehensive.

All interviews' verbatim used as evidence for the correlations present in Table 4.5 can be seen in Appendix F. This contribution extends the previously presented research workflow (cf. Figure 4.1) merging it with the tasks from Chapter 3 and adding a new task, namely *Analysis between BP and Forces*, as can be seen in Figure 4.8.

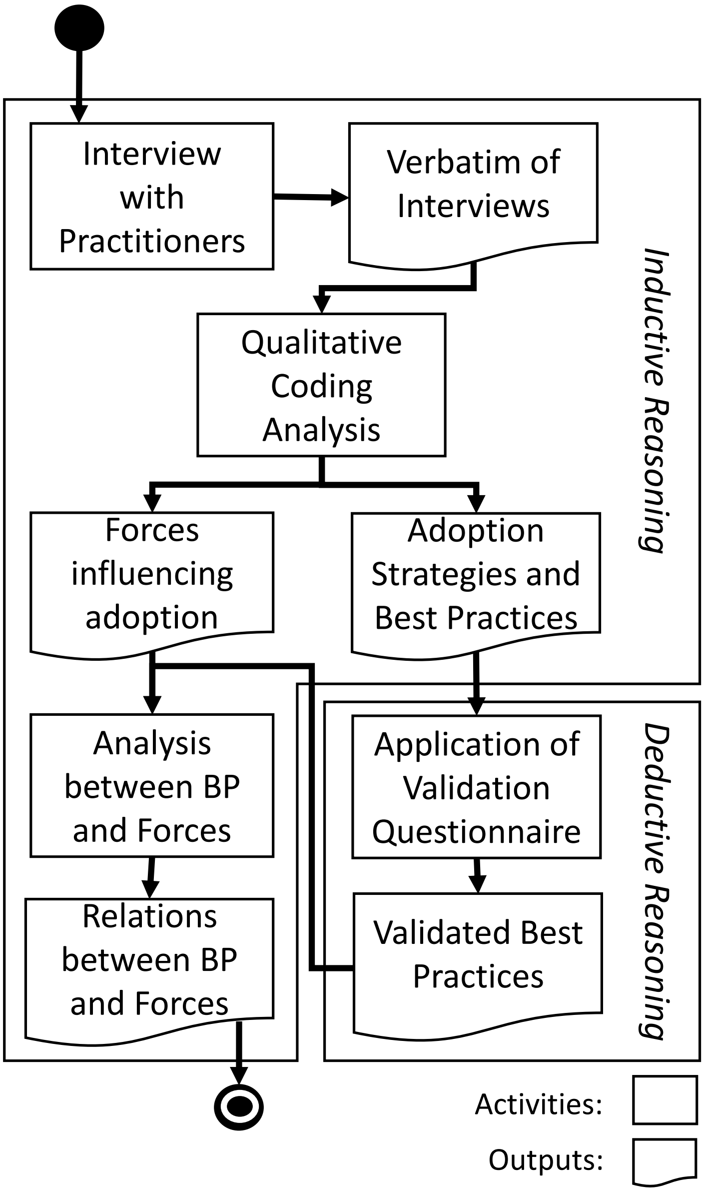


Figure 4.8: Expanded research method workflow.

Table 4.5: Best practices addressing MBSE adoption hindering forces.

Forces / Best Practices	BP01	BP03	BP05	BP07	BP08	BP09	BP10	BP11	BP12	BP13	BP14	BP16	BP17
Inertia Forces													
Doubts about improvements	x		x			x		x			x		
Immature tooling	x		x				x						
Incompatibility with existing tools	x		x										
Changing the way of thinking							x	x			x		
Insufficient training							x		x	x			x
Bad experiences													
Learning new tools	x				x		x				x		
Immature methodologies	x		x				x						
Migration is needed	x												
Development process does not fit		x											
Missing excitement					x		x						
No support													
Anxiety Forces													
ROI uncertainty		x	x		x	x		x			x		
Skills of employees	x		x								x		
Tools shortcomings	x				x	x							
Methodology shortcomings	x		x										
Large training efforts									x				
Role misunderstandings	x			x			x					x	
Migration issues			x								x		
Team Competence loss	x			x			x					x	

5

METHOD VALUE ASCRIPTION

When the options are numerous, knowing how to ascribe value properly is fundamental to achieve optimal decision-making results. Projects failing to do so can incur undesirable consequences, ultimately hindering goal achievement. An instance of this problem is MBSE method selection. Lacking the needed domain knowledge to choose methods and poorly defined adoption goals can lead to the selection of *quick-wins* that are sure to achieve successful implementation but without necessarily addressing the motivation for the adoption itself. In order to avoid such pitfall, we need to model the relevant elements related to the method value ascription process. Our goal is to provide a model which makes explicit much of the knowledge involved in the method value ascription process, *i. e.*, value ascription mechanisms, motivational and influencing factors.

5.1 INTRODUCTION

MBSE encompasses many methods used throughout the whole system development life cycle. Deciding which ones are the most interesting for a systems development team requires their proper value appraisal. Thus, understanding the mechanics of method value ascription helps to produce better decision-making for an effective MBSE adoption.

Value is a relational and emergent characteristic. It is relational because an object has value when ascribed by an agent and is emergent because it emerges from how well something affordances match the goals and needs of a given agent in a given context. Therefore, an object may have different values to different agents, or even according to different goals of the same agent (Sales et al., 2018).

Value is represented by the *Valuation relationship* element (cf. Figure 5.1) which connects the *Team* and *Value method*. The Valuation relationship is made of *Costs* and *Benefits*. The latter is related to how methods' intrinsic characteristics help the teams achieve their adoption goals within a specific context. The former are expenditures required to implement and execute the method. We consider value as in utility value, to not mistake with other kinds of values (*e. g.*, exchange value).

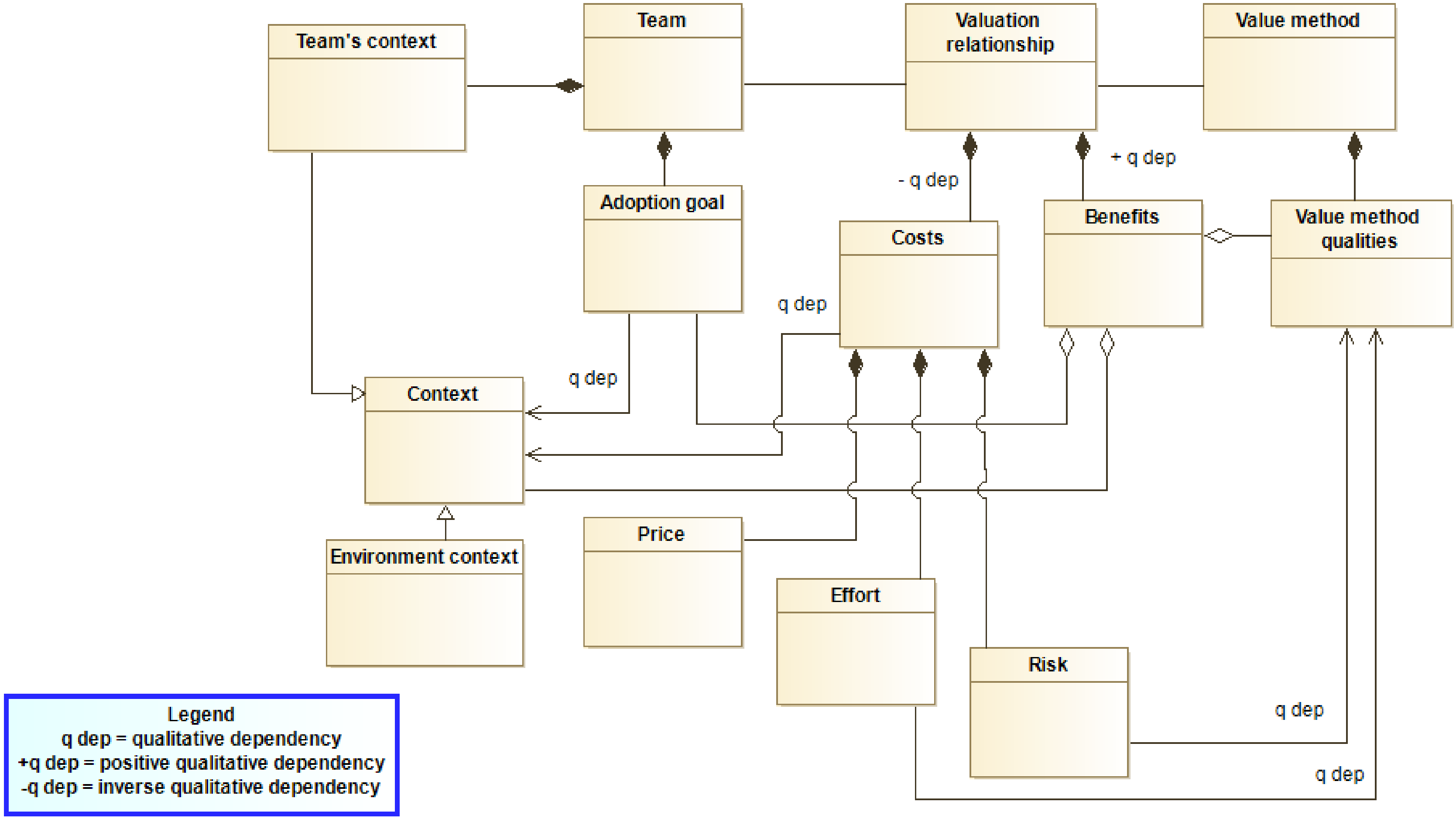


Figure 5.1: Method value ascription model.

5.2 CORE CONSTRUCTS

In this section, we describe the core constructs of the method value ascription model, which is depicted in Figure 5.1. In the following subsection we describe the elements of the model and next we describe the characteristics of their relations.

5.2.1 *Elements*

TEAM. is a development team ascribing value to the *Value method* which is characterized by the *Valuation relationship* element. It also defines the *Adoption goal* and has a *Team's context*.

VALUE METHOD. is the object of value ascription, *i. e.,* the method undergoing evaluation. It is related to the *Team* through the *Valuation relationship* and it has *Value method qualities*.

VALUATION RELATIONSHIP. is a significance attached to a *Value method* by the *Team*. It is composed of the Team perception of *Costs* and *Benefits* related to the Value method adoption.

VALUE METHOD QUALITIES. are intrinsic properties of the *Value method* which are independent of the *Context*. It has direct influence on the perception of *Benefits*. For instance, the acquired benefit of improved communication upon the adoption of a method is a method quality.

ADOPTION GOAL. is the reason why a *Team* ascribes value to a *Value method*. It is influenced by the *Context* and it composes the perception of the *Benefits* by the Team.

CONTEXT. is the interrelated conditions in which something exists or occurs (Merriam-Webster, 2021). It influences the value ascription because they change the *Benefits* perception, the *Adoption goal* definition, and the *Costs*. Context is divided into two sub-types, namely *Environment Context* and *Team's Context*, which are described in detail in the following:

- **Environment context** represents everything outside the boundaries of the *Team* and the *Value method* (*e. g.,* stakeholders, communicating systems, regulations, market competitors).

- **Team's context** are Team characteristics (*e.g.,* size, employed methods, members experience, employee turnover).

In general, a team has little influence in the environment context while having a high capacity to change its context.

BENEFITS. are relational properties from *Value method qualities* within a *Context* that allows a *Team* to achieve an *Adoption goal*. Together with the *Costs*, it composes the *Valuation relationship*.

COSTS. are the resources the *Team* must concede in order to adopt and put into practice the *Value method*. Together with the *Benefits*, it composes the *Valuation relationship*. Costs can be of three different types:

- **Price.** is the monetary expenditure required to have the method implemented in a team (*e.g.,* tool acquisition, training, infrastructure, licenses).
- **Effort.** is the workload required to learn, put into use, or acquire the value method. It has qualitative dependency on the *Value method qualities* and can be of the following types (Kambil, Ginsberg, and Bloch, 1996):
  - **Acquisition effort:** the time and cost needed to search for, evaluate and get the value object.
  - **Operations and maintenance efforts:** the maintenance and disposal costs, and the time to learn how to use the value object, the wait for it to perform, and monitoring.
  - **Complementary effort:** The time and cost needed to find and acquire complementary products or services associated with the method and its respective adoption goal.
- **Risk.** This cost is related to the probability of not fulfilling the goal with the *Value method* or the sacrifices to be more strenuous than predicted. It has qualitative dependency on the *Value method qualities* and can be classified in the following dimensions (Kambil, Ginsberg, and Bloch, 1996):
  - **Safety:** physical risks related to the application of the *value method*.
  - **Financial:** the risk that the price paid is higher than usual.
  - **Selection:** the risk of not choosing the best alternative for fulfilling *Adoption goal*.

  - **Delay**: the risk that the *Value method* will take more time than expected to be implemented or it will not perform on-time, thus incurring opportunity costs.
  - **Functional:** related to the risk that the *Value method* will not perform as predicted, now or in the future.

5.2.2 *Relations*

Three labels are used in the model's associations, namely *q dep*, *+q dep*, and *-q dep*. The first characterizes relationships with a qualitative influence on other elements, either positive or negative, the second label represents positive qualitative influence, and the third is used to represent negative influence relationships. For instance, *Costs* has a *-q dep* relation to the *Valuation relationship* while *Benefits* has a *+q dep* relationship. *Context* has a *q dep* relationship since it can have either positive or negative influence.

5.3 RELATION TO STRATEGIES AND BEST PRACTICES FOR MODEL-BASED SYSTEMS ENGINEERING ADOPTION

The benefits of employing empirically validated strategies and best practices in order to address MBSE adoption challenges are thoroughly discussed in Chapter 4. Their relation to the method value ascription model is represented through the inverse qualitative dependency (*i. e.*, *-q dep*) that the *Costs* element has with the *MBSE Strategies and Best Practices*. When development teams follow those, the *Effort* required to adopt a method is diminished since efficiency is increased, the associated *Risk* are mitigated, and the *Price* for adoption is diminished (*e. g.*, less expenditure in training). The aforementioned relation can be seen in Figure 5.2.

5.4 RELATION TO FORCES THAT DRIVE OR PREVENT MODEL-BASED SYSTEMS ENGINEERING ADOPTION

The MBSE adoption forces (cf. Chapter 3) are emergent motivations originated from the method adoption endeavor, thus consequences of its dynamics. The forces relations towards the method value ascription model elements are described in the following:

- *MBSE Internal Push Force* originates from within the organization, thus receiving positive qualitative dependency from the *Team's context*.

- *MBSE External Push Force* is directly influenced by the *Environment Context* since factors outside the organization boundaries boost this force.

- *MBSE Pull Force* has direct qualitative dependency relation to the perceived *Benefits* upon method adoption.

- *MBSE Anxiety Force* is related to the *Risk* element. Fears that the new MBSE methods will not be able to fulfill the expectations fuel this force.

- *MBSE Inertia Force* is related to the *Effort* element since the efforts associated with the adoption of new MBSE methods boost this force. The model depicts this through a positive qualitative relation with *Effort*.

The aforementioned relations are depicted in Figure 5.2.

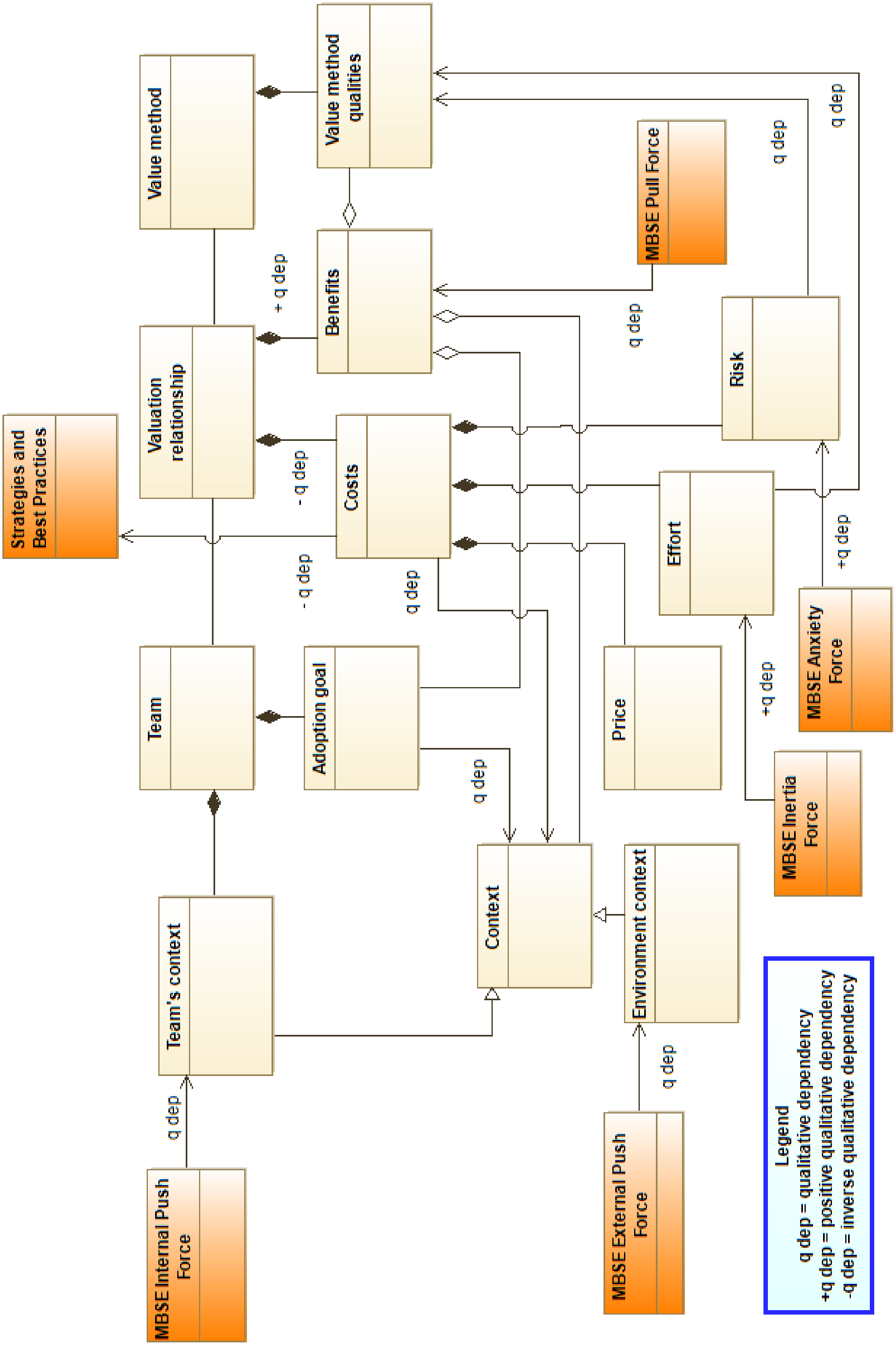


Figure 5.2: *Forces on MBSE adoption* and *Strategies and Best Practices* in relation to the model.

6

GOAL-BENEFIT DECISION CALCULUS FOR MODEL-BASED SYSTEMS ENGINEERING METHOD SELECTION

Model-based systems engineering (MBSE) methods can address many development challenges of modern systems. However, due to its comprehensive coverage, teams are struggling with selecting methods best aligned with their adoption goal, *i.e.,* not all methods are fit to deliver the pursued benefits. This chapter aims to relate goals that drive MBSE method adoption and candidate solutions, thus providing suggestions that highest yield the expected benefits. For this means, we propose a goal-benefit model and respective operationalization method. The model relates MBSE methods with benefits generated upon their adoption and finally with adoption goals. Our approach results in a prioritized list of MBSE methods ready to be transformed into an improvement roadmap. The approach was applied in six development teams located in Germany and Brazil. We also provide a sensitivity analysis to validate the model quantitatively. The approach was assessed positively by the practitioners involved in our case study, who stated that it supports the goal-oriented MBSE method adoption and improvement process.

6.1 INTRODUCTION

Process change is challenging (Conner, 1993; Hammer, 2007), especially for complex changes such as adopting MBSE whose numerous methods and processes encompass methods in all phases of the product development life-cycle (Amorim et al., 2019; Vogelsang et al., 2017; Whittle et al., 2017).

Teams adopt MBSE methods for many reasons. Sometimes their goal is to enhance the developed system's properties (*e.g.,* quality, desirable functionalities). Perhaps they pursue to increase the development process' efficiency (*e.g.,* cost, lead time). Others might need to tackle problems related to the complexity of new systems (*e.g.,* interdisciplinary development, the massive amount of text-based specifications) (Vogelsang et al., 2017).

Managers are unsure which MBSE methods to adopt. They might be tempted to go for a *low-hanging fruit* strategy, which is cheap, easy, and has high implementation success. However, considering goal satisfaction, will the expected benefits be harvested? As a result, many teams make little progress in their attempts to transform business processes (Hammer, 2007).

Understanding the team adoption goal towards MBSE adoption helps identify which methods bring the most benefits and build a strong business case for the initiative. According to a report released by the Project Management Institute (PMI)[1], projects and programs that are aligned with an organization's strategy are completed successfully more often than projects that are misaligned (77% vs. 56%). At the same time, only 60% of strategic initiatives meet their original goals and business intent. The report states that most executives admit that their organizations fall short and that there is a disconnection between strategy formulation and implementation (PMI, 2017). Nevertheless, linking adoption goals to methods is not straightforward. Goals are usually formulated on a high level of abstraction, which leads to myriads of ways to accomplish them.

The Model-based systems engineering maturity model (MBSE MM) is a possible solution for this problem (cf. Section 2.4). It provides a framework to assess the capabilities implemented in a development team. From the non-implemented capabilities, managers can choose which ones to adopt. However, how to choose the methods is not part of the MBSE MM, *i. e.,* it lacks built-in contingency to guide tailoring (Conboy and Fitzgerald, 2010) such as a decision calculus (Little, 1970; Pöppelbuß and Röglinger, 2011). The absence of these mechanisms diminishes the MBSE MM efficacy and can hinder its use.

Therefore, we propose a decision calculus approach to prioritize candidate capabilities from the MBSE MM according to the development team adoption goal and context. The approach is composed of a model and an instantiation method. The model breaks down the adoption goal into so-called competitive priorities (Krajewski and Ritzman, 2018), which are related to benefits afforded by capabilities upon their implementation. The instantiation method guides the weight assignment of the model relations according to the relevance of the elements towards their neighbors. According to the assigned weights, points are distributed along the resulting graph, from the adoption goal to the competitive priorities, benefits, and candidate capabilities. As a result, candidate capabilities are prioritized according to the overall points they collect, reflecting their relevance towards the adoption goal and the team's context. With this approach, we want to support teams choosing MBSE methods to adopt. The main contributions of this chapter are:

---

[1] `https://www.pmi.org/`

- A goal-benefit decision calculus approach to support MBSE methods selection. The approach is composed of:
    - A goal-benefit meta-model that relates MBSE methods with benefits, competitive priorities, and finally with the adoption goal.
    - A method for instantiating the meta-model through assigning quantitative impact measures its relations, which is used to derive a prioritized list of MBSE process improvement candidates.
- Qualitative and quantitative evaluations of the approach based on six case studies and an analysis of the model.

6.2 STUDY APPROACH

In this chapter, we pursue to answer the following research question:

How to select model-based systems engineering methods that highest yield expected benefits?

We answer this question through the creation of artifacts, namely a model and a method to operationalize the model. The approach must represent the relations between what is intended to achieve through MBSE adoption and the possible ways to do so. The next logical step is to evaluate if the proposed solution is fit. For this means we will use the Guidelines of Modeling (Becker, Rosemann, and Uthmann, 2000) which prescribes guidelines to ameliorate the quality of information models. The framework has three basic guidelines, namely *economic efficiency*, *relevance*, and *correctness*, which will be used as design requirements that the resulting approach of this study must fulfill. In the following we derive requirements from the respective guidelines:

R1. The approach must be cost-effective.
    - *Verification:* feedback from case study participants.

R2. The approach must react to different types of relationships between input and output variables.
    - *Verification:* one-at-a-time sensitivity analysis (Saltelli, 2008).

R3. The approach must improve decision-making.
    - *Verification:* feedback from case study participants.

The research described in this chapter was conducted in an iterative fashion, which was composed of the following tasks and the respective sections describing their outcomes:

- Understanding the constructs (cf. Section 6.3.1)
- Creating a model for the relations between these constructs (cf. Section 6.3.1)
- Developing a structured method for instantiating the model (cf. Section 6.3.2)
- Running an instantiation of the model in six case study (cf. Section 6.4.1)
- Performing a quantitative analysis of the model (cf. Section 6.4.2)

6.2.1 *Threats to validity*

Construct validity. If the constructs discussed during the case study are not interpreted in the same way by the researcher and the participants, there is a threat to construct validity. Thus, the approach and related concepts were explained through a workshop, and all steps were taken with our supervision. We also performed a literature review to understand the constructs and their relations properly. The MBSE experts, which participated in the task *Assess Benefits against candidate capabilities*, were directly involved in constructing the maturity model, which rules out stark misunderstandings of concepts. Additionally, their answers were aggregated using the median, thus enhancing this validity.

Researcher bias. Confirmation bias was mitigated through multiple sources of evidence, namely, feedback from the case study participants and quantitative analysis of the approach's execution. Additionally, we asked case study participants questions written with different wording to prove the same point.

Participant bias. Member checking was used to ask respondents to give systematic feedback on the analysis proposed by researchers when possible. It was systematic in the semi-structured interview process. We avoided implying a right answer through open-ended questions and asking people to rate their responses on a scale. The panel description (cf. Table 6.1) shows diversity of data sources. We asked the participants to write down their answers before saying them aloud to avoid a cognitive bias known as anchoring (Simmons,

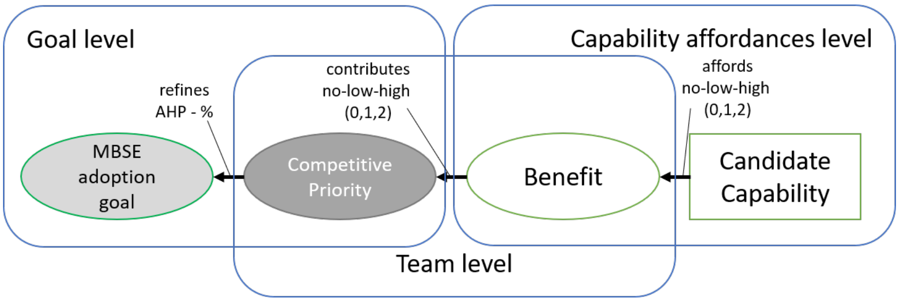


Figure 6.1: The goal-benefit meta-model elements and respective relations.

LeBoeuf, and Nelson, 2010) in which the first answer gravitates to the next ones.

6.3 THE GOAL-BENEFIT DECISION CALCULUS

Pöppelbuß and Röglinger (Pöppelbuß and Röglinger, 2011) proposed a set of design principles for developing maturity models. One of these was the inclusion of a decision calculus (Little, 1970) for prescriptive maturity models. A decision calculus is a quantitative decision support model that helps evaluate different alternatives concerning given objectives to derive an optimal recommendation (Bennett and Anderson, 1995; Peterson, 2017). In the context of the MBSE MM, this mechanism would help decision-makers to select the capabilities that are the most interesting to have their maturity further developed, according to the goal of the adoption strategy.

Therefore, we propose a goal-benefit decision calculus to characterize and break down relations between adoption goals and MBSE methods into a hierarchical structure of attributes. Such structure allows to systematically decompose and identify lower-level attributes (benefits capabilities affordances) which contribute to the achievement of higher-level attributes (competitive priorities), allowing a shift from rather abstract concepts like managerial considerations to more concrete attributes (Vetschera, 2006). The proposed decision calculus is composed of a goal-benefit model and an instantiation method.

In the following, we describe the model structure. Further, we propose a workflow to instantiate the model and generate recommendations.

6.3.1 *Model elements and relations*

The model is composed of four types of elements, namely *MBSE adoption goal*, *Competitive priorities*, *Benefits*, and *Candidate capabilities*. Their relations are depicted in Figure 6.1. The model forms a hierarchical tree structure of attributes, each level having specific considerations (cf. Figure 6.2). In the following, we describe the elements in detail and their relations.

6.3.1.1 *MBSE adoption goal*

The adoption goal is the reason why a team is undertaking the MBSE adoption project. Not being able to fulfill the goal is deemed a failure. The goal defines the "to-be" state after the method adoption benefits are harvested and are created based on the "as-is" state. Goals are defined on high levels of granularity using natural language; thus, they can be interpreted and fulfilled in many ways (*i.e.,* verification and validation challenge). This element is placed on the top of the tree structure (cf. Figure 6.2). Examples of adoption goals not limited to MBSE can be found in (Xu and Ramesh, 2007).

6.3.1.2 *Competitive priorities*

Competitive priorities are defined as the dimensions that a production system must possess to support the markets' demands in which an organization wishes to compete (Krajewski and Ritzman, 2018). Measures of competitive priorities fit in manufacturing strategy theory and are used to audit an organization's manufacturing strategy. Choices on which competitive priorities to emphasize should guide and constrain the design and operating decisions facing manufacturing executives (Ward et al., 2007). Four competitive priorities can be related to goals towards MBSE method adoption (Beck, 2000; Jitpaiboon, 2014):

- **Cost** is the invested value to deliver the product to the customer; thus, it influences profit. Lower costs are equivalent to higher efficiency. Are dimensions of Cost (Foo and Friedman, 1992; Hayes and Weelwright, 1984): manufacturing cost, value-added, selling price, running cost, service cost, and profit. In MBSE, cost reduction is achieved by optimizing the work (*e.g.,* traceability, automatization).

- **Quality** is described in terms of non-functional requirements. Regulations dictate acceptance levels of quality for potentially hazardous products/services. Quality can increase the costs (i.e., more reviews and scrutiny), but better methods and tools can increase quality without increasing running costs (through automated and dynamic analysis). More dimensions of this competitive priority can be found in (ISO/IEC, 2010).

- **Lead time** defines the time it takes to develop a product (i.e., how fast a new product can be delivered). Additionally, better-prepared competitors are likely to fulfill the needs of potential customers in advance. Refinements of Lead time are manufacturing lead time, due date performance, rate of product introduction, delivery lead time, frequency of delivery. Automation plays a significant role in MBSE methods in diminishing the Lead time.

- **Scope** is described in terms of the desirability of functional requirements. The product/service needs to provide what the user needs and what the customer wants, which should not be mistaken with project scope size. This dimension involves requirements elicitation, verification, and validation activities. Models are better ways to verify the feasibility of new functionalities.

Competitive priorities relate to the *MBSE adoption goal*, and *MBSE benefits*. This element populates the second level of the model tree (cf. Figure 6.2).

6.3.1.3 *Benefits*

A benefit is a value that is created as a result of the successful completion of a project, *e.g.*, method adoption (PMI, 2017). A benefit is a positive attribute that may support the adoption goal. For instance, *Easier handling of complexity* (Asan, Albrecht, and Bilgen, 2014) is a benefit generated by MBSE models and tools that support impact, coverage, and consistency analysis. The benefits associated with MBSE adoption used by this approach were collected from the literature and are described in more detail in Section 2.2.2. In our model, benefits relate to *Competitive priorities*, and *Candidate capabilities* (cf. Figure 6.1). In the following, we list these benefits and discuss some of their competitive priorities relations, which are not comprehensive and can change as new tools, methods, and systems are created.

- *Easier reuse of existing artifacts* (Salimi and Salimi, 2017)
  - **Cost:** reuse efforts are smaller than developing from scratch (Northrop, 2006).

  - **Quality**: reusing good quality artifacts improves the system's quality by diminishing the area of the code subject to failure.
  - **Lead Time:** reused parts are less prone to unplanned defect fixing and integration efforts are smaller than developing from scratch, enhancing chances of achieving deadlines.
  - **Scope:** is not directly related to this benefit because reuse can make product owners too clingy to their already existing "free" functionalities and poorly grasp what the real desire of the market is.
- *Better communication* (Hutchinson et al., 2011)
  - **Cost:** communication issues can generate rework.
  - **Quality**: communication issues can lead to defect injection (Dowson, 1997).
  - **Lead Time:** rework due to miscommunication impacts the time required to complete a system.
  - **Scope:** better requirements are likely to come up when engineers can communicate better.
- *Better understanding of problem domain* (Boehm, Gray, and Seewaldt, 1984)
  - **Cost:** poor understanding of the problem at hand usually leads to rework. Less iterations are required to reach a satisfactory level.
  - **Quality:** since the team will focus on relevant problems, more time is available to guarantee quality.
  - **Lead Time:** clear problem domains are less likely to bring new requirements during development, enhancing chances of achieving deadlines.
  - **Scope:** the better one understands the problem domain, easier it is to propose more aligned solutions.
- *Better understanding of solution space* (Harvey and Liddy, 2014)
  - **Cost:** new requirements are less likely to be discovered at late stages of development.
  - **Quality:** requirements are more likely to be well described, thus providing less room for mistaken design decisions.
  - **Lead Time:** good knowledge of what should be done makes it easier to reach agreed deadlines.
  - **Scope:** good understanding of the solution space increases chances of having good ideas for new functionalities.

- *Better estimates* (cost, impact of req. changes) (Dabkowski et al., 2013)
  - **Cost:** less risk to overrun budget.
  - **Quality:** when estimates are short, quality assurance activities might suffer to make the project fit agreed delivery dates .
  - **Lead Time:** well estimated targets are more likely to be reached.
  - **Scope:** receives little contribution from this benefit.
- *Less defects* (McConnell, 2004)
  - **Cost:** less effort fixing bugs, less defects found at later stage of development process (Afzal et al., 2013), less defect found by system users.
  - **Quality:** better system quality.
  - **Lead Time:** less time needed to fix bugs, thus more likely to reach targets.
  - **Scope:** less needed time for defect fixing is equal to more project time available to develop better functionalities.
- *Easier handling of complexity* (Asan, Albrecht, and Bilgen, 2014)
  - **Cost:** less project navigation and tracing efforts.
  - **Quality:** less defects due to missing information.
  - **Lead Time:** less time consumed tackling complexity.
  - **Scope:** more project time available to develop better functionalities.
- *Improved verification & validation* (McConnell, 2004)
  - **Cost:** less effort required to perform these tasks.
  - **Quality:** less risk of missing requirements implementation or misunderstandings.
  - **Lead Time:** evidence of the requirements implementation can be easily collected.
  - **Scope:** better ways to describe requirements fosters new interesting functionalities.
- *Improved quality of specification* (Fosse and Bayer, 2016)
  - **Cost:** rework due to poor specification is mitigated.
  - **Quality:** better specification requires less ad hoc decision-making for filling gaps, thus fewer opportunities to inject defects.

  - **Lead Time:** less effort is required to make sense of good specification.
  - **Scope:** better specification encompasses more information which is the input for creating better functionalities.
- *Efficient certification* (Helle, 2012)
  - **Cost:** reuse of previous certification artifacts and automation has implications on cost
  - **Quality:** tools to generate evidence also uncover quality issues.
  - **Lead Time:** certification takes time but evidence can be generated at early stages of development (through simulation or automated checks).
  - **Scope:** more project time available to develop better functionalities.

6.3.1.4 *Candidate capabilities*

Candidate capabilities are the units of MBSE methods and the solution space of the adoption initiative (*i.e.,* MBSE methods not employed by the team). They generate benefits that can (or not) help the team to reach its method adoption goal. The capabilities used in this approach are from the MBSE MM (cf. Section 2.4).

6.3.1.5 *Hierarchical levels*

The elements and their relationships form a three-level hierarchical tree (cf. Figure 6.2). Each level is documented in an artifact which is the output of a task of the instantiation method (cf. Section 6.3.2). The relations between elements are defined using either Analytic hierarchy process (AHP) (Saaty, 1980) or the *no-low-high* scale, which are described in the following:

- **Analytic Hierarchy Process** (Saaty, 1980) is a method to support multiple-criteria decision analysis. It works by building a hierarchical model of properties to reason a decision. The main decision is placed at the graph's root, while possible solutions are placed at the leaves. Connecting them are the properties of the solutions. Then, the properties' importance is compared in a pairwise fashion (*i. e.,* is property A more important than property B) and to which magnitude, expressed in an ordinal scale from 1 to 9. The comparisons are translated to weights that are assigned to the properties through the connecting edges. The parent importance is propagated to the children nodes pro-

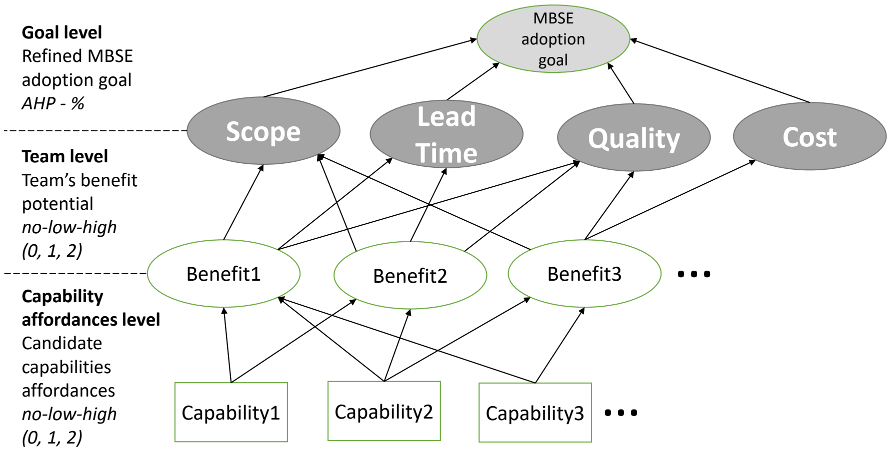


Figure 6.2: The hierarchical structure of attributes of the goal-benefit metamodel.

portionally to the assigned weights until all edges have been considered when reaching the leaves (i.e., solution). In the end, the leaf's values should sum up to the root's importance.

- **No-low-high scale** is a three-level scale for defining relationships, and it describes if there is no relation, if the relationship strength is low, or if it is high. This scale has little room for mistakes and provides the granularity required for this model. The scale can be translated to numbers for quantitative use, assigning zero for no relation, one for low, and two for high. This scale is used to define the *Team's context* and the *Capability affordances*. Other established frameworks use a similar scale (Horkoff et al., 2014), such as in the Business Intelligence Model, which uses a two-grade scale to describe influence relationships among goals (*i. e.*, weak positive and strong positive).

The hierarchical levels of the model (cf. Figure 6.2) are described in the following:

GOAL LEVEL. This level represents the refinement of the goal in competitive priorities. By refinement, we mean: when trying to achieve the goal, how each competitive priority should be considered. Every goal delivers a specific set of relations between those entities. The Goal level encompasses the MBSE adoption goal, the four competitive priorities, and the valuation of the relation between both. This level is developed during the task *Define MBSE adoption goal*, and *Refine MBSE adoption goal*. There are two artifacts related to this level, namely the *MBSE adoption goal*, and the *Refined MBSE adoption goal* (cf. Section 6.3.2). The relations value is defined in percentages, and there are four relationships of this kind in the model, one for each competitive priority.

TEAM LEVEL. The relations between competitive priorities and benefits represent how the latter contribute to the former based on the team's contextual and intrinsic characteristics. The model considers these factors in the assessment of the *Team context* relations (*i.e.,* a cost-efficient team will most likely benefit less from cost-related benefits). This relation is described using the *no-low-high* scale. There are forty relationships of this kind in the model (*i.e.,* four competitive priorities times ten benefits). These values are defined in the task *Characterize team context* and belong to the *Team's context* artifact (cf. Section 6.3.2).

CAPABILITY AFFORDANCES LEVEL. The relations of this level exists between benefits and capabilities and represent the benefits generated by adopting the respective capability. The value of the relations is context-independent; thus, reuse is possible, but the candidate capabilities selected are team-specific (*i. e.,* must be composed of non-implemented capabilities). It is defined using the same scale as the *Team level* relations (*i.e., no-low-high* scale). The number of these relations in the model is 10*(the number of candidate capabilities). This level is defined during the task *Assess candidate capabilities affordances* and the values are store in the *Candidate capabilities affordances*, and *Capabilities affordances repository* artifacts (cf. Section 6.3.2).

6.3.2 *The instantiation method*

In this subsection, we describe a method to guide the instantiating of the model. When the model is instantiated, values are assigned to the relations quantifying those regarding refinement/contribution/-generation. The method's outcome is a list of candidate capabilities prioritized according to their contribution towards the adoption goal. In the following, we describe the actors involved and the workflow tasks (cf. Figure 6.3). Later, we describe each task in detail. A running example of the method is provided in case study 1 (cf. Section 6.4.1.2).

6.3.2.1 *Actors*

Three groups of actors participate in the method execution:

- **Team members.** are the actors belonging to the team adopting the MBSE methods. Participants of this group are system engineers, developers, team leaders, and managers. Having more than one participant from this group when performing the tasks enhances the approach's results, and their answers should be discussed to reach a consensus.

- **Researchers.** encompasses the individuals applying the approach. This group participates in all tasks, either actively or supporting the other actors.

- **MBSE Experts.** are professionals with deep knowledge of MBSE methods, their application in teams, and the benefits these methods can deliver. They are consultants, researchers, instructors, among others.

6.3.2.2 *Tasks*

In this subsection, we describe the tasks required to instantiate the meta-model. Dependencies between tasks can be seen in Figure 6.3.

ASSESS TEAM PROCESS MATURITY. In this task, the process maturity of the team is assessed using the MBSE MM. Researchers ask team members through interviews whether they produce or manipulate specific artifacts described by the MBSE MM capabilities. If they answer yes, the capability is considered implemented. In case of doubt or lacking understanding, the researchers should ask the team members for artifacts related to the capabilities (*e. g.,* documents or models with the required information is documented). The outcome of this task is the current maturity profile of the team processes. Team members fit for this task are engineers performing the processes.

Output: *Maturity profile* – This artifact describes all the implemented capabilities in a team.

SELECT CANDIDATE CAPABILITIES. The candidate capabilities are selected from the MBSE MM based on the *Maturtity profile*. Any non-implemented capability can be a candidate. The number of candidate capabilities to be selected is not fixed and depends on the allocated effort for MBSE adoption. We suggest selecting at least two candidate capabilities per focus area for each adoption iteration (*i.e.,* a team having *Operational context* focus area until OPC C implemented would select as candidate capabilities OPC D and OPC E). Other capabili-

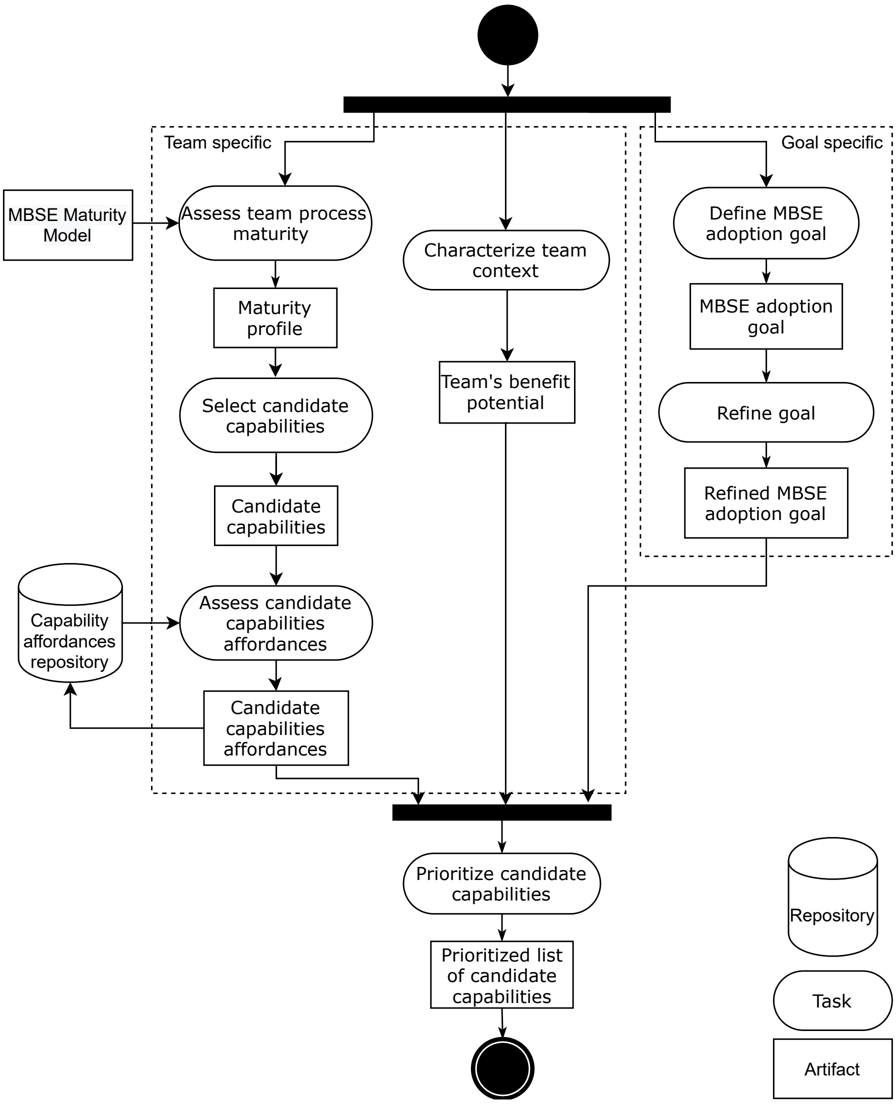


Figure 6.3: Workflow of the instantiation method.

ties can be added to the pool in a new run of the approach in the subsequent iterations.

Input: *Maturity profile*.

Output: *Candidate capabilites* – This artifact is a subset of unimplemented capabilities in the team.

ASSESS CANDIDATE CAPABILITIES AFFORDANCES. At this task, MBSE experts assess the magnitude of the benefits each candidate capabilities can generate when implemented in a team. For each pair (candidate capability, benefit), experts characterize the generation of benefits using the *no-low-high* scale (cf. Section 6.3.1.5). Moreover, their answers are aggregated using the median. The benefits generated by the candidate capabilities are not team-specific; thus, this task can be performed without considering the team context. Therefore, the outcome of this task can be reused in multiple case studies.

Input: *Candidate capabilities*.

Output: *Candidate capabilities affordances*.

DEFINE MBSE ADOPTION GOAL. The team members should describe, using natural language, the team's main goal for adopting MBSE methods. If many goals are elicited, the most important goal should be selected (there are many techniques to prioritize goals, which is out of this approach's scope). Participants with managerial roles are the best fit for this task.

Output: *MBSE adoption goal*.

REFINE GOAL. This task outcome is the refinement of the adoption goal into competitive priorities (cf. Section 6.3.1.2). The goal described in natural language can be interpreted in many ways; thus, a crispier understanding is *sine qua non*. Hereby Analytical Hierarchy Process (AHP) (Saaty, 1980) is used, a method designed to compare the importance of properties in a pairwise fashion. Team members compare competitive priorities with each other regarding their importance towards achieving the goal. After the calculation, the relevance of each competitive priority is delivered in percentages. Team members having managerial roles should perform this task, and their answers must achieve a consensus. The researchers assist this task.

Input: *MBSE adoption goal*.

Output: *Refined MBSE adoption goal*.

CHARACTERIZE TEAM CONTEXT. In this task, team members should assess how benefits contribute to the competitive priorities according to the team context using the *no-low-high* scale (cf. Section 6.3.1.5). Benefits can have little influence if the team is already strong in specific dimensions, or the opposite can happen, and an addressed benefit can effectively diminish a performance gap. Thus, the outcome of this task is team-specific.

Output: *Team's benefit potential* – This artifact represents the improvement potential of each MBSE benefit towards the competitive priorities within a team.

PRIORITIZE CANDIDATE CAPABILITIES. The last task compares each candidate capability relevance towards the goal by prioritizing them according to points distributed based on the instantiated weighted relations. Starting at the *MBSE adoption goal* (cf. Figure 6.2), the *Refined MBSE adoption goal* artifact defines how many *goal-points* each competitive priority will be assigned, which are calculated by multiplying 1000 points to the relations percentages. Next, these points are distributed to the benefits according to the weights defined in the *Team's context* artifact. Each benefit receives points from each competitive priority according to the ***Benefit-points calculation formula***:

(relation's weight)*(goal-points value)/(sum of all weights of respective competitive priority).

Then, the values are summed up to form the total *benefit-points* for each benefit. Further, we must distribute these *benefit-points* to the capabilities. The same process is performed at *Capability affordances level* using the information from the *Candidate capability affordances* artifact and the ***Candidate-points calculation formula***:

(relation's weight)*(benefit-points value)/(sum of all weights of respective benefit).

This calculation is performed for each pair (benefit, candidate capability) and is summed in the candidate capability. The final step is ordering the candidate capabilities according to the sum of the total points they received. The candidate capabilities with the most points are more relevant towards meeting the adoption goal considering the team's context. Figure 6.4 depicts this process.

Input: *Candidate capabilities affordances, Team's benefit potential, Refined MBSE adoption goal.*

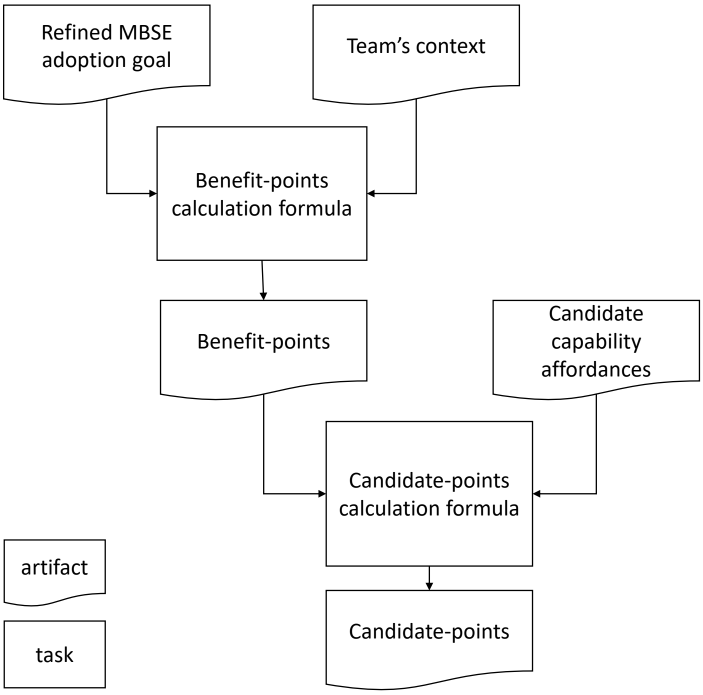


Figure 6.4: Workflow of the task *Prioritize candidate capabilities*.

Output: *Prioritized list of candidate capabilities* – The list is prioritized according to the importance of fulfilling the adoption goal.

The tasks *Define MBSE adoption goal* and *Refine goal* are goal-specific. *Characterize team context*, *Assess team process maturity*, and *Select candidate capabilities* are team specific. The outcome of *Assess candidate capabilities affordances* can be reused with different teams.

6.3.2.3 *Other elements of the workflow*

Two elements represented in the workflow provide support to the development of the activities. The MBSE MM is required for running the approach and is described in detail in Section 2.4. The *Capability affordance repository* is the repository where the output of the task *Assess candidate capabilities affordances* is stored for future use. The *Candidate capability affordances* artifact is not context-specific, meaning that another run of the approach (with the same team or with another team) can *re-use* the assessment of the same previously assessed candidate capability. This work collected 53 capability affordances that can be reused and enhanced in the approach further execution (cf. Table 6.3).

Table 6.1: Case study participants - demographic data.

ID	Industry	Team size	Role of participants
CS1	Automotive	5	Process manager, architect
CS2	Government	8	Process analyst, project manager
CS3	Education	9	Project manager, architect, developer
CS4	Defense	6	Project and product manager
CS5	Defense	4	Systems manager
CS6	Defense	12	Project manager

6.4 VALIDATION

In this section, we aim to prove that the approach fits its purpose. Thus it needs to adhere to the requirements defined in Section 6.2. This section is divided in two subsections, namely *Qualitative evaluation* and *Quantitative evaluation*. In the former, we report applying the approach in real development teams in six case studies where we collected the participants' impression regarding the approach itself and delivered results. Our intention was to verify requirements **R1** and **R3**. In the latter, we provide an analysis of the model based on metrics regarding its use. The intention was to verify requirement **R2**.

6.4.1 *Qualitative evaluation*

In this section, we provide evidence that the approach fulfills the requirements ***R1****: The approach must be cost-effective* and ***R3****: The approach must improve decision-making*. We seek to achieve this through six case studies in which we applied the approach in real-life development teams and collected their feedback about the delivered results and the experience of applying the approach. The case studies were carried out in teams located in Brazil and Germany. A demographic summary of the teams that participated in the case studies can be seen in Table 6.1.

The case study participants were debriefed over the model and method in a pitch meeting to present the approach. For the task *Refine goal*, the participants performed the AHP method using an online tool[2] to generate the values. We put together this task's results for each goal in Table 6.6 for comparing how each team refined their goals towards competitive priorities. The *Team's benefit potential* for each case study can be seen in Appendix E.

Once all tasks were performed (cf. Section 6.2) we held a meeting to present the results and collect the impressions of the participants towards the approach. These findings and how they corroborate to the fulfillment of the aforementioned design requirements are described at the end of each case study in a paragraph named *Reflections on the model*.

We provide a running example in case study 1 described in more detail than the other case studies (cf. Section 6.4.1.2). In the following subsections, we describe the task *Assess candidate capabilities affordances*, and how the case studies were performed.

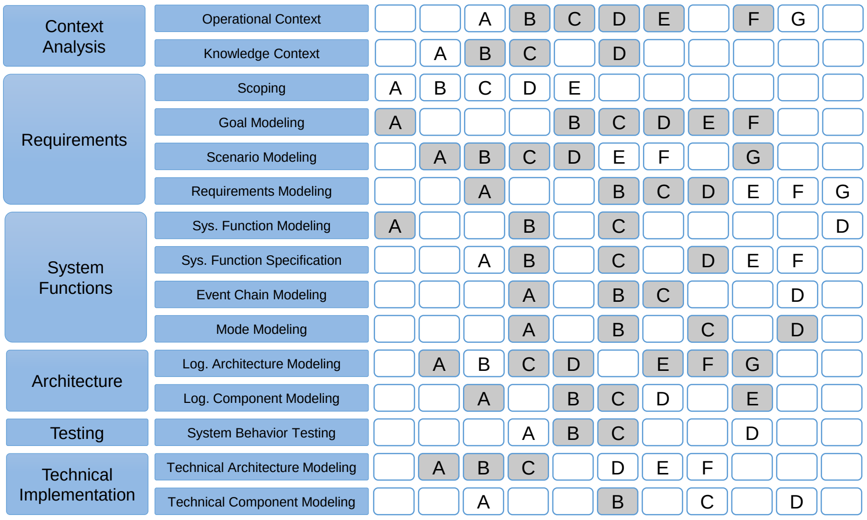
6.4.1.1 *Expert assessment*

For the case studies, we selected 53 candidate capabilities (cf. Figure 6.5) to be evaluated by MBSE experts at the task *Assess candidate capabilities affordances*. We canvass the opinion of five MBSE experts in a single-round survey (Pan et al., 1996). They were sent through e-mail a spreadsheet file containing a capability for each row and benefits for columns which they should grade their relation using the *no-low-high* scale (cf. Section 6.3.1.5). We collected their answers and aggregated the values using the median. The experts selected for this assessment were directly involved in developing the MBSE MM and had a deep knowledge of the capabilities. However, any MBSE expert with solid methods knowledge and experience would be fit for the assessment and eventual discrepancies would be ironed out through the responses median.

The sum of points assigned to each of the 53 capabilities can be seen in Table 6.3. The distribution of the values per capability followed a normal distribution (*i.e.,* bell-shaped curve) as shown in Figure 6.6. Therefore, the MBSE experts were neither optimistic nor pessimistic about the capabilities affordances. The criteria for selecting these candidate capabilities were based on the case studies teams' current maturity profile and the next two unimplemented respective capabilities of each focus area. The candidate capabilities for each case study can be seen in Figures 6.8-(b), 6.9-(b), 6.10-(b), 6.11-(b), 6.12-(b), and 6.13-(b).

---

[2] `https://bpmsg.com/academic/ahp_calc.php`



Context Analysis	Operational Context			A	B	C	D	E		F	G	
	Knowledge Context		A	B	C		D					
Requirements	Scoping	A	B	C	D	E						
	Goal Modeling	A				B	C	D	E	F		
	Scenario Modeling		A	B	C	D	E	F		G		
	Requirements Modeling			A			B	C	D	E	F	G
System Functions	Sys. Function Modeling	A			B		C					D
	Sys. Function Specification			A	B		C		D	E	F	
	Event Chain Modeling				A		B	C			D	
	Mode Modeling				A		B		C		D	
Architecture	Log. Architecture Modeling		A	B	C	D		E	F	G		
	Log. Component Modeling			A		B	C	D		E		
Testing	System Behavior Testing				A	B	C			D		
Technical Implementation	Technical Architecture Modeling		A	B	C		D	E	F			
	Technical Component Modeling			A			B		C		D	

Figure 6.5: Capabilities assessed by MBSE experts on the task *Assess candidate capabilities affordances*.

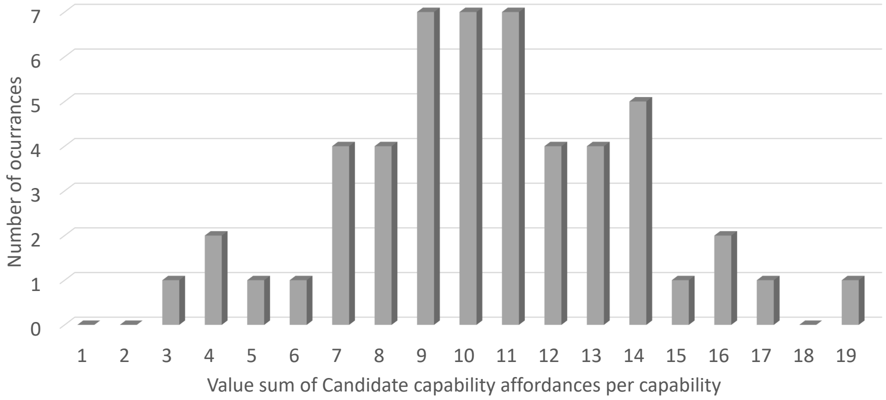


Figure 6.6: Distribution of weight-sum points amount per candidate capability assigned by MBSE experts.

Table 6.3: Weight-points sum of *Candidate capability affordances* artifact from the task *Assess candidate capabilities affordances*.

Cap.	**weight-sum**	**Cap.**	**weight-sum**	**Cap.**	**weight-sum**
LAM G	19	LAM E	11	SCM C	9
SBT C	17	RM B	11	TAM C	9
RM C	16	OPC D	11	MM C	9
LCM E	16	OPC E	11	GOM D	8
OPC F	15	SFS B	11	LCM A	8
TAM C	14	SCM D	11	LCM B	8
RM D	14	LCM C	11	SCM G	8
LAM D	14	SCM B	10	GOM F	8
SFS C	14	ECM C	10	GOM E	7
SFS D	14	RM A	10	TAM A	7
SFM C	13	SFM B	10	MM A	7
SCM A	13	ECM B	10	KNC C	7
SBT B	13	TAM B	10	SFS A	6
MM D	13	TCM B	10	GOM B	5
OPC C	12	ECM A	9	GOM A	4
OPC B	12	LAM A	9	KNC D	4
LAM C	12	GOM C	9	KNC B	3
LAM F	12	MM B	9	-	-

6.4.1.2 *Case study 1 (CS1)*

This case study is described with greater detail to serve as a running example. In this case study, the approach was applied to a system development team of a Tier-1 manufacturer of rolling element bearings for automotive, aerospace, and industrial uses. The case study started with the task *Assess team process maturity* which was performed through face-to-face interviews with four systems engineers. The outcome of this task is the team's *Maturity profile* (cf. Figure 6.7). During the task *Select candidate capabilities* 23 candidates capabilities were considered by the researchers for this case study (cf. Figure 6.8-(b)).

The remaining tasks were performed through two teleconferences, lasting one hour each, with two participants; a process management specialist and an architect. We started with the activity *Define adoption goal*. The participants explained that the team needed to improve interdisciplinary work between system development and software development. After some brainstorming rounds, the goal was finally defined as:

"Enhance communication between software and system development teams."

The *MBSE adoption goal* was used as input for the next task, *Refine goal*. The pairwise comparison of each competitive priority can be seen in Table 6.4, and the calculated result in percentages in Table 6.6. At first, the participants were surprised that *Scope* (53.6%) was ranked higher than *Quality* (29.8%), but when revisiting the evaluation, they stated that *Scope* was more important than *Quality* for a scale of 3 (cf. Table 6.4). Then they realized their assumption was based on the team's priorities in which *Quality* is more important than *Scope*, but considering the adoption goal, *Scope* was more important. Nevertheless, *Quality* was ranked second most important competitive priority. Next, we performed the activity *Characterize team context*. The result of this activity can be seen in Table 6.5.

Finally, the task *Prioritize candidate capabilities* was performed. The first five items from the *Prioritized list of candidate capabilities* belong to four focus areas; the first capability from *Requirements Modeling*, two capabilities from *Operational Context*, and one capability from *System Function Modeling* and *Systems Function Specification*. The last two capabilities had requirements not yet present in the team; thus, the required capabilities must be implemented before they are considered. The five first prioritized capabilities can be seen in Figure 6.8-(c).

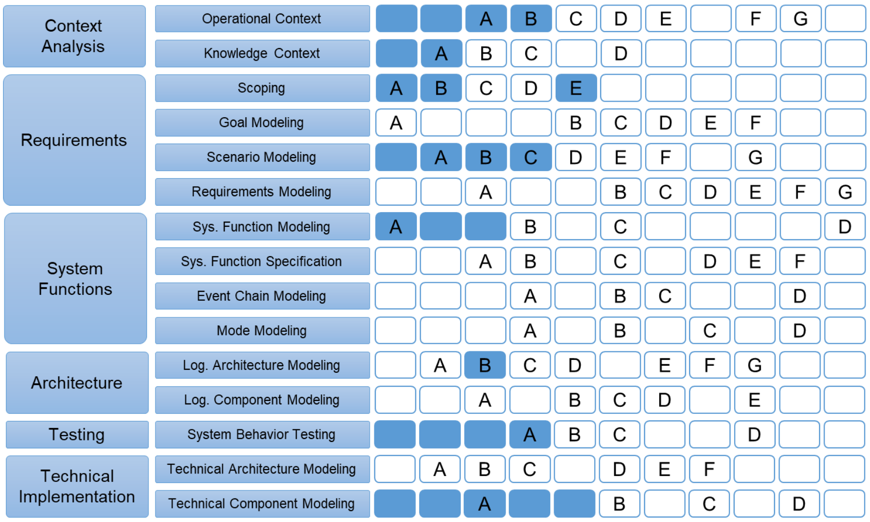
REFLECTION ON THE MODEL: The participants stated that the approach *"helps assess the team priorities in the first place"* and *"proposes how*

Table 6.4: AHP pairwise comparison from task *Refine goal* from case study 1.

Pairwise comparison	**Answer**	**How much more?**
Cost or Quality	Quality	5
Cost or Lead Time	Lead Time	5
Cost or Scope	Scope	7
Quality or Lead Time	Quality	5
Quality or Scope	Scope	3
Lead Time or Scope	Scope	5

Table 6.5: Instance of *Team's benefit potential*, output of the task *Characterize team context* from case study 1.

MBSE benefit	**Cost**	**Quality**	**Lead Time**	**Scope**
Easier reuse of existing artifacts	2	2	1	0
Better communication	0	2	1	2
Better understanding of problem domain	1	1	1	1
Better understanding of solution space	0	1	2	2
Better estimates	2	1	1	0
Less defects	2	2	1	0
Easier handling of complexity	1	2	1	1
Improved verification & validation	2	2	1	0
Improved quality of specification	1	1	1	1
Efficient certification	1	2	1	1
Sum	14	16	11	8



Context Analysis	Operational Context	▨	▨	▨ A	▨ B	C	D	E		F	G	
	Knowledge Context	▨	▨ A	B	C		D					
Requirements	Scoping	▨ A	▨ B	C	D	▨ E						
	Goal Modeling	A				B	C	D	E	F		
	Scenario Modeling	▨	▨ A	▨ B	▨ C	D	E	F		G		
	Requirements Modeling			A			B	C	D	E	F	G
System Functions	Sys. Function Modeling	▨ A	▨	▨	B		C					D
	Sys. Function Specification			A	B		C		D	E	F	
	Event Chain Modeling				A		B	C			D	
	Mode Modeling				A		B		C		D	
Architecture	Log. Architecture Modeling		A	▨ B	C	D		E	F	G		
	Log. Component Modeling			A		B	C	D		E		
Testing	System Behavior Testing	▨	▨	▨	▨ A	B	C			D		
Technical Implementation	Technical Architecture Modeling		A	B	C		D	E	F			
	Technical Component Modeling	▨	▨	▨ A	▨	▨	B		C		D	

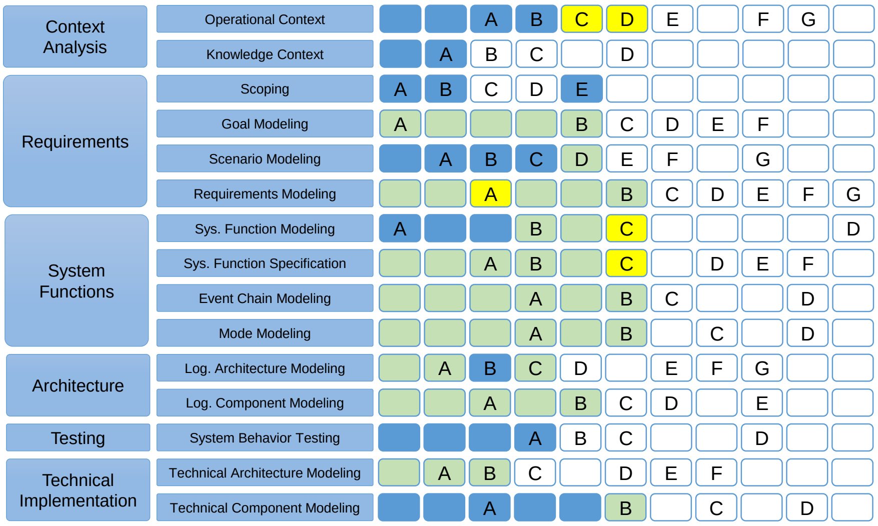
Figure 6.7: Case study 1 - Maturity Profile.

the team could evolve its processes". They also said that *"the results were a good starting point when deciding how to execute MBSE process maturity improvement"*. These statements are evidence that the approach fulfills requirement **R3**. The participants rated the approach cost-efficient, thus fulfilling requirement **R1**.

6.4.1.3 *Case study 2 (CS2)*

Case study 2 team develops educational management information systems for a university with functionalities such as course registering and grade management. The complete method was performed in one face-to-face meeting, which lasted around two and a half hours. Three participants attended the meeting, namely a project manager, an architect, and a developer. After performing the task *Assess team process maturity* and having in hands the maturity profile of the team (cf. Figure 6.9-(a)) we selected 24 candidate capabilities (cf. Figure 6.9-(b)). The team members were concerned with not having enough documentation available to understand the systems once someone left the team, *i. e.*, loss of knowledge due to employee turnover. Additionally, much effort was required to transfer knowledge to new team members. The following goal was defined based on the issue mentioned above:

"Improve knowledge transfer of the systems to new team members."



Context Analysis	Operational Context	🟦	🟦	🟦 A	🟦 B	🟨 C	🟨 D	E		F	G	
	Knowledge Context	🟦	🟦 A	B	C		D					
Requirements	Scoping	🟦 A	🟦 B	C	D	🟦 E						
	Goal Modeling	🟩 A	🟩	🟩	🟩	🟩 B	C	D	E	F		
	Scenario Modeling	🟦	🟦 A	🟦 B	🟦 C	🟩 D	E	F		G		
	Requirements Modeling	🟩	🟩	🟨 A	🟩	🟩	🟩 B	C	D	E	F	G
System Functions	Sys. Function Modeling	🟦 A	🟦	🟦	🟩 B	🟩	🟨 C					D
	Sys. Function Specification	🟩	🟩	🟩 A	🟩 B	🟩	🟨 C		D	E	F	
	Event Chain Modeling	🟩	🟩	🟩	🟩 A	🟩	🟩 B	C			D	
	Mode Modeling	🟩	🟩	🟩	🟩 A	🟩	🟩 B		C		D	
Architecture	Log. Architecture Modeling	🟩	🟩 A	🟦 B	🟩 C	D		E	F	G		
	Log. Component Modeling	🟩	🟩	🟩 A	🟩	🟩 B	C	D		E		
Testing	System Behavior Testing	🟦	🟦	🟦	🟦 A	B	C			D		
Technical Implementation	Technical Architecture Modeling	🟩	🟩 A	🟩 B	C		D	E	F			
	Technical Component Modeling	🟦	🟦	🟦 A	🟦	🟦	🟩 B		C		D	

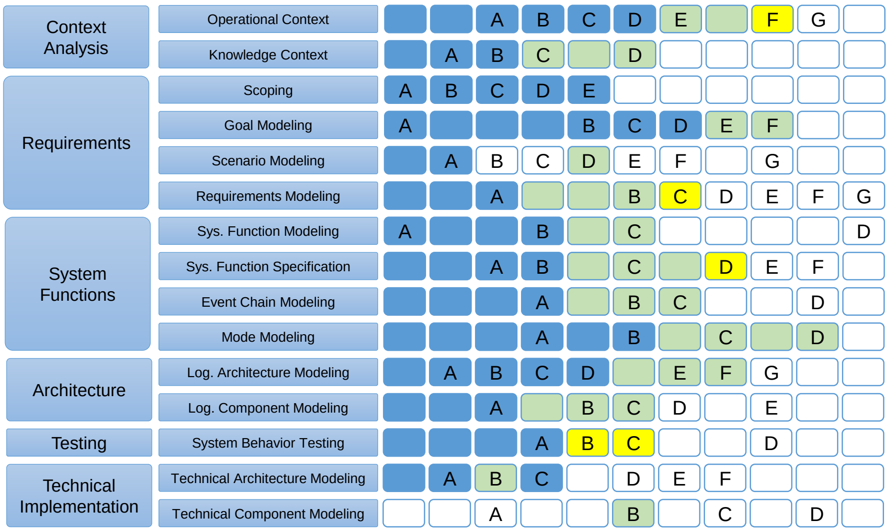
Figure 6.8: Case study 1 - (a) blue capabilities: current team MBSE maturity profile, (b) green and yellow capabilities: candidate capabilities for adoption, (c) yellow capabilities: five first prioritized capabilities.

The *Refined MBSE adoption goal* artifact had *Quality* as the leading competitive priority with 45% followed by *Lead Time* with 35.3% (cf. Table 6.6). The *Prioritized list of candidate capabilities* had two capabilities from *System Behavior Testing* focus area among their five best scoring. This is reasonable since *Quality* was rated the most important competitive priority towards goal achievement. Other relevant focus area were *Systems Function Specification*, *Requirements Modeling* and *Operational Context* (cf. Figure 6.9-(c)). From all these, only the *Systems function specification* capability requires a previous unimplemented capability (SFS C).

REFLECTION ON THE MODEL: The participants found the approach quite helpful since it *"led them to reflect on the team's current situation within the questions raised"*. The results were considered relevant because they *"were aligned with the problems identified by the team beforehand"*. They said the approach provided them with a direction to tackle those issues (compliance with **R3**). The approach consumed a reasonable amount of time in the participants' eyes, thus complying with **R1**.

6.4.1.4 *Case study 3 (CS3)*

The team from case study 3 develops information systems for a government agency. Their duties include the creation of systems to sup-



Context Analysis	Operational Context	🟦	🟦	🟦 A	🟦 B	🟦 C	🟦 D	🟩 E	🟩	🟨 F	G	
	Knowledge Context	🟦	🟦 A	🟦 B	🟩 C	🟩	🟩 D					
Requirements	Scoping	🟦 A	🟦 B	🟦 C	🟦 D	🟦 E						
	Goal Modeling	🟦 A	🟦	🟦	🟦	🟦 B	🟦 C	🟦 D	🟩 E	🟩 F		
	Scenario Modeling	🟦	🟦 A	B	C	🟩 D	E	F		G		
	Requirements Modeling	🟦	🟦	🟦 A	🟩	🟩	🟩 B	🟨 C	D	E	F	G
System Functions	Sys. Function Modeling	🟦 A	🟦	🟦	🟦 B	🟩	🟩 C					D
	Sys. Function Specification	🟦	🟦	🟦 A	🟦 B	🟩	🟩 C	🟩	🟨 D	E	F	
	Event Chain Modeling	🟦	🟦	🟦	🟦 A	🟩	🟩 B	🟩 C			D	
	Mode Modeling	🟦	🟦	🟦	🟦 A	🟦	🟦 B	🟩	🟩 C	🟩	🟩 D	
Architecture	Log. Architecture Modeling	🟦	🟦 A	🟦 B	🟦 C	🟦 D	🟩	🟩 E	🟩 F	G		
	Log. Component Modeling	🟦	🟦	🟦 A	🟩	🟩 B	🟩 C	D		E		
Testing	System Behavior Testing	🟦	🟦	🟦	🟦 A	🟨 B	🟨 C			D		
Technical Implementation	Technical Architecture Modeling	🟦	🟦 A	🟩 B	🟦 C		D	E	F			
	Technical Component Modeling			A			🟩 B		C		D	

Figure 6.9: Case study 2 - (a) blue capabilities: current team MBSE maturity profile, (b) green and yellow capabilities: candidate capabilities for adoption, (c) yellow capabilities: five first prioritized capabilities.

port regulations and decrees. At the interview time, eight engineers worked on this team. We performed the tasks in two face-to-face meetings with a duration of one hour each. In the first meeting, we performed the task *Assess team process maturity* with two team members, namely a project manager and a requirements engineer. In the second meeting, we executed the rest of the tasks with the same project manager from the first meeting and a process analyst. Before the second meeting, we asked the participants to think about adoption goals and provided them with a spreadsheet to describe the relations between competitive priorities and benefits. We asked them to define those relations in the same way as the task *Characterize team context*, so they could reflect on their own, thus avoiding the “anchoring effect” (Simmons, LeBoeuf, and Nelson, 2010), and diminish the task duration (results can be seen in Appendix E). Twenty-one candidate capabilities were considered for this case study.

When asked to produce method adoption goals, the participants stated they had rework problems due to miscommunication and poorly described requirements. They also had effort estimation issues. In a dead-lock to decide between both issues we proposed to work with two goals (cf. Table 6.6):

CS3_G1: “Reduce rework due to communication issues and scope definition”

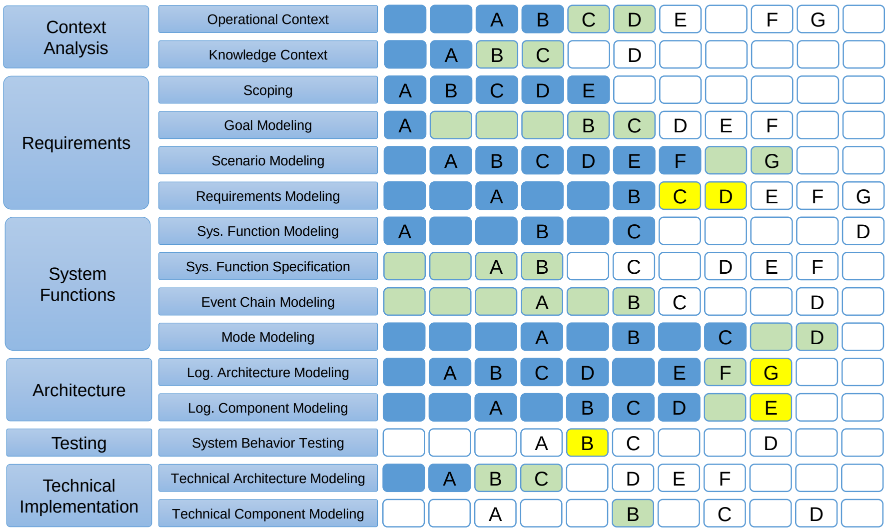
CS3_G2: "Improve effort estimation".

We performed the task *Refine adoption goal* twice, one time for each goal. Both goals delivered slightly similar results (cf. Table 6.6), being *Lead time* the most important competitive priority (CS3_G1: 50.3% and CS3-G2: 68%). For CS3_G1, *Scope* was the second most important while *Quality* was the third, whilst for CS3_G2 the opposite happened. Moreover, *Quality* had a similar percentage in both goals (CS3_G1: 13.7% and CS3_G2: 14.9%). After performing the task *Prioritize candidate capabilities* for each goal, we found out that the first five most important capabilities were the same for both goals (cf. Figure 6.10-(c)), although they featured in different order. This outcome was expected due to both goals' *Refined MBSE goal adoption* similarities. The results suggested adopting 2 capabilities from *Requirements Modeling* and one capability from *Logical Component Modeling*, *Logical Architecture Modeling*, and *System Behavior Testing*. However, the last two capabilities have requirements not present in the current team's *Maturity profile*, namely LAM G and SBT B.

REFLECTION ON THE MODEL: The participants agreed with the approach results, especially regarding system testing capabilities. They stated that *"the results help decide the next improvement steps"* and recommended the approach use in their team. They said *"the approach identified many improvement points that were under the radar"* (thus compliance with **R3**). The approach application using two goals was an exciting experience. Compared to the other "one-goal" case studies, the extra effort was not significant; only the tasks *Define MBSE adoption goal, Refine goal, and Prioritize candidate capabilities* needed to be executed twice. The participants disclosed that the time required to perform the approach was acceptable (compliance with **R1**).

6.4.1.5 *Case study 4 (CS4)*

The team of the case study four developed systems for military operations. It was composed of six team members and applied Scrum development methodology. We performed all workflow steps (cf. Section 6.3.2) in a single face-to-face meeting lasting one and a half-hour. A single team member with the roles of project and product manager participated in the case study. Since the team worked under an agile methodology, few artifacts besides code were produced used. Thus, very few capabilities were marked as being present (cf. Figure 6.11-(a)) since MBSE MM is artifact-oriented (cf. Section 2.4). After performing



Context Analysis	Operational Context	🟦	🟦	🟦 A	🟦 B	🟩 C	🟩 D	E		F	G	
	Knowledge Context	🟦	🟦 A	🟩 B	🟩 C		D					
Requirements	Scoping	🟦 A	🟦 B	🟦 C	🟦 D	🟦 E						
	Goal Modeling	🟦 A	🟩	🟩	🟩	🟩 B	🟩 C	D	E	F		
	Scenario Modeling	🟦	🟦 A	🟦 B	🟦 C	🟦 D	🟦 E	🟦 F	🟩	🟩 G		
	Requirements Modeling	🟦	🟦	🟦 A	🟦	🟦	🟦 B	🟨 C	🟨 D	E	F	G
System Functions	Sys. Function Modeling	🟦 A	🟦	🟦	🟦 B	🟦	🟦 C					D
	Sys. Function Specification	🟩	🟩	🟩 A	🟩 B		C		D	E	F	
	Event Chain Modeling	🟩	🟩	🟩	🟩 A	🟩	🟩 B	C			D	
	Mode Modeling	🟦	🟦	🟦	🟦 A	🟦	🟦 B	🟦	🟦 C	🟩	🟩 D	
Architecture	Log. Architecture Modeling	🟦	🟦 A	🟦 B	🟦 C	🟦 D	🟦	🟦 E	🟩 F	🟨 G		
	Log. Component Modeling	🟦	🟦	🟦 A	🟦	🟦 B	🟦 C	🟦 D	🟩	🟨 E		
Testing	System Behavior Testing				A	🟨 B	C			D		
Technical Implementation	Technical Architecture Modeling	🟦	🟦 A	🟩 B	🟩 C		D	E	F			
	Technical Component Modeling			A			🟩 B		C		D	

Figure 6.10: Case study 3 - (a) blue capabilities: current team MBSE maturity profile, (b) green and yellow capabilities: candidate capabilities for adoption, (c) yellow capabilities: five first prioritized capabilities for both goals.

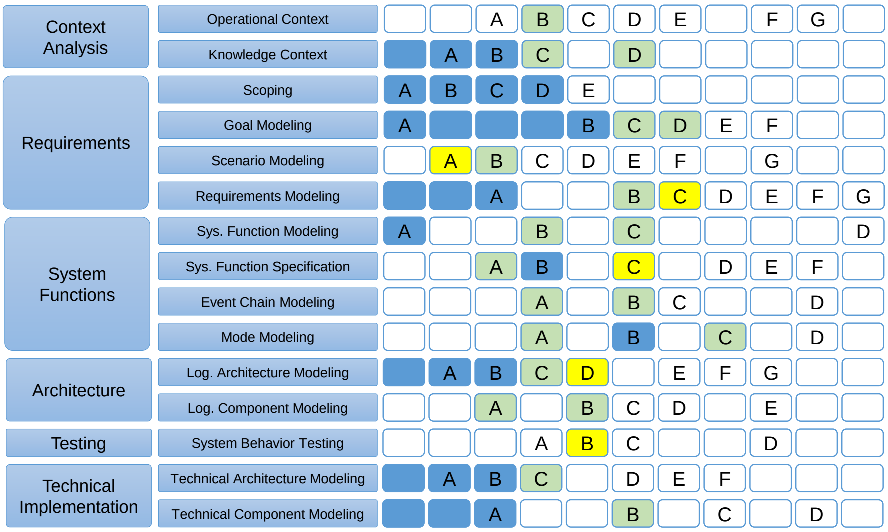
the maturity assessment, 24 candidate capabilities were selected (cf. Figure 6.11-(b)).

The team adoption goal was defined as:

"Automate system analysis".

Next, the participant applied AHP to refine the goal's relation with the competitive priorities (cf. Table 6.6). *Quality* was defined as the most important competitive priority with 68.0% of relevance, followed by *Lead Time* (19.8%), and not so far *Scope* (13.0%) (cf. Table 6.6). Belonged to the five first from the *Prioritized list of candidate capabilities* a capability from *Scenario Modelling* focus area which the team had no capability insofar (cf. Figure 6.11). The same for *System Behavior Testing*. Other focus areas from this set of capabilities were *Requirements Modeling*, *System Function Specification*, and *Logical Architecture Modeling* (cf. Figure 6.11-(c)).

REFLECTIONS ON THE MODEL: The case study 4 participant stated that *"the model presents things that could be done and currently are not"*. The approach makes improvement opportunities explicit; thus, adherence to requirement **R3**. The participant suggested that "identify and document" capabilities should be split (*i.e.*, identifying without documenting shall be possible). He also suggested that the approach could support partial capability implementation. Considering the team's de-



Context Analysis	Operational Context			A	B	C	D	E		F	G	
	Knowledge Context	🟦	A	B	C		D					
Requirements	Scoping	A	B	C	D	E						
	Goal Modeling	A	🟦	🟦	🟦	B	C	D	E	F		
	Scenario Modeling		A	B	C	D	E	F		G		
	Requirements Modeling	🟦	🟦	A			B	C	D	E	F	G
System Functions	Sys. Function Modeling	A			B		C					D
	Sys. Function Specification			A	B		C		D	E	F	
	Event Chain Modeling				A		B	C			D	
	Mode Modeling				A		B		C		D	
Architecture	Log. Architecture Modeling	🟦	A	B	C	D		E	F	G		
	Log. Component Modeling			A		B	C	D		E		
Testing	System Behavior Testing				A	B	C			D		
Technical Implementation	Technical Architecture Modeling	🟦	A	B	C		D	E	F			
	Technical Component Modeling	🟦	🟦	A			B		C		D	

Figure 6.11: Case study 4 - (a) blue capabilities: current team MBSE maturity profile, (b) green and yellow capabilities: candidate capabilities for adoption, (c) yellow capabilities: five first prioritized capabilities.

velopment methodology, *i. e.*, Scrum, receiving such feedback seems reasonable. We see this as a granularity issue on the capabilities' description. If a capability can be partially implemented, it should be broken down into two full-implementable and less complex capabilities. The participant perceived the time spent to execute the approach compatible with the perceived benefits (thus adherence to requirement **R1**).

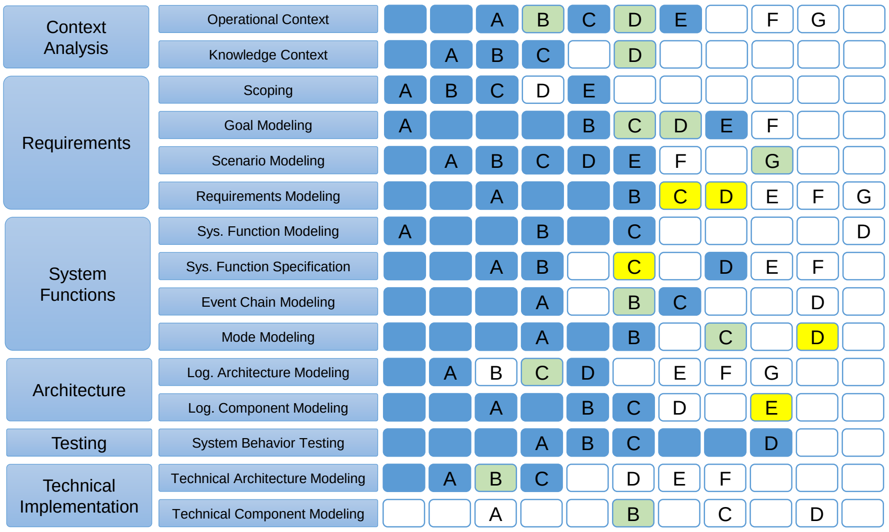
6.4.1.6 *Case study 5 (CS5)*

The team from case study five developed recruiting and staff mobilization information systems for military operations and had four members. Most of their work at the time was related to systems development for replacing analogical business processes. The case study data was gathered in one and a half-hour long face-to-face interview with the system manager.

The team member stated that the team adoption goal was:

"Improve the ability of the team".

This goal is quite generic, and it is yet not clear which benefits can help fulfill it. Further, we performed the task *Refine goal* and the results put great emphasis on *Lead Time* (43.4%), *Scope* (34.6%), and *Quality*(18.3%) (cf. Table 6.6). From the prioritized list of candidate



Context Analysis	Operational Context			A	B	C	D	E		F	G	
	Knowledge Context		A	B	C		D					
Requirements	Scoping	A	B	C	D	E						
	Goal Modeling	A				B	C	D	E	F		
	Scenario Modeling		A	B	C	D	E	F		G		
	Requirements Modeling			A			B	C	D	E	F	G
System Functions	Sys. Function Modeling	A			B		C					D
	Sys. Function Specification			A	B		C		D	E	F	
	Event Chain Modeling				A		B	C			D	
	Mode Modeling				A		B		C		D	
Architecture	Log. Architecture Modeling		A	B	C	D		E	F	G		
	Log. Component Modeling			A		B	C	D		E		
Testing	System Behavior Testing				A	B	C			D		
Technical Implementation	Technical Architecture Modeling		A	B	C		D	E	F			
	Technical Component Modeling			A			B		C		D	

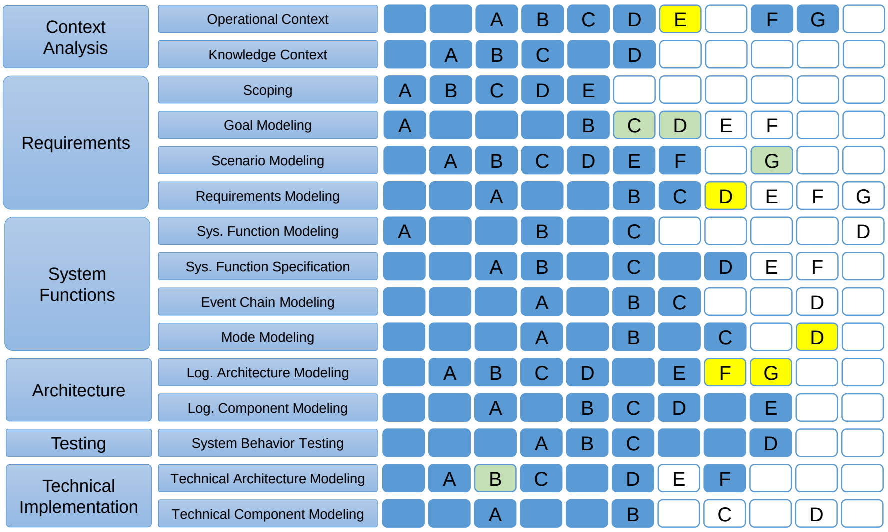
Figure 6.12: Case study 5 - (a) blue capabilities: current team MBSE maturity profile, (b) green and yellow capabilities: candidate capabilities for adoption, (c) yellow capabilities: five first prioritized capabilities.

capabilities, among the first five (cf. Figure 6.12-(c)) were two capabilities from the *Requirements Modeling* focus area (RM C and RM D). According to the *Maturity profile*, the team had only two implemented capabilities from this focus area, and further development would most likely bring great benefit towards the adoption goal.

REFLECTIONS ON THE MODEL: The participant perceived the approach *"very important"* and it *"gives the opportunity to think"* about the processes of the team. He also said that it *"shows the things that are done well, and what possible improvement points are"* (adherence to **R3**). He suggested a more elaborate description considering capability adoption and stated the execution effort was acceptable (compliance to **R1**).

6.4.1.7 *Case study 6 (CS6)*

Case study six was performed in a team specialized in system cyber-security. Its twelve members' duty was to support other teams in the same organization regarding cyber-security. Their duties involved inspection of files for malware and proof testing other teams developed systems. Much of the work they need to carry was mainly repetitive, and any software development was for internal use (*e.g.*, scripts). This setting could be seen in their *Maturity profile* (cf. Figure 6.13-(a)),



Context Analysis	Operational Context			A	B	C	D	E		F	G	
	Knowledge Context		A	B	C		D					
Requirements	Scoping	A	B	C	D	E						
	Goal Modeling	A				B	C	D	E	F		
	Scenario Modeling		A	B	C	D	E	F		G		
	Requirements Modeling			A			B	C	D	E	F	G
System Functions	Sys. Function Modeling	A			B		C					D
	Sys. Function Specification			A	B		C		D	E	F	
	Event Chain Modeling				A		B	C			D	
	Mode Modeling				A		B		C		D	
Architecture	Log. Architecture Modeling		A	B	C	D		E	F	G		
	Log. Component Modeling			A		B	C	D		E		
Testing	System Behavior Testing				A	B	C			D		
Technical Implementation	Technical Architecture Modeling		A	B	C		D	E	F			
	Technical Component Modeling			A			B		C		D	

Figure 6.13: Case study 6 - (a) blue capabilities: current team MBSE maturity profile, (b) green and yellow capabilities: candidate capabilities for adoption, (c) yellow capabilities: five first prioritized capabilities.

which disclosed a high degree of automation. We interviewed the team's project manager in a face-to-face meeting lasting one and a half hours. Only nine candidate capabilities were selected for this case study due to the high maturity level of its processes. The participant defined the MBSE adoption goal as

> *"Bring greater automation to artifact treatment tasks"*.

After performing the *"Refine goal"* task (cf. Table 6.6), *Quality* was defined as the most important competitive priority (61%) followed by *Scope* as second (26.2%). After the *Prioritize candidate capabilities* task, the five best-ranked candidate capabilities were two from *Logical Architecture Modeling* and one from *Operational Context*, *Requirement Modeling*, and *Mode Modeling* (cf. Figure 6.13-(c)).

REFLECTION ON THE MODEL: The case study participant stated that *"without doubt the approach helps to decide the next improvement steps"* (compliance to **R3**). He declared that he would have liked to spend more time applying the approach, *i. e.*, the approach application was perceived to be hastily, and he would have liked to discuss the concepts in more detail. We were trying to apply the approach using less time as possible to make it compliant with R1. Anyhow, his remarks are valid, and perhaps we should be more sensitive to the team members' need for more discussion.

Table 6.6: Instance of *Goal refinement* for each goal of the case studies.

Goal/Competitive priority	**Cost**	**Lead T.**	**Quality**	**Scope**
CS1: Enhance communication between software and system development teams.	4.7%	11.9%	29.8%	**53.6%**
CS2: Improve knowledge transfer.	3.7%	35.3%	**45.0%**	16.0%
CS3-1: Reduce rework due to communication issues and scope definition.	3.6%	**50.3%**	13.7%	32.4%
CS3-2: Improve effort estimation.	5.4%	**68.0%**	14.9%	11.7%
CS4: Automate system analysis.	4.9%	19.8%	**62.3%**	13.0%
CS5: Improve ability of the team.	3.7%	**43.4%**	18.3%	34.6%
CS6: Automate artifact treatment tasks.	3.0%	9.8%	**61.0%**	26.2%
Average:	4.1%	34.1%	**35.0%**	26.8%

Table 6.7: Top 5 capabilities from the sensitivity analysis one-at-a-time for each competitive priority plus *Refined MBSE adoption goal* from case study 1.

Posi.	**Cost**	**Quality**	**Lead time**	**Scope**	**CS1**
1.	SFS D	SFS C	SFS D	RM B	SFS C
2.	SFS C	SFS D	SFS C	SFS D	SFS D
3.	SFM C	SFM C	SFM C	SFS C	RM B
4.	LAM C	LAM C	OPC C	RM A	OPC C
5.	OPC C	OPC C	LAM C	OPC C	SFM C

6.4.2 *Quantitative evaluation*

In this subsection, we intend to test if the approach adheres to the requirement ***R2***. *The approach must react to different types of relationships between input and output variables*. For this means, we would like to test how different values attributed to competitive priorities could change the prioritization of candidate capabilities. One way to achieve this is through sensitivity analysis which performs the role of ordering the strength and relevance of the inputs in determining the variation in the output (Vetschera, 2006). We selected one of the most common approaches called One-at-a-time (OAT) sampling (Saltelli, 2008) where only one parameter changes values between consecutive simulations. We decided to change the importance of competitive priorities to simulate the influence of different goal refinements in the approach output. For this simulation we used the *Team's benefit potential* values (cf. Table 6.5) and candidate capability set from the case study 1 (cf. Section 6.4.1.2). We also added the results of the *Refined MBSE adoption goal* from case study 1 (CS1) to the comparison.

The simulations consisted of selecting one competitive priority to give the most significant importance, whilst the other competitive priorities were given equal priority compared to each other during the task *Refine goal*. This rating resulted in 750 points for the most important competitive priority and 83 points for the others. We repeated this task four times, one for each competitive priority. Then we analyzed the five first candidate capabilities from the task *Prioritize candidate capabilities* for each of these configurations (cf. Table 6.7).

In this analysis, the capabilities *System Function Specification* C and D, SFS C and SFS D for short, were ranked most relevant twice each (SFS C in the *Quality* and CS1 groups, and SFS D in the *Cost* and

Lead time groups), and were present in all groups. This outcome is no surprise since these capabilities can deliver many benefits with strong influence, *i. e.,* they summed many points from the MBSE expert assessment (cf. Table 6.3). Together with capability OPC C, they were always present in all groups (cf. Table 6.7). The *Scope* group had two capabilities from the *Requirements Management* focus area (*i.e.,* RM A and RM B) as most important while only the CS1 group had RM B in its list. This focus area is essential for choosing the best features, thus the high performance in this group. With this analysis, we present evidence that the model complies with requirement **R2**. The analysis was done considering the case study 1 team-specific relations between benefits and competitive priorities. Different teams with distinct values for the relations would provide other results; thus, the same analysis between different case studies would not make sense.

6.5 DISCUSSION

This section discusses cross-case studies impressions, the approach mechanics, its novelty, and limitations.

6.5.1 *Cross-case analysis*

Concomitantly while presenting the case studies, we discussed the particularities found during their execution in a paragraph named *Reflections on the model*. In this subsection, we would like to discuss and compare cross-case studies impressions.

A thing that caught our attention after performing the task *Refine goal* was that in all case studies, **the competitive priority *Cost* was categorized as the least important** (cf. Table 6.6). We have a couple of hypotheses regarding this phenomenon. One hypothesis is that the teams did not face fierce market competition regarding the developed system's cost; perhaps they only provided services on a fixed contract with little room to benefit from this competitive priority. Another hypothesis is that the case study participants did not belong to a hierarchy level where cost is of concern; thus, emphasizing the other competitive priorities. A third hypothesis is a perception that process improvement is a costly endeavor; thus, people hardly see it as a way to save money, although better processes are meant to reduce costs. We cannot be sure of the cause, and further empirical research is needed to assess these hypotheses.

During the case studies, the task execution order was not always the same, and this happened due to the availability of case study participants or the limited interview time available. In case study 2 (cf. Section 6.4.1.3), the task *Characterize team context* was executed before task *Define MBSE adoption goal* and the MBSE adoption goal defined by the team gave us the impression that the participants were influenced by what they learned during the approach execution when defining the goal. A similar phenomenon happened in case study 4, where the goal definition seemed influenced by capability description during the task *Assess team process maturity*. To prevent such issues, we believe that the most appropriate task order would be first to perform the *Define MBSE adoption goal* task, then *Characterize team context* and finally *Assess team process maturity*. This ordering shall prevent influence in the goal definition regarding the MBSE benefits and capability description.

The approach was designed to be executed considering a single MBSE adoption goal. However, in case study 3 (cf. Section 6.4.1.4), we performed the approach using two goals. The team was unable to decide on a single goal; then, we suggested trying with both. For this, we executed the tasks *Refined MBSE adoption goal* and *Prioritize candidate capabilities* one time for each goal. Since both goals were for the same team, the other tasks needed to be executed only once. The five best candidate capabilities were similar for both adoption goals since the *Refined MBSE adoption goal* artifact for both goals were similar (cf. Table 6.6). In the end, it became clear which capabilities were very interesting for both goals, and this gave the team higher assurance on the results. However, we still think that deciding on a single goal is better, and using two goals makes interpreting the results less objective.

We also perceived differences in the approach execution concerning development methodology. In case study 4 (cf. Section 6.4.1.5), during the task *Assess team process maturity*, few capabilities could be checked as implemented in the team. The reason for this phenomenon was the team's development methodology, namely Scrum. Agile methodologies advocate minimal use of artifacts, so the team's effort is solely focused on system development (Beck, 2000). In turn, the capabilities of the MBSE MM are verified through the existence of artifacts created in the development process. We believe this is a limitation of the MBSE MM, which might need to be modified for considering artifact-low methodologies (Wagenaar et al., 2018).

6.5.2 *Approach mechanics*

The type of problem we are proposing to solve can be classified as a multi-attribute problem with a given set of alternatives (Franco and

Montibeller, 2011; Vetschera, 2006). The set comprises capabilities that could be implemented in a development team, and the attributes are benefits afforded once these capabilities are implemented, which must be aligned with the adoption goal.

The approach accuracy relies on the team context perception by the team members, thus gathering as many associates as possible from relevant roles enriches the approach's outcome. A drawback is the approach execution cost increase. The participant's experience should also be of concern; more senior associates usually can provide better answers.

The amount of points a candidate capability receives as a result of the task *Prioritize candidate capabilities* is variable and depends on the total number of selected candidate capabilities. Thus, the points a candidate capability receives can only be used to compare different "runs" of the approach when considering the same candidate capabilities.

The *Candidate capability affordances* artifact contains information on the benefits each candidate capability can deliver and their respective intensity, using the no-low-high numeric scale (cf. Section 6.3.1.5). The sum of these values in each capability can reach up to 20 (*i.e.,* at most 2 per relationship times ten benefits). Within this selected set of 53 capabilities from the case studies (cf. Section 6.4.1.1), the capability with the highest value had 19, while the capability with the least got only 3 (cf. Table 6.3). Such difference is realistic since some key capabilities can greatly benefit some teams (especially ones with higher maturity *e.g.,* LAM G) while others deliver just little benefits. This phenomenon consequence is that these lower-scoring capabilities are less likely to be prioritized than the high-scoring ones.

Considering the benefits side of the same relation, the value sum range depends on the number of capabilities considered. For the 53 candidate capabilities, this value can range between zero and 106 (*i.e.,* 53 capabilities times 2). The benefit with highest value sum was *Improved verification and validation* with 67, followed closely by *Improved quality of specification* (66) and *Better communication* (65). Summing higher values means that many capabilities can deliver this benefit or with stronger intensity. The benefit with the least value sum was *Easier reuse of existing artifacts* (43).

The instantiation method execution (cf. Figure 6.3) required between one hour and a half, and four hours for the case studies. The task *Refine goal* requires six tool-supported comparisons between competitive priorities using AHP. *Characterize team context* requires assessment of 40 relations in a three-degree scale. *Assess candidate capabilities affordaces* requires the assessment of 800 relations when applied in the whole model (80 capabilities times 10 Benefits). The small amount of time needed for running the approach, the relatively few assessment units, and the possibility of reuse is additional evidence that the approach fulfills the requirement **R1**.

An improvement opportunity for the approach that we realized after the case studies is replacing the Scope competitive priority name for something more intuitive. Scope is usually associated with the size of a project or systems, while in our approach, the term is used to define the capacity to produce market-competitive functionalities. At times, we needed to remind the case study participants, not considering the times we did not perceive the misunderstanding and that no amendment was made. The problem was not the concept but the name, which was misleading at times. Perhaps using a different name such as Functionality could prevent that.

6.5.3 *The approach novelty*

Our approach segments and relates different domains (organizational, MBSE) and allows experts' knowledge reuse by non-experts. That is the main difference from other prioritization techniques, which requires knowledge about the elements in question and the motivation for their prioritization at the same time. Our approach uses AHP to refine the process improvement goal in the four competitive priorities. Process engineering approaches are using AHP to support software process prioritization (Lorenz and Brasil, 2014) and business process re-engineering (Rao, Mansingh, and Osei-Bryson, 2012). However, AHP requires executors to have in-depth knowledge about the assessment's goal and elements. The executors, in our case, are the team members who know the former but not the latter. Thus, we need experts to report the gains related to capability implementation. Our approach tackles this issue by collecting and linking the MBSE experts' knowledge with the team members' context and goal knowledge. As more MBSE expert assessments are gathered, the more robust and precise the approach becomes. The *Candidate capability affordances* can be reused in different teams and enhanced through further expert analysis. Additionally, AHP is not adequate with many options because the number of comparisons increases drastically, and the focus is lost among all the possibilities.

6.5.4 *Limitations*

In any process adoption endeavor, several factors need to be taken into account (*e.g.*, project constraints), many of which are not currently contemplated by our approach. Thus, the approach's results should not be taken as it is, but be used as a starting point for discussions on MBSE method adoption. Further dimensions not considered in this model but relevant when adopting MBSE are costs, project availability for piloting, time flexibility, knowledge of the team, and criticality of projects (Amorim et al., 2019; Vogelsang et al., 2017).

6.6 RELATION BETWEEN THE GOAL-BENEFIT MODEL AND THE METHOD VALUE ASCRIPTION MODEL

The goal-benefit elements instantiates the method value ascription model elements (cf. Chapter 5) as seen in Figure 6.14. The approach elements are depicted using the orange color, elements in blue are related to the MBSE MM (*MBSE Candidate Capabilities, MBSE Maturity Model, and Team's MBSE Maturity Profile*). The *Adoption goal* element, which is an essential part for method value ascription, is instantiated by the *Refined MBSE adoption goal*. The *Value Methods* are instantiated by the *MBSE Candidate Capabilities* selected after the *Assess team process maturity* task is performed. The *MBSE Benefits* instantiate the *Benefits* elements which, together with the *Costs* element, builds the *Valuation Relationship*. The *Value Method Qualities* is instantiated by the *Capability affordances*. The *Team's Context* is composed of two elements, the *Team's benefit potential* and the *Team's MBSE Maturity Profile*.

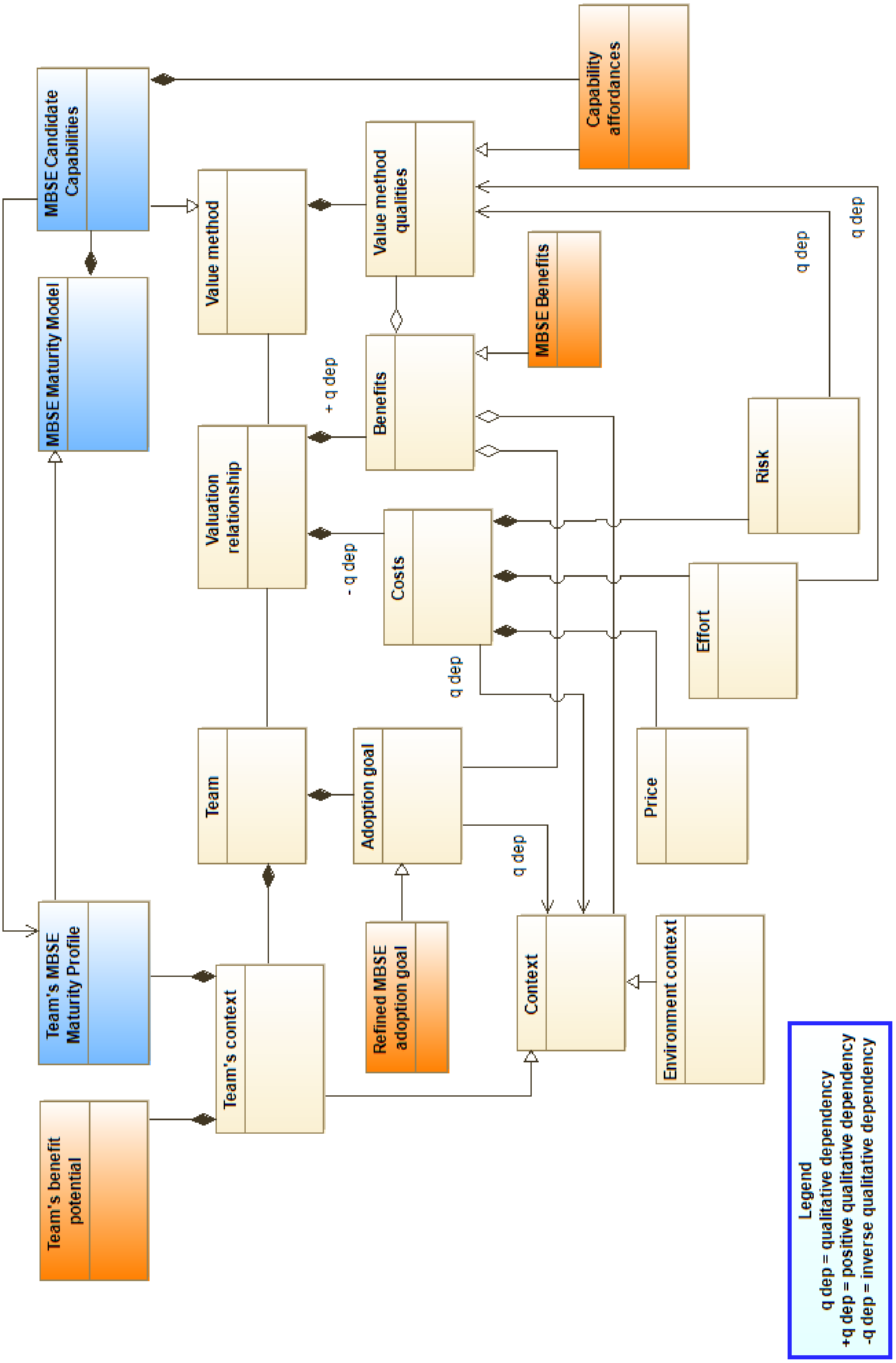


Figure 6.14: Elements of the approach in relation to the value method ascription model (cf. Chapter 5).

7

STATE OF THE ART

This chapter presents the relevant work related to this thesis's topic, namely studies on MBSE adoption experiences, adoption best practices, and method selection. After summarizing each work, we provide a paragraph to discuss the relation with our thesis. Our goal was to describe the relevance of selected work units and compare it to the contributions of this thesis.

Section 7.1 outlines works on MBSE adoption. Based on empirical surveys, the selected studies attempting to extract impressions from practitioners' experiences towards MBSE adoption. Additionally, some works present best practices and strategies to improve adoption success. This subsection strongly relates to Chapter 3, where we analyze the forces that foster and hinders MBSE adoption in embedded systems organizations, and Chapter 4, where we present eighteen strategies and best practices improving the adoption endeavor efficiency.

Section 7.2 outlines work on organizational context modeling and its influence on method selection. We show that the definition of factors is very domain-dependent, and their relation with method selection is not easy to establish due to domain-specific characteristics. In Chapter 5, we denote the relevance of the context towards method value ascription and differentiate it between environment and team context. In Chapter 6, the team's context is modeled in two ways, through the *Maturity profile* and the *Team's benefit potential*.

Section 7.3 outlines work on goal modeling and respective concrete actions towards goal fulfillment. We show that the proposed models are pretty good for informative and elicitation purposes, but they lack prioritization methods or more sophisticated decision support. In Chapter 5, we describe how the goal influences the benefits' perception. In Chapter 6, the MBSE adoption goal is refined using competitive priorities and related to benefits harvested upon method adoption.

Section 7.4 outlines work on method selection rationale in different software engineering domains. We show that the current work is relevant but too specific or general. However, the selected approaches provided us with suitable material for building our approach (cf. Chapter 6) fitted for our needs.

Section 7.5 outlines work on the area of Benefits Management. We look into the most prominent diagrams, namely the Benefits Dependency Network, the Benefits Dependency Map, and the Results Chain Modeling. Finally, we compare these diagrams elements with the goal-benefit model elements proposed in Chapter 6.

7.1 MODEL-BASED SYSTEMS ENGINEERING ADOPTION

MBSE method adoption is not so straightforward, thus several works address this topic.

Bone and Cloutier, 2010 report on a survey conducted by the OMG, in which participants were asked about MBSE adoption within their organization. The main reason to adopt MBSE was "to improve the quality of requirements and design to reduce downstream defects", which was selected by 72.9% of the respondents. *Culture and general resistance to change* was identified in the study as the largest inhibitor for MBSE adoption (cf. Figure 7.1). The study found that SysML is being used primarily for large-scale systems. Their results show overall satisfaction with MBSE (respondents rated satisfaction with 3.77 out of 5.00 points). Despite the high satisfaction of respondents, inertia is difficult to overcome in larger companies. They concluded that issues with MBSE adoption and usage might lie outside of MBSE and SysML.

Relation to our thesis. Their study scope and findings have some overlaps with our studies scope (*e.g.,* inhibitors for MBSE adoption) and findings in both of our studies (cf. Chapter 3 and Chapter 4). However, we had a more exploratory (*i.e.,* open coding (DeCuir-Gunby, Marshall, and Mcculloch, 2011)) approach using interviews whilst they built hypotheses before gathering empirical evidence through surveys. We used surveys to confirm our theory building (cf. Chapter 4).

Motamedian, 2013 performed an applicability analysis for MBSE. Similar to the results of Bone and Cloutier (Bone and Cloutier, 2010), Motamedian found that MBSE is widely used in specific application areas. She reported that 50–80% of respondents declared using MBSE in real programs or projects work in defense and aircraft industries. In contrast, overall responses, only around 10% of participants claimed that they use MBSE in their organization. The study identified *lack of related knowledge and skills* as the main barrier to MBSE introduction. She also reports on an online survey among MBSE practitioners. The

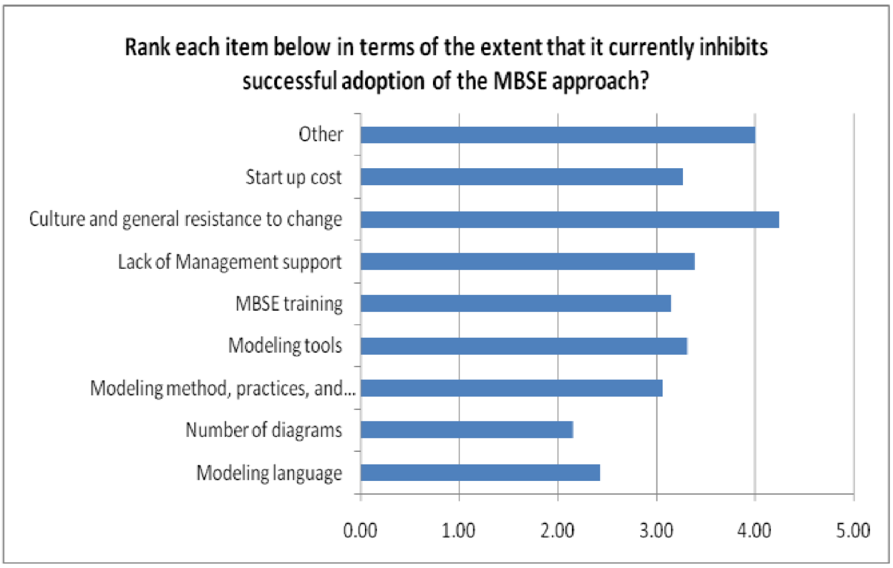


Figure 7.1: Inhibitors to MBSE adoption (from (Bone and Cloutier, 2010)).

author's goals were: (1) to highlight the position of MBSE in real projects, (2) to assess the popularity rate of MBSE concept among engineers, and (3) the usage besides the advantages (i.e., barriers and concerns of using "modeling language" and "modeling tools") in MBSE efforts among various industries.

Relation to our thesis. The aforementioned study overlaps with ours when looking for barriers and concerns to MBSE introduction. In our work, we used the term *hindering forces* (cf. Chapter 3) and their finding as the main barrier for adoption was the same as ours (*i. e.,* Skill of employees).

Fosse and Bayer, 2016 present in an MBSE workshop the lessons learned while implementing MBSE in the Jet Propulsion Laboratory of the American National Space Agency (NASA) at the Mars2020 project. The MBSE adoption goals of the aforementioned project were (1) achieve better awareness of the technical baseline, (2) improve the communication, understanding, and visibility of the design, (3) focus on heritage design deviations, and (4) let products drive model implementation. They list and discuss the benefits of MBSE, compares it to traditional Systems Engineering (SE),

Relation to our thesis. Although their study is not peer-reviewed and comprises only a single case study, the project that they analyzed (MARS2020) was very relevant, many of their findings are aligned to ours (*i. e.,* their lessons learned matched with many of our strategies and best practices (cf. Chapter 4)). Further work is required to guarantee generalization.

Carroll and Malins, 2016 describe the benefits of MBSE adoption extracted from a series of case studies. At the end of the study, the authors present a list of implementation lessons drawn from the findings. For them, the cultural changes necessary to implement an MBSE approach successfully (roles, rewards, behavior, and support at all levels) were described as the most difficult challenges to overcome.

Relation to our thesis. Our list of strategies and best practices (cf. Chapter 4) is similar to the implementation lessons presented by the authors. They also elicited benefits of MBSE adoption, which is not the focus of our research, but it is part of our approach for selecting MBSE methods (cf. Chapter 6).

(Kuhn, Murphy, and Thompson, 2012) and (Aranda, Damian, and Borici, 2012) reported on the experience of using MBSE in large companies. Kuhn et al. (Kuhn, Murphy, and Thompson, 2012) focus on contextual forces and frictions of MBSE adoption in large companies. They found that *diffing in product lines*, *problem-specific languages and types*, *live modeling*, and *traceability between artifacts* are the main drivers for adopting MBSE. Aranda et al. (Aranda, Damian, and Borici, 2012) focus more on developers and infrastructure changes. They conclude that MBSE brings developers closer, disrupts organizational structures, and achieves productivity improvements.

Relation to our thesis. The frictions, drivers, and contextual forces from (Kuhn, Murphy, and Thompson, 2012) are very similar to the hindering and fostering forces in our work (cf. Chapter 3). The authors from (Aranda, Damian, and Borici, 2012) also investigated the benefits of adoption, similar to our minor contribution (cf. Section 2.2.2).

karban et al. (Karban et al., 2010) reported on the advantages of MBSE introduction with a concrete example. They analyzed the development of a telescope system with SysML modeling. As this is a complex system, it was necessary to model variants of function, interfaces, and structure. The introduction of MBSE was evaluated very positively.

Relation to our thesis. This report of a successful MBSE adoption shed light on the fostering forces (especially pull) that makes MBSE adoption

HOHL ET AL., 2016 did a similar work as we and inspired the categorization schema from Chapter 3. They performed their study using similar categorization of motivational factors for process change (*i. e.,* Forces of Progress (Klement, 2018)). In their case, the process adoption was agile methods.

Relation to our thesis. The authors consider two extra forces, triggers, a fostering force, and context, which is a hindering force. We did not include these in our analysis because the trigger and the push force are very similar, and the context force bears similarities to the other two hindering forces. Additionally, context can be a fostering force at times.

HUTCHINSON ET AL. published many papers that focus on the adoption of Model-Driven Engineering (MDE) in the context of software development (Hutchinson, Rouncefield, and Whittle, 2011; Hutchinson, Whittle, and Rouncefield, 2014; Hutchinson et al., 2011; Whittle, Hutchinson, and Rouncefield, 2014; Whittle et al., 2017). Their research comprises case studies built around semi-structured interviews and online surveys devised to gather information about MDE usage through closed questions (e.g., diagrams used for modeling, modeling languages, MDE purpose of use). In (Hutchinson, Rouncefield, and Whittle, 2011), the authors describe the practices of three commercial organizations as they adopted MBSE. In (Hutchinson, Whittle, and Rouncefield, 2014), their findings are summarized into ten dimensions of organizational attitude to MDE adoption, half being helpful responses and the other half being unhelpful.

Relation to our thesis. The authors' responses are described in a general manner. Table 7.1 provides the relations between their helpful responses and the best practices described by us in Chapter 6. In (Hutchinson et al., 2011; Whittle, Hutchinson, and Rouncefield, 2014), the authors state that MDE should be tried on projects that "can not fail". We also reported similar finding as presented by BP02. In (Whittle et al., 2017), they present a taxonomy of MDE tool related issues; 'Chaining tools together' and 'Flexibility of tools' are covered by BP06, 'Sustainability of tools over the long term' and 'How to select tools' are related to BP08. Albeit the intersection of our findings with the work of the aforementioned authors, our contribution goes further:

- Their focus was MDE (a software development methodology that encompasses models and code generation) whilst our focus is on MBSE (a systems engineering methodology devised for covering mechatronics, electronic, and software parts). MDE is a subset of MBSE (i.e., MDE $\subseteq$ MBSE).

Table 7.1: Hutchinson's responses and related best practices.

Response	**Best practice**
Adaptive	BP14, BP16, BP17
Business led	-
Committed	BP03, BP07, BP10, BP18
Iterative	BP05
Progressive	BP05

- Their case study findings are based on anecdotal evidence, whilst we used questionnaires to empirically validate and generalize our conclusions with practitioners in a triangulation (Denzin, 2006) fashion (i.e., the questionnaire was built based on the conclusions we derived from the interviews (cf. Section 4.2)).

- Their findings are laid out within discussions of case studies whilst we provide a crisp list of best practices and a prioritized list of the six most important ones, aiming to help practitioners know where to focus.

OTHER STUDIES Besides these meta-studies on MBSE adoption, several case studies exist on applying model-based techniques to complex systems in different domains (e.g., telescope systems (Karban et al., 2010), railway (Böhm et al., 2014), automotive (Vogelsang, Femmer, and Winkler, 2015), maritime traffic (Vogelsang et al., 2014)). Not specific to MBSE, but technology transfer, in general, is the focus of Diebold et al. (Diebold, Vetrò, and Fernández, 2015). They identified barriers to knowledge transfer in two German research projects and pointed out solutions. Besides, they propose an evaluation framework for assessing technology transfer in software engineering projects.

7.2 CONTEXT MODELING

In this section, we discuss the relevant studies on attempts to shape the solution space of method adoption and maturity models based on situational factors (*e.g.*, team size, development philosophy, team's sector).

SITUATIONAL METHOD ENGINEERING provides a solution to the problem of selection, creation, or modification of methods for organizations or their projects based on their contextual characteristics (Henderson-Sellers et al., 2014). It is an extension of the Method Engineering (Brinkkemper, 1996) and has been the focus of numerous authors (Henderson-Sellers and Ralyté, 2010). In this approach, method fragments (or method chunks) are selected from a method database to build tailored methods according to the organization purpose and context. The situational method engineering puts greater emphasis on the organization context during its execution.

Relation to our thesis. The aforementioned approach focuses on in-house construction as opposed to the adoption of ready-made methods. We are interested in the problem of method selection, but not in the creation or modification, since we use capabilities defined by the MBSE MM (cf. Section 2.4). The contingency for method selection proposed by the situational method engineering is not applicable without further ado.

BEKKERS ET. AL. The authors have extensive work on organization situational context and software product management process improvement. In (Bekkers et al., 2008), the authors analyzed which situational factors influence process selection of software product management method fragments (Brinkkemper, 1996). Their results are 27 situational factors in 5 categories and respective influence levels on the method selection. In (Bekkers and Spruit, 2010; Bekkers et al., 2010), the authors propose the Situational Assessment Method for the Software Product Management Maturity Matrix. The proposed approach helps software product managers evaluating and improving their processes through recommendations based on the context of the organization and the organization itself. These recommendations determine which capabilities from the maturity matrix are adequate to the organization being assessed.

Relation to our thesis. The authors' work is very similar to ours in many points; both seek ways to select the best suitable methods according to the organizational context, and both work with focus area maturity models (the MBSE MM in our case). Our work differentiates from theirs on considering the team adoption goal and the benefits generated by the methods.

BAARS AND MIJNHARDT In (Baars et al., 2016), the authors analyzed the influence of organizational characteristics in an information security focus area maturity matrix. They conclude that the maturity framework has a poor model fit and should consider differences between characteristics of organizations. Further, in (Mijnhardt, Baars, and Spruit, 2016), the authors propose a model for relating measurable organizational characteristics in the context of the aforementioned maturity framework. These characteristics help SMEs distinguish and prioritize risk mitigation.

Relation to our thesis. The goals and elements of their studies are very similar to ours, *i. e.*, selecting best fitting capabilities from a maturity matrix. A plus point of their work is that they describe crispy measurements for the situational factors, making their approach more precise. However, they also do not consider the adoption goal for capability selection.

CLARKE AND O'CONNOR (Clarke and O'Connor, 2012) proposes a reference framework of situational factors affecting the software development process. The resulting framework was built by applying data coding techniques from grounded theory into related research. The framework consists of eight classifications and 44 factors, such as the nature of the application(s) under development, team size, requirements volatility, and personnel experience. They claim their framework provides support for practitioners defining and maintaining software development processes.

Relation to our thesis. The problem with situational factors is that they are domain-specific, and their influence on capabilities is hard to validate empirically. We initially wanted to work in this direction, but we soon realized that situational factors could have different meanings in different contexts. When defining *Team's benefit potential*, situational factors are implicitly considered in our approach (cf. Chapter 6), and, from our point of view, this is the most appropriate way to acknowledge them.

7.3 GOAL MODELING

Goal modeling aims to break down goals into sub-goals and relate them to concrete activities for their fulfillment. In Requirements Engineering, it helps to understand and document the stakeholders' goals towards a system, which provides means for deriving systems requirements (Horkoff and Yu, 2011; Van Lamsweerde, 2001). The following frameworks were investigated to understand how these relations could be modeled and which parts could be interesting for our approach presented in Chapter 6.

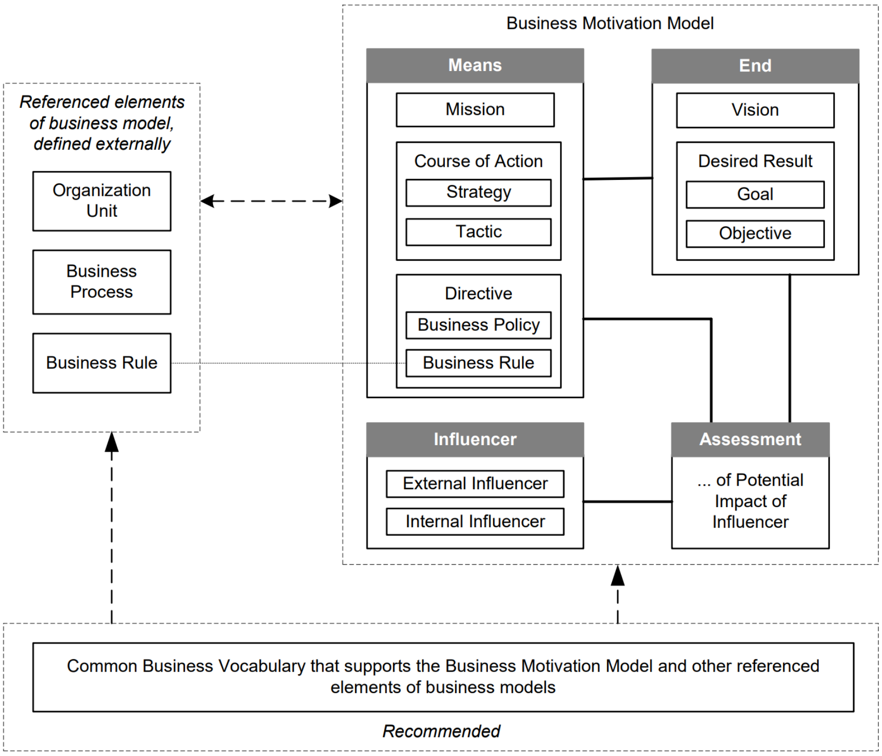


Figure 7.2: Overview of the Business Motivation Model.

BUSINESS MOTIVATION MODEL (OMG, 2010) provides a structure for developing, communicating, and managing business plans. The model defines *"Ends"* and *"Means"* elements (cf. Figure 7.2). The former describes the envisioned state an organization wants to achieve, to either change or maintain its current position relative to its market and competition. The latter describes how the organization chooses to achieve the former and is composed of Mission, Course of Action, and Directives. Ends are refined in Vision and Desired Results, and Desired Results as Goals and Objectives.

Relation to our thesis. The Desired Result from the End element is very similar to the MBSE adoption goal element of the goal-benefit model from Chapter 6. The Course of Action from the Means element is analogous to the Candidate Capabilities of the same model. The Business Motivation Model does a great job relating these elements. However, this model does not provide tools for comparing strategies.

THE ISTAR FRAMEWORK (Yu et al., 2011) is a modeling language that helps to understand the problem domain by modeling the *as-is* and *to-be* situations. It was created for modeling organizational environments and encompasses information systems (*i. e.*, resources), actors and their goals (*i. e.,* goal, soft goal), and tasks. The model represents actors depending on each other to perform tasks, use resources, and finally achieve their goals. The model exposes dependencies between these elements, which are many times competing.

Relation to our thesis. The iStar framework is very efficient for identifying stakeholders' goal and their relations, enabling engineers to better reason about systems. In our approach, we consider a single goal and stakeholder (*i. e.,* MBSE adopting organization), and our concern is how to relate the MBSE methods with the adoption goal meaningfully.

THE KAOS FRAMEWORK (Van Lamsweerde, 2001) is a goal modeling framework for requirements engineering. KAOS is an acronym for Knowledge Acquisition in autOmated Specification. The framework has four types of interlinked models, namely the *goal model*, the *responsibility model*, the *object model*, and the *operation model* (cf. Figure 7.3). The goal element can be either refined into subgoals describing how it can be reached or justified by higher-level goals that explain its introduction. The goal refinement ends either at an expectation (when related to an environment agent), a requirement (related to a system agent), or a domain property.

Relation to our thesis. The KAOS framework makes explicit the justification of system requirements by linking them to higher-level goals. The goal of our approach (cf. Chapter 6) is to link MBSE methods with adoption goals to provide decision support when selecting those methods. Thus, the KAOS framework helped us to see the possible kinds of relations that goals can have.

GOAL STRUCTURING NOTATION The Goal structuring notation (GSN) is a modeling language for documenting safety, security, and dependability arguments. The diagrams are built through logic-based maps and are used to compose assurance cases for critical systems. They depict design goals related to strategies, assumptions, solutions, and justifications (cf. Figure 7.4). GSN is used within the Nuclear, Defence, Aerospace and Rail domains.

Relation to our thesis. The GSN bears a similar goal as our approach presented in Chapter 6, namely the relation between goal and concrete steps. Their purpose is different, however. In GSN, this relation serves to provide arguments that the respective goal is fulfilled, whilst in our approach, the purpose is to assess the best way of achieving the goal considering the candidate solutions.

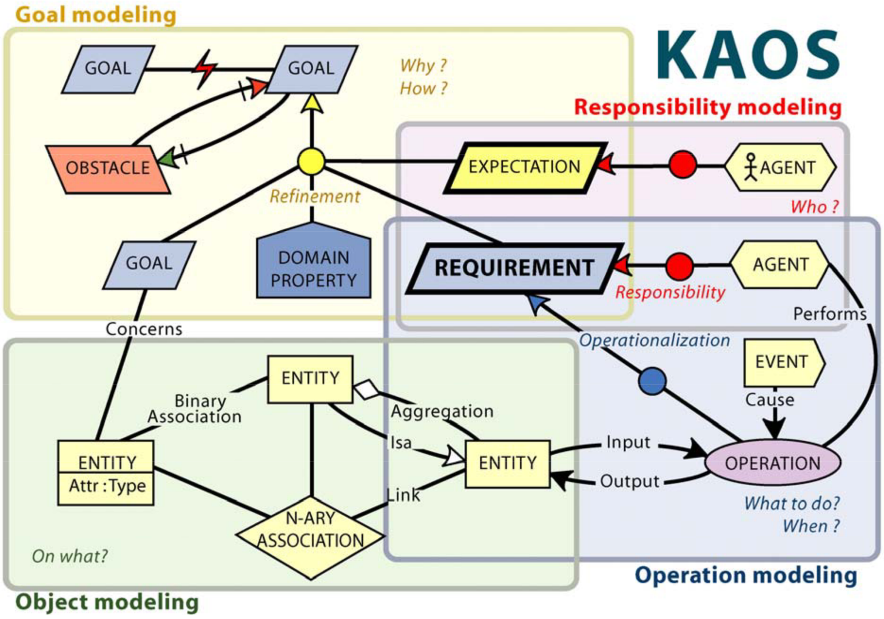


Figure 7.3: Overview of the KAOS meta-model (Dardenne, van Lamsweerde, and Fickas, 1993).

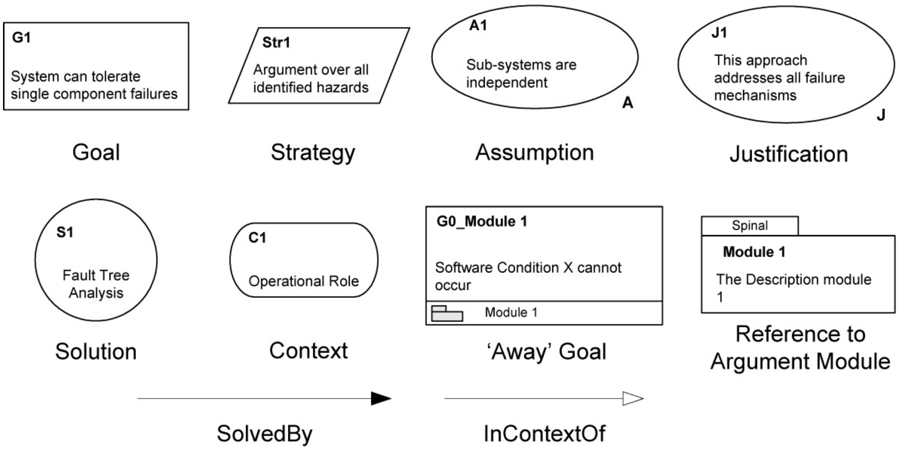


Figure 7.4: Elements of the Goal Structuring Notation.

GQM+STRATEGIES (Basili et al., 2007) is an extension of the Goal-Question-Metric approach (Basili, Caldiera, and Rombach, 1994) to support organizational goals. This approach links software measurement goals to the software organization's higher-level goals and further goals and business-level strategies. These relationships allow higher-level decisions to be supported by measurement data. Elements of this approach are strategy, goals, assumptions, and context. Goals are differentiated into growth goals, success goals, maintenance goals, and specific focus goals.

Relation to our thesis. The approach proposed is very similar to ours (cf. Chapter 6) in the sense that both seek to link organizational-level goals to ways to achieve these goals. While the author's approach is very general and its implementation requires further ado, our approach is MBSE specific and can be easily used without any deviations. There are extensions to address rationales of decisions in a more concrete manner, such as (Mandić and Gvozdenović, 2017; Trendowicz, Heidrich, and Shintani, 2011). However, they still require further ado and the participation of domain specialists. Also, the rationale of decisions does not reach the level of granularity we believe is reasonable.

7.4 METHOD SELECTION

In this subsection, we discourse about method selection work.

ÅGERFALK AND WISTRAND (Ågerfalk and Wistrand, 2003) propose the inclusion of the rationality dimension in the description of methods. This dimension shall store the method creator's values and assumptions about the problem domain when creating the method. It is divided into two different kinds of sub-rationale: method prescriptions anchored in goals, referred to as goal rationale, and goals anchored in values, namely value rationale (cf. Figure 7.5). In their model, goal achievement relations describe hierarchies between goals, and values are related to each other by a value anchoring relation.

Relation to our thesis. The reason for using a particular method can be based on its contribution to other higher-level goals and its realization of (parts of) the method's underlying philosophy as expressed by identified values. The proposed approach bears some similarity to ours (cf. Chapter 6). However, this approach does not compare methods to select the most interesting ones for the organization. Also, the value of a method is not static and depends on the organization's goal and context (cf. Chapter 5).

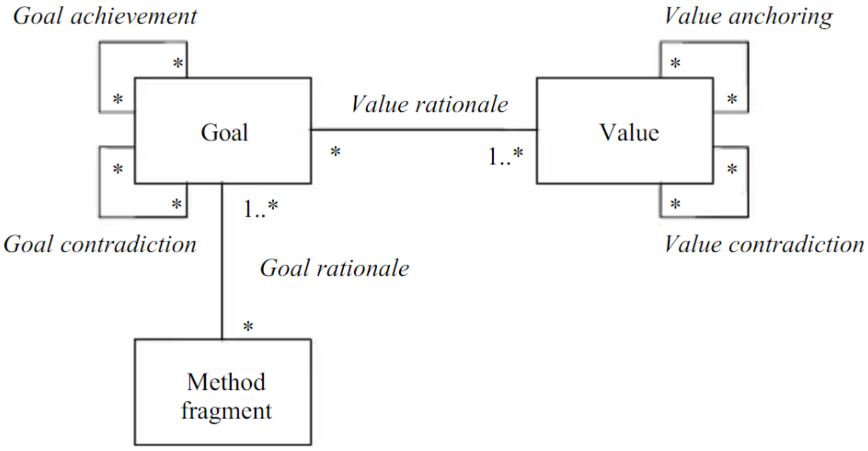


Figure 7.5: Method rationale as constituted by goals, values and their relationships (Ågerfalk and Wistrand, 2003).

CONBOY AND FITZGERALD (Conboy and Fitzgerald, 2010) developed a framework to improve eXtreme Programming (XP) (Beck and Andres, 2004) method tailoring effectiveness. They extracted best practices from a literature review and interviews with researchers and validated their findings with practitioners, which are summarized in the following set of recommendations for research and practice:

- Explicit statement of method boundaries: teams had difficulty identifying where XP should and should not be applied.

- Contingency built into the method itself to guide tailoring: tailoring efforts were based on team members' opinions and preferences.

- Clear description of method rationale behind method practices: advantages and disadvantages of each practice was unclear.

- Independence of individual method practices: the social and softer nature of XP practices makes it very difficult to identify co-dependencies and knock-on effects between practices.

- Identification of project context dependencies: XP adoption decision rarely involved a formal analysis of situational dependencies and was often driven by one single "champion" without input from team members or stakeholders.

- Familiarity with a range of methods and method fragments: easier practices were selected due to developers' lack of knowledge of all XP practices.

- Disciplined and purposeful approach to method tailoring: inconsistent adoption of practices across different team members and difficulty monitoring adherence.

Relation to our thesis. The issues found by Conboy and Fitzgerald are partially because of the characteristics of XP, partially due to issues related to bad practices in methods adoption. In our exploratory studies we found similar kind of problems (cf. Chapter 3) and proposed respective solutions (cf. Chapter 4). Additionally, their study design bears some similarities to ours.

BOEHM AND TURNER (Boehm and Turner, 2003) devised an approach for selecting between agile and planned practices based on identifying situational dependencies. The approach considers five critical agility and plan-driven factors: team size, project criticality, requirement change rate, personnel experience, and chaos thriving organizational culture (cf. Figure 7.6). It also comprises a risk assessment scheme composed of three classes of risk items:

- Environmental risk: technology uncertainties, many stakeholders, and complex system of systems.

- Agile risks: scalability, simple design, personnel turnover, and lack of skilled people in agile methods.

- Plan-driven risks: rapid change, demand for fast results, emergent requirements, lack of skilled people in plan-driven methods.

Teams can use the approach to balance agile and plan-driven methods in a customized software development strategy. The authors claim their approach can help organizations and projects take advantage of agile and plan-driven methods' benefits while mitigating many of their drawbacks.

Relation to our thesis. Context modeling and decision support approach to select methods that best fit a team was also pursued by us (cf. Chapter 6). Boehm and Turner's focus was between agile and plan-drive development whilst our approach was solely on MBSE methods.

HUANG AND HAN (Huang and Han, 2006) proposes a decision support model for determining the priorities of the Capability Matu-

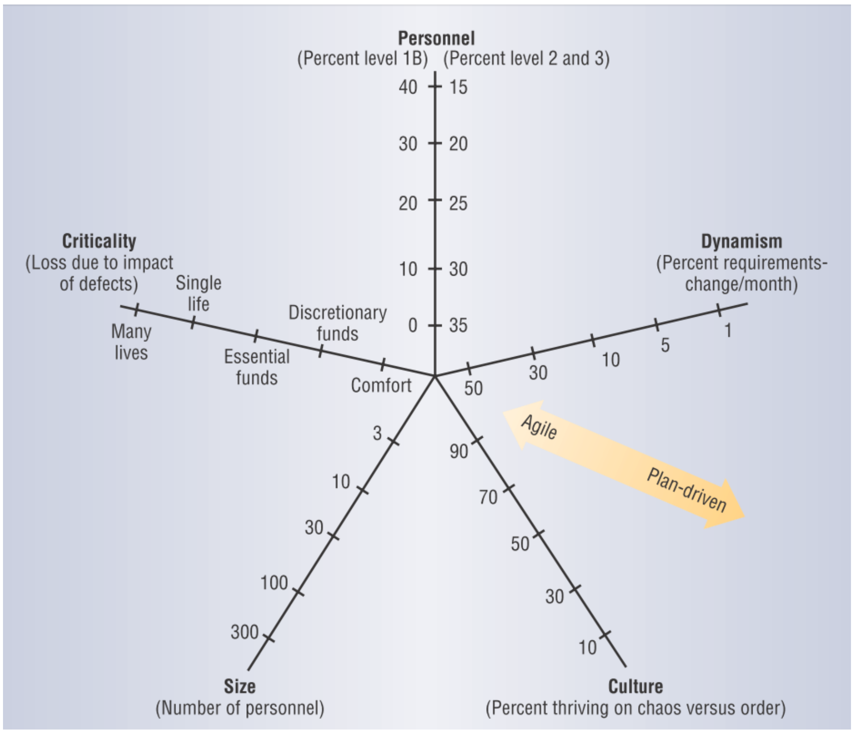


Figure 7.6: The five axes represent the factors used to distinguish between agile methods (toward the graph's center) and plan-driven methods (that appear towards the periphery) (Boehm and Turner, 2003).

rity Model Integration (CMMI) process areas. Their methods are based on the fourteen general system characteristics from the Function Point Analysis method (Garmus and Herron, 2001). In the original method, these characteristics are used for calibration (*i. e.*, value adjustment factor) when evaluating the overall complexity of a software application. The authors use the same scale and assessment concepts.

Relation to our thesis. In this study, the authors had the same goal as our contribution in Chapter 6, namely to provide decision support for selecting methods. Both methods use a maturity model to assess the current state and possible development paths, their work focusing on the CMMI framework while we aimed for the MBSE MM. Like us, they assessed the characteristics of the methods with experts, and they also have a small context assessment represented. However, our work focused on the adoption goal, while their work focused on the developed system characteristics.

7.5 BENEFITS MANAGEMENT

Benefits Management (or Benefits Realization Management) is a project management discipline aiming to close the gap between strategy planning and project execution by ensuring the implementation of the most valuable initiatives (Badewi, 2016; Breese et al., 2016; Serra and Kunc, 2015). It comprises structured processes to define and align project outcomes, costs, benefits, and business strategy. At the project planning phase, benefits identification and quantification can help identify a preferred investment option and deliver the best return. Such information is a building block for the justification of a project and becomes part of the business case, which is a significant part of our approach presented in Chapter 6. Established project management methodologies recognize the importance of benefits management (Frederiksz, Hedeman, and van Heemst, 2010; Sowden, 2011).

Several benefit modeling diagrams are described in the literature having elements representing benefits and their relation to decisions and solutions. These diagrams help to identify and display the benefits aligned with the expectations of the sponsor and stakeholders. They can be used to validate the project scope, identify associated benefits and cost areas of the project, and be used as a starting point in preparing project plans. In the following, we present three diagrams and compare them with our approach.

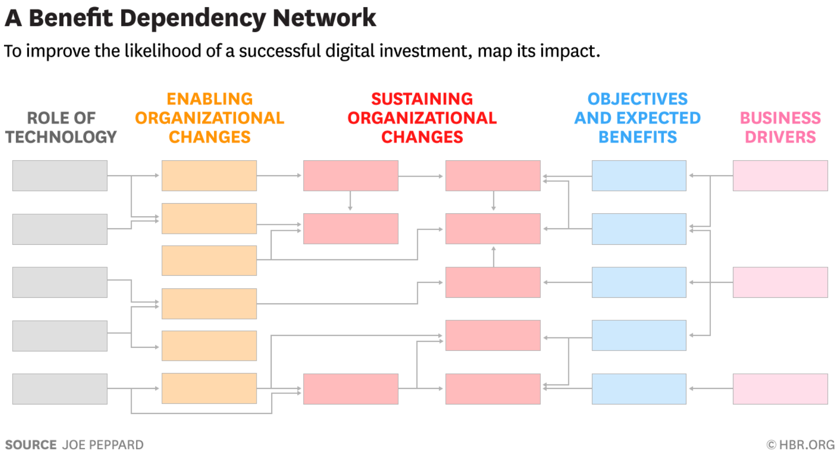


Figure 7.7: Benefits Dependency Network meta-model (Peppard, 2017).

7.5.1 *Benefit Dependency Network (BDN)*

The Benefits Dependency Network (BDN) (Peppard, Ward, and Daniel, 2007) was created to get the most out of digital investments (*e.g.*, information systems) in organizations. It helps to identify and map the changes required to deliver expected benefits and outcomes.

In this diagram, changes are categorized into two types: sustaining and enabling change. The former are permanent changes to working practices, processes, or relationships that will cause the benefits to be delivered (*e.g.*, new processes). The latter are one-off changes required for achieving the sustaining changes (*e.g.*, training and education in using a new system). The elements of this type of diagram are:

- **Business drivers:** High-level goals for undergoing change
- **Objectives:** The goals related to the Business drivers.
- **Expected benefits:** The harvested benefits of adopting change contribute to achieving the Objectives.
- **Sustaining organizational changes:** The change will stay after the change project is finished.
- **Enabling organizational changes:** Activities required to introduce the sustaining organizational changes (*e.g.*, training).
- **IS/IT enablers:** Tools (*e.g.*, information systems, technology) needed to support changes and achieve benefits.

The first three elements (Business drivers, Objectives, and Expected benefits) define the reasons to change. The Sustaining organizational changes explain the processes that require change, and the last two elements (Enabling organizational changes and IS/IT enablers) define how to achieve it.

The BDN is instantiated from right to left (cf. Figure 7.7). It starts at the business drivers and the objectives and expected benefits, and map the sustaining organizational changes (*i.e.,* required changes to structures, processes, work practices), and enabling organizational changes (*i.e.,* what is required to sustain those changes). Further, measures and responsibilities for benefits and changes are assigned and time scales established.

The approach's goal is to align investment objectives with critical business drivers, which helps organizations avoid digital investments to be steered by technology (left-hand side of the network) but driven by business demand (right-hand side of the network). The BDN was successfully used for streamlining patient administration in a hospital, implementing a customer-relationship-management (CRM) system in a financial services organization, rolling out a global enterprise-resource-planning (ERP) system for a pharmaceutical company, and promoting collaboration in a technology company (Peppard, 2017).

7.5.2 *Benefits Dependency Map (BDM)*

Benefit Dependency Map (BDM) (Bradley, 2010) is a goal-benefit modeling technique useful for relating project tasks and outputs to a business's strategic objectives. It shows business change necessary to achieve objectives through benefits and makes priorities clearer, thus helping planning. Benefits are the measurable addition of value (*i. e.,* End benefits) and are often dependent on other benefits (*i. e.,* Intermediate benefits) that must be realized first. Benefits are realized from Enablers to Business change, to Benefits, *i. e.,* from left to right. Enablers and Business changes leading to major benefits should be prioritized.

When read from left-to-right, the BDM shows reasons for pusuing change. In the opposite direction, the diagram shows how the benefits are to be achieved. An example of a BDM diagram can be seen in Figure 7.8. The elements belonging to the BDM diagram are better explained in the following.

- **Objective:** End goal of the initiative.

- **End benefit:** The benefits required to fulfill the goal.

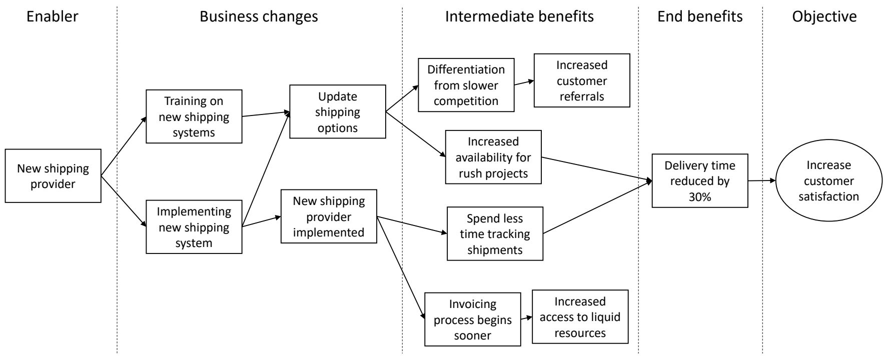


Figure 7.8: Benefits Dependency Map example.

- **Intermediate benefit:** Benefits that either contribute to the end benefit or put the company at an advantage. Intermediate benefits

- **Business changes/outputs:** Tasks (*e.g.*, develop training, implement systems or processes) and artifacts (*e.g.*, documents, project outputs) that generate benefits

- **Enabler:** The element (*e.g.*, systems, processes) required to achieve the goal. They might be either purchased or created from scratch.

7.5.3 *Results Chain Modeling (RC)*

Results Chain Modeling (Torpp, 1999) is a benefit modeling technique that shows the connections between activities, outcomes, and associated assumptions (cf. Figure 7.9). Through this model's building process, activities required to be performed and the respective outcomes to be achieved become explicit, and the realization of benefits becomes easier to manage. The following elements belong to this model:

- **Activity:** used to characterize the work that contributes to an outcome. It can be used to represent activities, projects, programs. The square form represents this element.

- **Outcome:** represents the result or outcome of an activity. There are two types of outcomes, intermediate ones and ultimate. A circle form represents this element.

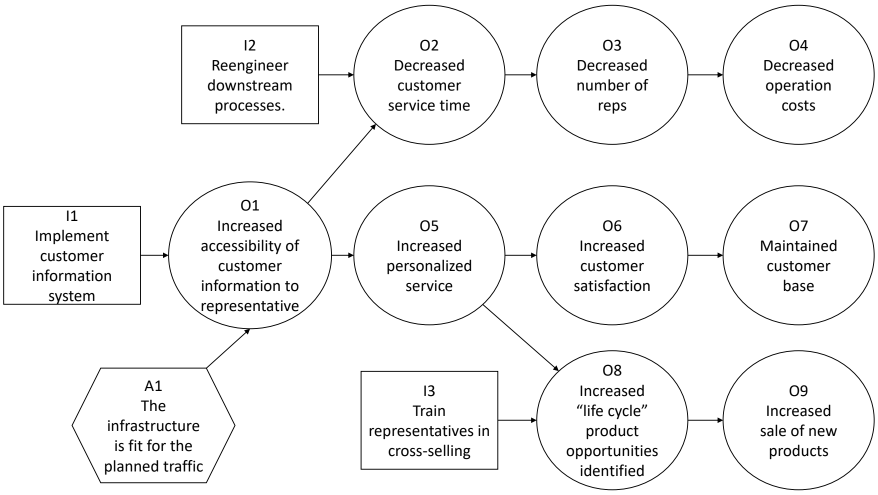


Figure 7.9: Results Chain Modeling example (adapted from (Torpp, 1999)).

- **Assumption:** describes any consideration required to achieve an outcome that is not in control. Among the benefit modeling techniques presented in this section, the RCM is the only one with such an element. A hexagon is used to represent this element.

Activities are described using an imperative verb (*e.g.*, document, identify, test). Outcomes are defined using the past tense of a verb (*e.g.*, created, reduced). Activities are connected to outcomes by arrows denoting their output. There are no connections between activities, and an outcome can be connected to other outcomes (*i.e.*, can generate different outcomes). The result chain model is read from left to right. The primary benefit of a project is the outcomes depicted on the right side of the model. Other outcomes are considered intermediate. Although dependency between elements is explicit, this model does not contemplate a timeline (*i.e.*, when the outcome is achieved or when an activity should be performed compared to other not directly connected activities).

7.5.4 *Comparison of benefits management diagram elements and the proposed approach*

The presented diagrams have similar elements, and they also share a resemblance with the model elements proposed in this thesis (cf. Chapter 6), which are compared in Table 7.2. These diagrams have elements representing benefits and their relation to decisions and solutions.

The relations are not one-to-one. For instance, our approach's equivalent of benefits relates to two elements in the BDM, namely End Benefit and Intermediate Benefits. The former, together with the Bonding objective, could be related to the Competitive priorities. The BDN requires three elements before benefits are delivered, the BDM needs two, our approach has only one (Candidate capabilities).

In contrast, Investment objectives from BDN and Bounding objectives from BDM are related to both the Adoption goal and Competitive priorities. The Outcome element from RC diagram relates to three other elements of our approach. Despite the small variety of elements, this diagram has an element type not present in the others diagrams, namely, the Assumption element, which provides a rationale for the chain of relations. The previously described GSN notation also has an Assumption element. In our approach, assumptions are implicitly considered when defining the relations weights between the elements.

These diagrams had a significant influence on our approach design. However, besides documenting the relations between elements, these diagrams do not provide quantitative decision support. Our approach differentiates itself for using competitive priorities as middleware between adoption goals and benefits, weighted relations, and a quantitative prioritization mechanism.

Table 7.2: Equivalence of elements from the approach proposed in this thesis (cf. Chapter 6) and the elements of Benefits Management diagrams.

Our approach	**BDN**	**BDM**	**RC**
MBSE adoption goal	Investment Objectives	Bounding objective	Outcome
Competitive priorities	Investment objectives	Bounding objective, End benefit	Outcome
Benefits	Benefits	End benefit, Intermediate benefit	Outcome
Candidate capabilities	Business changes, Enabling changes, IS/IT enablers	Business change, Enabler	Activity
–	–	–	Assumption

8

CONCLUSIONS AND OUTLOOK

In this chapter, we summarize the contributions of this thesis, discuss possible improvements of the presented work, and directions for further research.

8.1 SUMMARY

The work of this thesis was built around the problem statements *"We need to understand how to efficiently introduce new MBSE methods in development teams"*, and *"We need a method to provide decision support when selecting the MBSE methods that are most appropriate for each team considering respective contextual characteristics and adoption goal"* (cf. Section 1.2). We claimed the solutions proposed by this thesis and respective supporting evidence address the aforementioned problems. In the following, we conclude that the contributions provided in this thesis support these claims.

8.1.1 *Forces that drive or prevent Model-based Systems Engineering adoption in embedded systems industry*

In Chapter 3, we investigated the motivational forces that either propel or prevent MBSE adoption. Four research questions guided our efforts, and their answers are given in the following:

RQ1: WHAT ARE PERCEIVED FORCES THAT PREVENT MBSE ADOPTION IN THE INDUSTRY?

- **RQ1.1: What are habits and inertia that prevent MBSE adoption?** – Practitioners were not so sure about the benefits of adopting MBSE; thus, *Doubts about improvements* was the code most present in *inertia* category. Issues regarding tools such as the risk of *Incompatibility with existing tools*, and the perception that the available tools are still too immature (*i. e., Imature tooling*) were both the next two factors in this category preventing MBSE adoption.

- **RQ1.2: What are anxiety factors that prevent MBSE adoption?** – Concerns that MBSE method adoption investments would not pay off were the main anxiety factor (*i. e., ROI uncertainty*). The respondents were also worried that the MBSE way of developing systems could diminish the engineers' skills (*i. e., Skills of employees*). Like in inertia force, the tool was also a strong anxiety factor (*i. e., Tooling shortcomings*).

Uncertainty regarding benefits delivery was the starkest concern in both categories. This finding could be related either due to expectations misconception or fears steaming from reported or experienced unsuccessful adoption endeavors. Tool-related concerns were also prominent; non-seamless toolchains can increase effort due to information transfer or by requiring re-work with multiple tools. MBSE is very tool intensive, and this concern can come from a misconception of the maturity of the current tools.

RQ2: WHAT ARE PERCEIVED FORCES THAT FOSTER MBSE ADOPTION IN THE INDUSTRY?

- **RQ2.1: What are perceived issues that push the industry towards MBSE?** – A trend of current systems is the *Growing complexity*, and this is the main reason why teams are moving towards MBSE adoption. Are also highly relevant in this category the *Quality issues*, which relates to the aforementioned increase of complexity, and the *Time pressure* which is a reflection of fierce market competition.

- **RQ2.2: What MBSE benefits are perceived as most attractive?** – In the *push* category, the *Easier handling of complexity* that MBSE is able to provide is considered the most relevant. Additionally, practioners see value in MBSE due to *Early feedback on correctness*, and *Documentation support*.

The results show that system complexity issues are the main driver for practitioners to seek MBSE methods in both fostering forces categories.

Organizational change is never easy, especially when trying to introduce complex approaches such as MBSE. In Chapter 3, we looked for the reasons and factors that either prevent or foster companies from adopting MBSE. For this means, we created a forces framework for analyzing 20 interviews, which were coded in several discussion rounds. Based on our results, practitioners may challenge their decision processes and adoption strategies. Researchers may study our results and find evidence to quantify and detail the considerations of practitioners. We conclude that bad experiences and frustration about

MBSE adoption originate from false or too high expectations. Nevertheless, companies should not underestimate the necessary efforts for convincing employees and addressing their anxiety.

8.1.2 *Strategies and best practices for Model-based Systems Engineering adoption in embedded systems industry*

The complexity and pervasiveness MBSE creates challenges that, when not adequately addressed, can jeopardize its implementation in development teams. In Chapter 4, we identified strategies and best practices for MBSE adoption through a series of interviews. These practices were then classified into four big groups: piloting, knowledge building, tools and process, and management. Further, they were discussed, compared to related work, and summarized. These summaries were used to build a questionnaire in which each respondent could express their level of agreement with the best practice. Then, we identified which hindering forces (cf. Chapter 3) could be mitigated by the best practices using the interview verbatim quotes as evidence (cf. Appendix F).

Learning from others' experience and applying proven best practices can save effort and enhance success chances. In the long run, benefits outweigh the costs and hurdles; however, it is necessary to identify success factors and share best practices enabling efficient and effective MBSE adoption in the industry. Not surprisingly, most of the best practices address organizational aspects of the adoption, such as how to pilot an MBSE adoption initiative, how to transfer knowledge, and what role the management has to play.

8.1.3 *Method value ascription*

In Chapter 5, we proposed a model to describe the elements and respective relations that influence the method value ascription by a development team. The model considers associated costs and perceived benefits upon method adoption and helped us better understand the role of its elements, interplay, and gaps that still need to be addressed by future work (*i. e.,* costs). Further, we identified the relations to this thesis other contributions towards the model elements (cf. Figure 8.1).

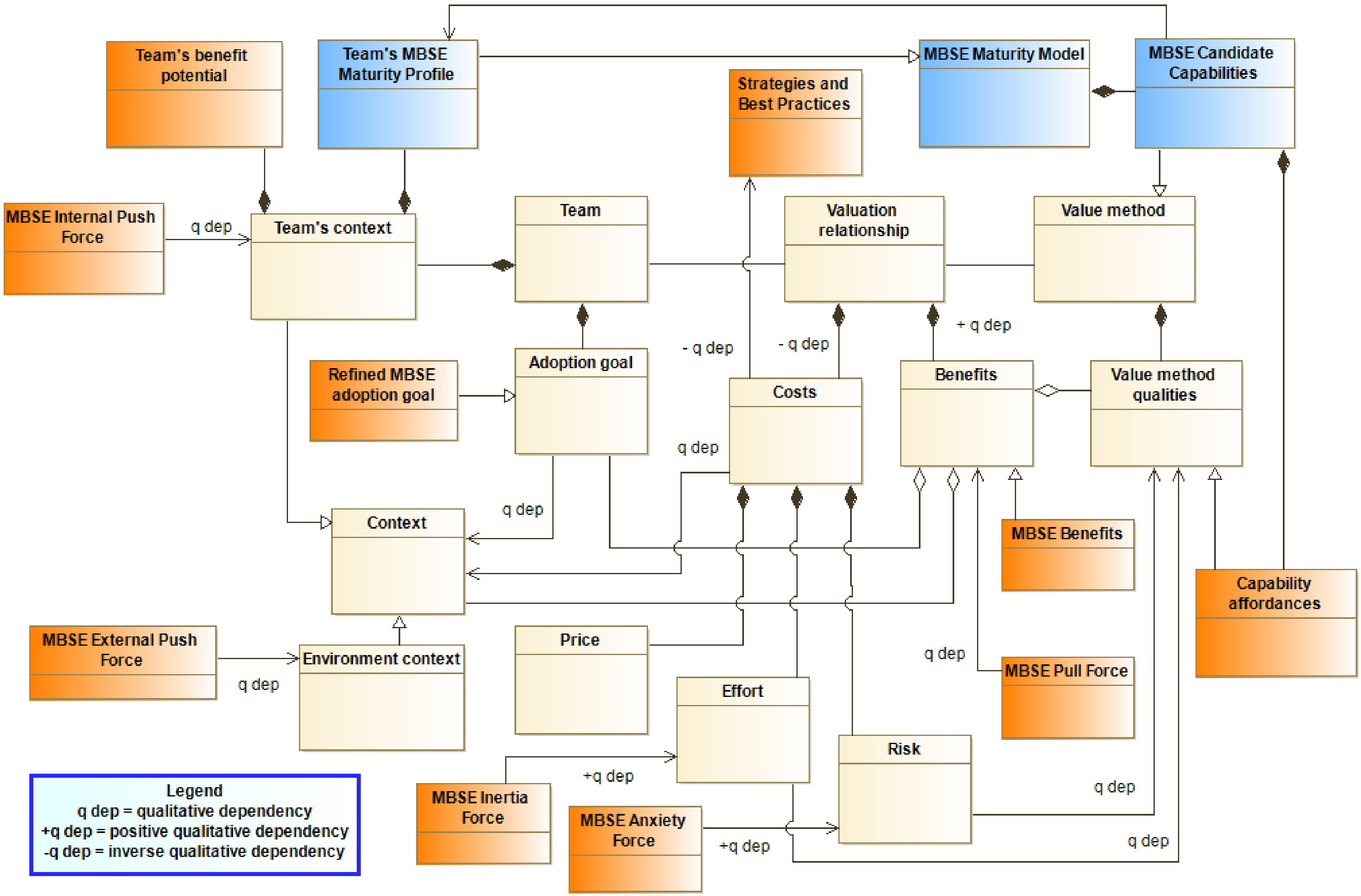


Figure 8.1: Relations between all contributions of this thesis.

8.1.4 *Goal-benefit decision calculus for Model-based Systems Engineering method selection*

Realizing abstract goals to concrete actions is never an easy task. This characteristic is especially true when the solution space is vast, and domain knowledge is required, which is the case of Model-based systems engineering method adoption. In Chapter 6, we proposed a decision calculus approach for development teams to trace the adoption goal to MBSE methods. The approach is composed of a goal-benefit model and an instantiation method. It can prioritize the solution space by breaking down the adoption goals into competitive priorities and benefits afforded by methods upon adoption. The instantiation method describes how to assign weights to the model element relations, representing their relevance towards neighbors. Once the model is instantiated, we can distribute points to the candidate methods that can be prioritized accordingly. The approach was applied in six case studies in development teams located in Brazil and Germany.

According to the case study participants' feedback, the proposed approach provides decision support, which helps to compose the business case and enhance confidence in decision-making. We also analyzed the model regarding its inputs and outputs, which showed that the model could prescribe diverse solutions to different cases. The contribution of this work is to provide structured thinking for decision-makers to deliver more reliable method adoption business plans.

8.2 OUTLOOK

This section discusses improvement opportunities for the presented work and proposes directions for future research.

8.2.1 *Investigate adoption forces further*

The results presented in Chapter 3 lays the stepping stones to understand further the dynamics of MBSE adoption. We could analyze the data to investigate correlations between roles and the identified categories and dependencies between the forces as future work. Additionally, the research community may create mechanisms to identify the forces within development teams more effectively and systematically, analyze how hindering forces can be further mitigated, and understand how to harvest fostering forces synergy.

8.2.2 *Horizontal extension of the decision model*

Many existing maturity models have the same shortcoming as the MBSE MM; they lack proper guidance in selecting and prioritizing actual improvement measures (Pöppelbuß and Röglinger, 2011). Although the approach presented in Chapter 6 was created having the MBSE MM as a reference framework, we believe that after some modifications, the same rationale could be used with other frameworks and even for other domains (*e. g.*, Software Product Management (van de Weerd, Bekkers, and Brinkkemper, 2010)). This possibility is especially true considering focus area maturity models. Hereby domain-specific benefits need to be elicited, and competitive priorities might need to be added or removed when adapting the current model.

8.2.3 *Vertical extension of the decision model*

As demonstrated in Chapter 5, a method's value is composed not only from the generated benefits upon its adoption but also the related costs, and the latter is not considered by the approach presented in this thesis (cf. Chapter 6). As future work, we would like to create a cost model that can be merged into the current approach, thus turning it into a goal-cost/benefit model. We have found studies that have started to define the cost of modeling (UML, for instance (Jolak et al., 2017)), and we might need to pursue a similar path if we want to define the cost for implementing and adopting the MBSE MM capabilities.

8.2.4 *Model context further*

The context plays a significant role in the whole topic of this thesis. However, modeling it is challenging; many authors have elicited contextual factors that could influence adopting new methods (Clarke and O'Connor, 2012). However, they were only successful at identifying possible context characteristics without properly relating them to decisions towards the methods, with very few exceptions (Bekkers et al., 2008). In our approach, we model the development team context in two ways, through the assessment of the *Maturity Profile* and the *Team's benefits potential* (cf. Chapter 6). We do not model the environment, although we recognize its influence on method value ascription (cf. Chapter 5). We see this as a research venue opportunity that is very incipient, not only regarding maturity models but in the software engineering field as a whole.

8.2.5 *Perform longitudinal studies*

In this thesis, we provide solutions for the planning phase of the MBSE method adoption (cf. Chapter 6). As future work, we could assess how the team perceives the recommended capabilities after they are adopted. Of course, confounding variables must be managed to avoid misperceptions, for instance, due to poor adoption process.

8.2.6 *Perform expert assessment in all MBSE MM capabilities.*

As part of the work performed in Chapter 6 we asked MBSE experts to evaluate the benefits generated by the 53 capabilities from the MBSE MM (cf. Section 6.4.1.1). At the time, we did not perform this taks for the whole MBSE MM to save the experts from the hassle of assessing unnecessary capabilities. However, this assessment is a fundamental part of the approach, and when non-assessed capabilities are selected to become candidates in future runs, their assessment will be required. The lack of this information adds an extra layer of work for someone employing the approach, which might hinder its application. Assessing the whole set of capabilities from the MBSE MM is needed in future work.

8.2.7 *Develop benefits monitoring metrics*

In order to verify if the team harvested the benefits from the implemented capabilities (cf. Chapter 6), metrics such as Key Performance Indicators (KPI) need to be defined. Such information is missing and is a beneficial research venue to be pursued. However, defining these metrics is not trivial since every team is different, and these characteristics also change over time. For instance, how a KPI for the benefit *"Better communication"* would look like?

A

MODEL-BASED SYSTEMS ENGINEERING MATURITY MODEL

The MBSE MM and its capabilities are described in Table A.1 and Table A.2.

Table A.1: The positioning of the capabilities according to maturity levels.

Engineering Function	Focus Area	0	1	2	3	4	5	6	7	8	9	10	11
Context Analysis	Operational Context (OPC)				A	B	C	D	E		F	G	
	Knowledge Context (KNC)			A	B	C		D					
Requirements	Scoping (SCO)		A	B	C	D	E						
	Goal Modeling (GOM)		A				B	C	D	E	F		
	Scenario Modeling (SCM)			A	B	C	D	E	F		G		
	Requirements Management (RM)				A			B	C	D	E	F	G
System Functions	System Function Modeling (SFM)		A			B		C					D
	System Function Specification (SFS)				A	B		C		D	E	F	
	Event Chain Modeling (ECM)					A		B	C			D	
	Mode Modeling (MM)					A		B		C		D	
Architecture	Logical Architecture Modeling (LAM)			A	B	C	D		E	F	G		
	Logical Component Modeling (LCM)				A		B	C	D		E		
Integration & Testing	System Behavior Testing (SBT)					A	B	C			D		
	System Quality Testing (SQT)					A				B	C	D	
Technical Implementation	Technical Architecture Modeling (TAM)			A	B	C		D	E	F			
	Technical Component Modeling (TCM)				A			B		C		D	

Table A.2: The MBSE MM capabilities description.

Context Analysis	A	B	C	D	E	F	G
Operational Context (OPC)	Actors and external systems *identified and documented*	Actors and external systems *modeled*	Context functions and data *identified and documented*	Context behavior *identified and documented*	Context functions and data *modeled*	Context behavior *modeled*	Context behavior *executable*
Knowledge Context (KNC)	Sources of possible requirements are *identified and documented*	Relationships between requirements and their sources *identified and documented*	Importance and volatility of the requirement sources *identified and documented*	Requirement sources and their relation to the requirements *modeled*	-	-	-
Requirements	A	B	C	D	E	F	G
Scoping (SCO)	System vision *identified and documented*	General solution concept *identified and documented*	User classes and their properties *identified and documented*	Purpose and explicit exclusions of systems *identified and documented*	Development constraints *identified and documented*	-	-

Goal Modeling (GOM)	Stakeholder goals regarding function and quality *identified and documented*	sub-goals that are necessary to achieve stakeholder goals *identified and documented*	Relationships between goals, system functions, and environmental resources *identified and documented*	Influence of system properties in stakeholder goals *identified and documented*	Stakeholder goals, sub-goals, and their relationships *modeled*	Quality issues in goal descriptions *automatically analyze*	-
Scenario Modeling (SCM)	Scenarios *identified and documented*	Scenario exception triggers *identified and documented*	Alternative scenarios in case of exceptions *identified and documented*	Dependencies between scenarios *identified and documented*	Scenarios *modeled*	Dependencies between scenarios *identified and documented*	Quality issues in scenarios *automatically analyzed*
Requirements Specification (RS)	Requirements *identified and documented*	Functional requirements *modeled*	Quality requirements *modeled*	Quality issues in functional requirements *automatically analyzed*	Quality issues in quality requirements *automatically analyzed*	Completeness and correctness of functional requirements[1] *automatically analyzed*	Completeness and correctness of quality requirements[1] *automatically analyzed*
System Functions	A	B	C	D	E	F	G
System Function Modeling (SFM)	System functions *identified and documented*	Dependencies between system functions *identified and documented*	Describe system functions and their dependencies *modeled*	The system functions together with a description of the context *simulated*	-	-	-

System Function Specification (SFS)	System functions[2] *identified and documented*	Function coverage[3] *identified and documented*	Describe a system function including its inputs and outputs *modeled*	Describe the behavior of the system functions *modeled*	The behavior description of the system functions *executable*	Function coverage[3] *automatically analyzed*	-
Event Chain Modeling (ECM)	Sub-functions that implement a system function *identified and documented*	Describe sub-functions including their inputs and outputs *modeled*	The behavior of the sub-functions *modeled*	The behavior of the system functions based on the network of their sub-functions *simulated*	-	-	-
Mode Modeling (MM)	Operating states of the system *identified and documented*	How operational states influence system functions *identified and documented*	Operational states and their transitions *modeled*	Whether transitions between operational states are correctly maintained by the system functions *automatically analyzed*	-	-	-
Architecture	A	B	C	D	E	F	G

Logical Architecture Modeling (LAM)	The decomposition of the system into logical components *identified and documented*	Alien logical components *identified and documented*	The relation between logical components and system functions that they realize *identified and documented*	Dependencies between logical components *identified and documented*	Logical components and their dependencies *modeled*	The behavior of logical component clusters to check for integration problems *automatically analyzed*	The logical components together with a description of the context *simulated*
Logical Component Modeling (LCM)	Logical components and their interfaces *identified and documented*	Logical components and their interfaces *modeled*	Logical components interface properties[4] *identified and documented*	Behavior of logical components *modeled*	The behavior description of logical components *executable*	-	-
Integration & Testing	A	B	C	D	E	F	G
System Behavior Testing (SBT)	Test cases *identified and documented*	The relationship between test cases and associated system models[5] *identified and documented*	Test cases for checking system behavior *modeled*	Test cases from system and/or context *automatically generated*	-	-	-

System Quality Testing (SQT)	Test cases to check system qualities[6] *identified and documented*	The relationship between test cases for checking qualities and associated system models [7] *identified and documented*	Describe test cases for checking system qualities *modeled*	Test cases for checking system qualities from system and/or context *automatically generated*	-	-	-
Technical Implementation	A	B	C	D	E	F	G
Technical Architecture Modeling (TAM)	Decomposition of the system into technical components *identified and documented*	Sensors and actuators that interface with the execution platform *identified and documented*	The relation between technical components and logical components or functions that they realize *identified and documented*	The mapping of logical input and outputs to inputs and outputs of technical components *identified and documented*	For each technical component, we whether the component can be assigned to exactly one engineering discipline[8] *identified and documented*	The technical components and their relations *modeled*	-
Technical Component Modeling (TCM)	Input and output signals for each technical component *identified and documented*	Describe the interface of technical components *modeled*	Describe the behavior of technical components *modeled*	The behavior description of technical components *executable*	-	-	-

---

[1] w.r.t. the operational context, goals, or described scenarios

[2] the name, a description, the function's purpose, involved actors, and pre- and post-conditions

[3] which requirements, scenarios, or goals the system function fulfills

[4] assumed properties about component inputs and guaranteed properties of component outputs

[5] (e.g., requirements, functions, or components)

[6] (i.e., non-functional requirements)

[7] (e.g., requirements, functions, or components)

[8] (e.g., mechanical, electrical, software)

B

INTERVIEW GUIDE

This appendix presents the interview guide used in the studies from Chapter 3 and Chapter 4.

1. Preamble
    - The aim of the interview is identifying the introduction needs and hurdles at interested companies.
    - All data concerning individuals will remain anonymous.
2. Information about the participants - Questions concerning the size of the company and the current state of MBSE projects.
    2.1. What do you develop in your company?
    2.2. How big are company and development teams?
    2.3. Who are the decision-makers for an MBSE introduction?
    2.4. Who is involved in decision making?
    2.5. What is your definition of MBSE?
    2.6. What is the current state of introduction of MBSE in the company?
3. Motivation and needs Questions about motivation and needs of introduction of MBSE.
    3.1. What makes MBSE interesting to you?
    3.2. How urgent is the MBSE demand due to the mentioned advantages?
    3.3. How does such an issue develop in the company?
    3.4. What are the costs?
4. Hurdles and overcoming Questions about introducing MBSE from today's view and possible approaches to overcome.
    4.1. What challenges do you see in an MBSE introduction?
    4.2. What influences of hurdles did you identify concerning the introduction of MBSE?
    4.3. What measures do you plan to overcome the hurdles?
    4.4. Where do you see the big challenges in your daily work?
5. Employee training Importance of training in the introduction process.

   5.1. What knowledge does the company have concerning MBSE?
   5.2. Which employee groups should be trained differentiated for MBSE?
   5.3. Where should be the focus of the training be placed?
   5.4. How many employees should be trained? (also in % of all users)
   5.5. What is the expected financial effort and expenditure of time for the training per employee?
   5.6. Is online training required?

6. Integration Integration of the methods and tools in existing landscapes.
   6.1. Do you want to migrate or translate data?
   6.2. Do interfaces exist with other internal systems?
   6.3. Expected changes in the process.
   6.4. Expected organizational changes.
   6.5. Expected financial effort and expenditure of time for integration.
7. Questions directed to system developers Specific questions to the developers who use or are supposed to use MBSE.
   7.1. What is the object of development you are working on?
   7.2. For which steps in the process are you responsible?
   7.3. How closely do you work with your customers?
   7.4. Are there special requirements for processes, methods or documentation?
   7.5. Which tools do you use?
   7.6. What do you get from outside? In which form?
8. Questions if you introduced MBSE. This chapter is omitted if MBSE has not yet been introduced. The aim is to determine the experiences in terms of "Lessons Learned" with the partial or complete introduction of MBSE in the.
   8.1. How do you use model-based development?
   8.2. Which tools are you already using for MBSE?
   8.3. What are the benefits to the development process?
   8.4. Were the expectations met? Which were not?
   8.5. Can the benefits be quantified?

   - 8.6. Were there unexpected annoyances in the use of methods and tools? How did you avoid them?
   - 8.7. Were there unexpected additional uses when using the methods and tools?
   - 8.8. How has your work changed with the model-based approaches?
   - 8.9. What were the hurdles during the introduction?
   - 8.10. How were the hurdles overcome?
   - 8.11. How did you build the necessary knowledge?
   - 8.12. Would you introduce the methods and tools, again?
   - 8.13. What new challenges do you see for the future?
- 9. Wrap-Up
   - 9.1. Is there anything else you can tell me about the subject?

C

QUESTIONNAIRE

In this chapter we present the questionnaire used in the survey from Chapter 4.

C.1 PREAMBLE

Survey on MBSE Adoption Best Practices

Dear MBSE practitioner,
My name is Tiago Amorim and I am a PhD candidate at TU-Berlin with the topic MBSE maturity improvement roadmap. Currently I am doing a research on MBSE adoption strategies and best practices in Embedded Systems Industry and I have collected some findings that I would like to validate with practitioners which have some experience with MBSE adoption.

I prepared a very FAST (i.e., 10 min. max.) survey (Google Forms) and I ask you to answer it and share it with relevant peers as much as possible.
Of course, the survey is anonymous and we will only report aggregated values.
The survey must be answered BEFORE 03.05, 23:59.

For further questions: buarquedeamorim@tu-berlin.de

Thank you for your collaboration.
Tiago Amorim

C.2 DEMOGRAPHIC QUESTIONS

1. What is your Industry Sector?
2. What is your role in your organization?
3. How big is your organization (employees)?
    - ○ 1-20
    - ○ 21-100
    - ○ 101-1000
    - ○ > 1000

4. Have you ever participated or observed some endeavor to introduce Model-based Engineering in a company?

C.3 MBSE ADOPTION BEST PRACTICES

How strongly do you agree that the following strategies help adopting Model-based System Engineering (MBSE) in companies? *(Strongly Disagree, Disagree, Agree, Strongly Agree)*

C.3.1 *Piloting*

- ▷ The organization should start adopting MBSE as part of a new project.
- ▷ The pilot project should create real value for the organization (i.e., no didactic project).
- ▷ The pilot project should have enough budget and time allocated to bear the overhead of adoption.
- ▷ No translation of old artifacts except for reusable artifacts.
- ▷ Start small in terms of project and team size in order to acquire some experience

C.3.2 *Tools and processes*

- ▷ Used tools should have open interfaces and homogeneous workflows.
- ▷ The whole tool chain should be replaced.
- ▷ Many tools might be required.
- ▷ All engineers should have access to the tools.
- ▷ Tool acquisition should be thoroughly planned.
- ▷ The new MBSE processes should be well documented to better understand what tool is required.

C.3.3 *Knowledge building*

- ▷ All engineers should get, at least, basic training in MBSE.
- ▷ Use training examples from the domain of the organization to ease the understanding.

- ▷ The context of an organization should affect the strategy to build knowledge of an organization.
- ▷ There should be a planned form of later evaluation to fill eventual gaps

C.3.4 *Management*

- ▷ Make the advantages of MBSE clear to the users.
- ▷ Have technically prepared people to support your engineers (i.e., not sales personnel).
- ▷ Bring everyone to adoption (i.e., avoid creating castes).
- ▷ If you have good engineers let them do the work for you, it is cheaper and they will engage more (i.e., empowering).
- ▷ Management should unify all employees towards adoption

C.4 ADOPTION BEST PRACTICES PRIORITIZATION

From the best practices you read in the previous section, which do you consider most important? Select up to 5 options.

- ☐ The pilot project should create real value for the organization (i.e., no didactic project).
- ☐ The pilot project should have enough budget and time allocated to bear the overhead of adoption.
- ☐ No translation of old artifacts except for reusable artifacts.
- ☐ Start small in terms of project and team size in order to acquire some experience
- ☐ The organization should start adopting MBSE as part of a new project.
- ☐ Used tools should have open interfaces and homogeneous workflows.
- ☐ The whole tool chain should be replaced.
- ☐ Many tools might are required.
- ☐ All engineers should have access to the tools.
- ☐ Tool acquisition should be thoroughly planned.
- ☐ The new MBSE processes should be well documented to better understand what tool is required.
- ☐ All engineers should get, at least, basic training in MBSE.

☐ Use training examples from the domain of the organization to ease the understanding.

☐ The context of an organization should affect the strategy to build knowledge of an organization.

☐ There should be a planned form of later evaluation to fill eventual gaps.

☐ Make the advantages of MBSE clear to the users.

☐ Have technically prepared people to support your engineers (i.e., not sales personnel).

☐ Bring everyone to adoption (i.e., avoid creating castes).

☐ If you have good engineers let them do the work for you, it is cheaper and they will engage more (i.e., empowering).

☐ Management should unify all employees towards adoption

D

CODE BOOK

This appendix presents the codebook used in Chapter 3 through a mind map diagram (cf. Figure D.1).

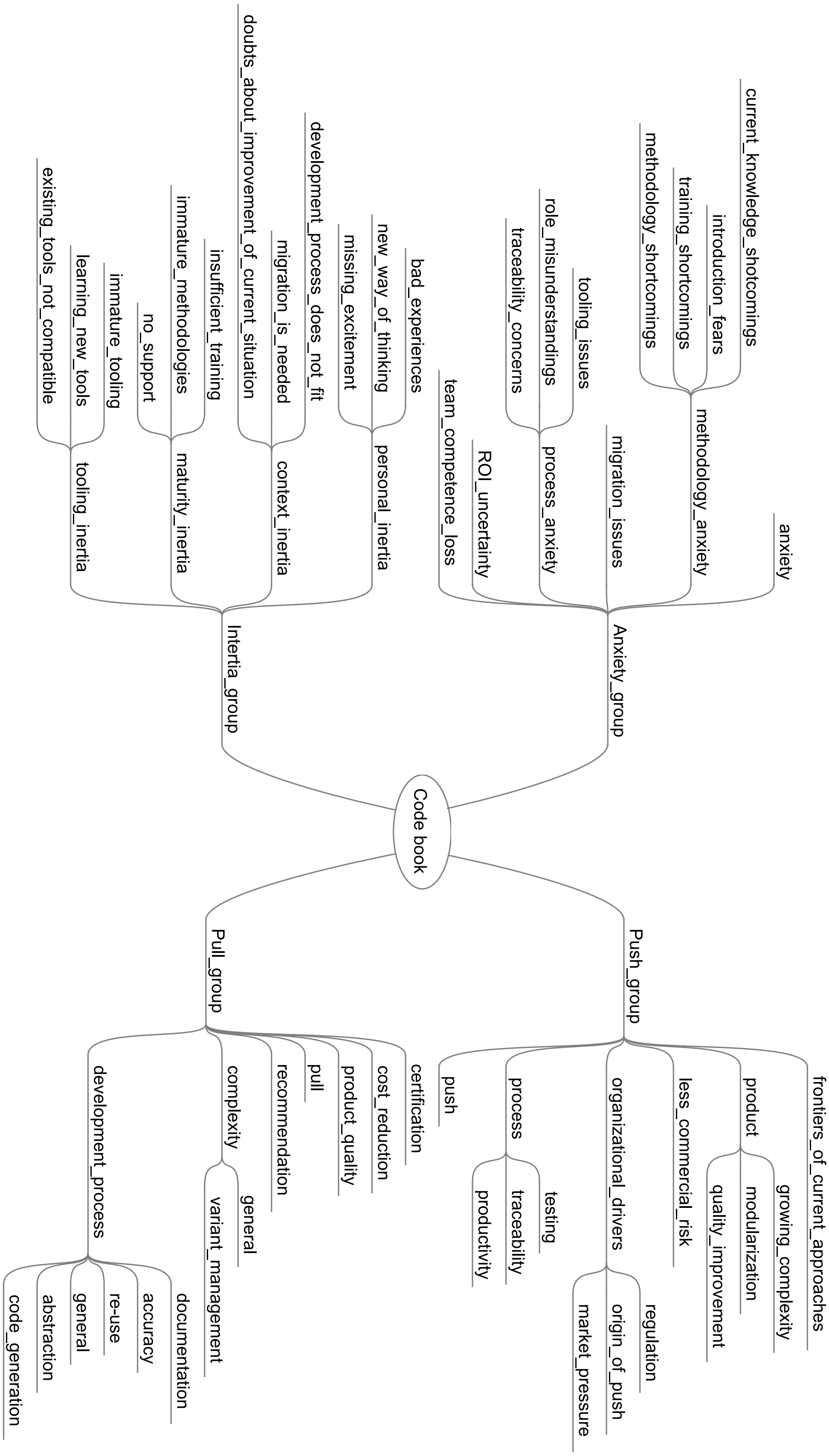


Figure D.1: Codebook mindmap.

E

REFINED MBSE ADOPTION GOAL AND TEAM'S BENEFIT POTENTIAL FROM CASE STUDIES OF CHAPTER 6

In this Appendix, we present the *Refined MBSE adoption goal* and *Team's Benefit Potential* from case studies of Chapter 6.

Table E.1: *Refined MBSE adoption goal* and *Team's Benefit Potential* of Case Study 1.

Goal: Enhance communication between software and system development teams.				
Competitive Priorities	Cost	Quality	Lead time	Scope
Goal Refinement	4,7%	11,9%	29,8%	53,6%
Benefits				
Easier reuse of existing artifacts	2	2	1	0
Better communication	0	2	1	2
Better understanding of problem domain	1	1	1	1
Better understanding of solution space	0	1	2	2
Better estimates	2	1	1	0
Less defects	2	2	1	0
Easier handling of complexity	1	2	1	1
Improved V&V	2	2	1	0
Improved quality of specification	1	1	1	1
Efficient certification	1	2	1	1
Sum:	14	16	11	8

Table E.2: *Refined MBSE adoption goal* and *Team's Benefit Potential* of Case Study 2.

Goal: Improve knowldge transfer.

Competitive Priorities	Cost	Quality	Lead time	Scope
Goal Refinement	3,7%	45,0%	35,3%	16,0%
Benefits				
Easier reuse of existing artifacts	1	2	2	0
Better communication	1	2	1	1
Better understanding of problem domain	2	2	2	2
Better understanding of solution space	1	2	2	2
Better estimates (cost, impact of req. changes)	2	1	2	2
Less defects	2	1	2	0
Easier handling of complexity	1	0	2	2
Improved V&V	1	2	2	0
Improved quality of specification	1	2	2	2
Efficient certification	0	0	0	0
Sum:	12	14	17	11

Table E.3: *Refined MBSE adoption goal* and *Team's Benefit Potential* of Case Study 3.

Goal1: Improve effort estimation.

Goal2: Reduce rework due to communication issues and scope definition.

Competitive Priorities	Cost	Quality	Lead time	Scope
Goal Refinement 1	5,4%	11,7%	68,0%	14,9%
Goal Refinement 2	3,6%	13,7%	50,3%	32,4%
Benefits				
Easier reuse of existing artifacts	1	2	2	0
Better communication	1	2	2	1
Better understanding of problem domain	1	2	2	0
Better understanding of solution space	1	2	2	1
Better estimates (cost, impact of req. changes)	1	2	2	0
Less defects	1	2	2	0
Easier handling of complexity	1	2	2	1
Improved V&V	1	2	1	1
Improved quality of specification	1	2	1	0
Efficient certification	0	0	0	0
Sum:	9	18	18	4

Table E.4: *Refined MBSE adoption goal* and *Team's Benefit Potential* of Case Study 4.

Goal: Automate system analysis.				
Competitive Priorities	Cost	Quality	Lead time	Scope
Goal Refinement	4,9%	62,3%	19,8%	13,0%
Benefits				
Easier reuse of existing artifacts	2	1	2	0
Better communication	1	2	2	0
Better understanding of problem domain	1	2	1	2
Better understanding of solution space	1	2	2	1
Better estimates (cost, impact of req. changes)	1	1	0	1
Less defects	2	2	1	2
Easier handling of complexity	2	2	2	1
Improved V&V	1	2	1	1
Improved quality of specification	1	2	1	1
Efficient certification	0	0	0	0
Sum:	12	16	12	9

Table E.5: *Refined MBSE adoption goal* and *Team's Benefit Potential* of Case Study 5.

Goal: Improve ability of the team.				
Competitive Priorities	Cost	Quality	Lead time	Scope
Goal Refinement	3,7%	18,3%	43,4%	34,6%
Benefits				
Easier reuse of existing artifacts	2	1	2	1
Better communication	0	1	1	1
Better understanding of problem domain	2	2	2	2
Better understanding of solution space	2	2	2	2
Better estimates (cost, impact of req. changes)	0	0	0	0
Less defects	2	1	2	2
Easier handling of complexity	0	1	1	1
Improved V&V	2	2	2	2
Improved quality of specification	2	2	2	2
Efficient certification	0	0	0	0
Sum:	12	12	14	13

Table E.6: *Refined MBSE adoption goal* and *Team's Benefit Potential* of Case Study 6.

Goal: Automate artifact treatment activities.

Competitive Priorities	Cost	Quality	Lead time	Scope
Goal Refinement	3,0%	61,0%	9,8%	26,2%
Benefits				
Easier reuse of existing artifacts	1	2	1	2
Better communication	0	1	1	2
Better understanding of problem domain	1	2	2	2
Better understanding of solution space	1	2	2	2
Better estimates (cost, impact of req. changes)	1	2	2	2
Less defects	1	2	2	1
Easier handling of complexity	1	2	2	2
Improved V&V	1	2	1	2
Improved quality of specification	1	2	2	2
Efficient certification	1	2	1	2
Sum:	9	19	16	19

F

EVIDENCE FOR THE RELATIONS BETWEEN MBSE ADOPTION HINDERING FORCES AND BEST PRACTICES

In this appendix, we present the participants' interview quotes which serve as evidence for how best practices can be used to mitigate MBSE adoption hindering forces, as described in Section 4.4. The information is organized considering each best practice; then, for each participant quote related to the best practice, the following code is used: BPXX_PYY, where XX denotes the best practice and YY indicates the participant number. Then, quotes of the participants related to hindering forces are presented using the code GROUP:FORCE, where GROUP can be either Anxiety or Inertia, and FORCE is the force's name. For instance, BP01_P01 indicates a quote related to Best Practice 01 and interview participant 01. In the following, quotes are listed.

BP01: The organization should start adopting MBSE with new projects.

BP01_P01*"New start in new project preferred"*, *"Existing projects remained in old surroundings and were gradually archived"*

- **Anxiety:Methodology shortcomings** *"There is a lack of a consistent methodology - thus uncertainties" / "The demarcation of the method against the processes is rarely solved"*
- **Anxiety:Role misunderstanding** *"There is a lack of understanding of one's own role in the overall context. The people in training do not understand the incorporation of their role"*

BP01_P03*"Set up own project for method development instead of accompanying from series development"*

- **Inertia:Doubts about improvements** - *"Low acceptance in the meantime"*
- **Inertia:Imature tooling / Inertia:Learning new tools** - *"Users first chose Excel before because of startup problems"*
- **Inertia:Incompatibility with existing tools** - *"Tedious development and testing of methodology; Efforts for specific tool extensions through plugins"*
- **Inertia:Imature methodologies / Inertia:Imature tooling** - *"Method problems (and also some general tool problems) were noted, but could not be solved (despite support hotline)"*

BP01_P11*"Do not recycle the past"* - **Anxiety:Skill of employees** - *"Decision not yet taken to switch to new plan, not sure if that works"*

BP01_P13*"Remove [team] from existing organization, set up your own project, the rest continues as before. Within a few years, new product generation has higher market share.", "Backwards compatibility, so that only changes need to be redesigned (includes not only code, but also specs, tests, test infrastructure)"*

- **Anxiety:Skill of employees / Anxiety:Tooling shortcomings / Anxiety:Role misunderstandings / Anxiety:Team competence loss** - *"Usually, developers are in several projects. But: two methods, different tools => is not good"*
- **Inertia:Migration is needed** - *"Legacy problem is real hurdle: old approach must continue to be used and maintained"*

BP03: The pilot project should have enough budget and time allocated to bear the overhead of adoption.

BP03_P11*"The first projects must be able to bear the burden."* - **Anxiety:ROI uncertainty** - *"It is a mistake to believe that an advantage will be felt in the first project"*

BP03_P17*"Business case because of investment hurdle in the beginning"*

- **Inertia:Development process does not fit** - *"Established and certified processes. Change costs money".*
- **Anxiety:ROI uncertainty** - *"Can this be recovered through savings?"*

BP05: Start small in terms of project and team size in order to acquire some experience.

BP05_P01*"Often pilot installation to gain experience, then expansion companies-wide"*

- **Inertia:Doubts about Improvements** - *"Developer with Simulink know-how (just one example): Understanding, but doubting the meaning ('as before, just a bit more abstract - where is the benefit?')"*
- **Anxiety:Methodology shortcomings** - *"There is a lack of a consistent methodology - thus uncertainties" / "The demarcation of the method against the processes is rarely solved"*

BP05_P02*"It is advisable to start with a partial introduction first, later through a consistent MBE. - needed breath" - "If necessary implementation strategy with 3-4 months observation of what the developer does, and then decide how to improve the process with MBE"*

- **Inertia:Immature tooling** - *"Insufficient tools made the introduction difficult or impossible (instability or bad GUI)"*
- **Inertia:Incompatibility with existing tools** - *"Lack of integration with existing tools has made the introduction difficult or stopped development process does not fit - The process of avionics is so slow that changes can only be introduced step by step"*
- **Anxiety:ROI uncertainty** - *"Often the hurdles were not overcome, because the conversion would have become much more expensive for the company"*

BP05_P05*"Introduction as 'submarine project' by dedicated developer, then application in own department, now overarching roll-out."* - **Inertia:Immature methododlogies** - *"Resistances due to misunderstandings / overlaps with existing MBW work in Simulink"*

BP05_P06*"Start in some areas that you can better oversee [...] Best to start in individual areas according to company guidelines"* - **Anxiety:Skill of employees / Anxiety:Migration issues** - *"Not all at once, people do not want that either"*

BP05_P19*"Only 1-2 people, then the rest of the team"*

- **Anxiety:ROI uncertainty** - *"Doubts about usability, use, flexibility"*
- **Anxiety:ROI uncertainty / Anxiety:Methodology shortcomings** - *"Doubts about maturity, scalability"*
- **Anxiety:Skill of employees / Anxiety:Methodology shortcomings** - *"Maintainability: Stability of the language, uncertainty over time and between colleagues"*
- **Anxiety:Methodology shortcomings** - *"Maintainability open, not clear if someone can later use the formal language"*

BP07: All engineers should have access to the tools.

BP07_P11*"The worker level must also get the tools"* - **Anxiety:Role misunderstandings / Anxiety:Team competence loss** - *"Avoid division in the team (worker / modeler)"*

BP08: Tool acquisition is very costly therefore should be thoroughly planned.

BP08_P01*"Development of a holistic approach/methodology"* - **Inertia:Learning new tools / Inertia: Missing excitement** - *"Complex tools with a lack of intuitive operation"*

BP08_P09*"Develop an adoption strategy"*

- **Anxiety:ROI uncertainty** - *"Introduction costs (tools, training)"*

- **Anxiety:Tooling shortcomings** - *"Tools (Cost, maturity, 'lock-in' fears, integrability, qualification)"/ "But still trouble with tools: operation incomprehensible, can do more than necessary"*

BP09: Have the new MBSE processes well documented so you better understand what tool you will need.

BP09_P01*"If the standard process is well documented, the MBE implementation will work easier."* - **Anxiety:Methodology shortcomings** - *"There is a lack of a consistent methodology - thus uncertainties" / "The demarcation of the method against the processes is rarely solved"*

BP09_P08*"First the process was set, then the tool decision"*

- **Inertia:Doubts about improvements** - *"Introduction costs (tools, training) - It takes a great emergency to justify the cost of implementing Doors (including the cost of creating the information model)"*
- **Anxiety:Tooling shortcomings** - *"I need tools to compare my models with reality"*

BP10_P17*"Tool support for automation of work"* - **Anxiety:ROI uncertainty** - *"Can this be recovered through savings?"*

BP10: All engineers should get, at least, basic training in MBSE.

BP10_P01*"Training, training, training!"*

- **Inertia:Insufficient training** - *"Inadequate training"*
- **Inertia:Immature methodology** - *"Unclear terminology"*
- **Inertia:Missing excitement** - *"Complex tools with a lack of intuitive operation"*
- **Anxiety:Role misunderstandings** - *"The people in training do not understand the incorporation of their role"*

BP10_P06*"Basic training for all users of MBSE"* - **Inertia:Immature tooling / Inertia:Learning new tools** - *"EA not user friendly. Issues distributed over many menus, you always have to search."*

BP10_P11*"Broad basic training of all employees - Everyone should have the same understanding"*

- **Inertia:Changing the way of thinking** - *"For many MA something completely new [...] is outside your comfort zone"*
- **Anxiety:Role misunderstandings / Anxiety:Team competence loss** - *"Avoid division in the team (worker / modeler)"*

BP11: Using examples that are familiar to the domain of the organization eases the understanding. Model some existing artifacts for using as examples.

BP11_P02"*[Training team] has modeled examples of clients and then presented them [sic] to the client (we'll show you how we do it)*"

- **Inertia:Doubts about improvements** - "*The user of the MBE (developer) must be convinced that the added value must be clear to him, then it's easier*"
- **Anxiety:ROI uncertainty** - "*'Coaching on the job' at launch is important, but it costs a lot*"

BP11_P11"*Examples from similar industry help*" - **Inertia:Changing way of thinking** - "*For many MA something completely new [...] is outside your comfort zone*"

BP12: Many strategies can be used to build knowledge of an organization, the context should be taken into consideration.

BP12_P05 "*Model should be reference. To do this, connect other tools (e.g., importing Doors for initial filling, then only update exports back to Doors, generated code frames force interface fidelity)*" - **Inertia:Insufficient training** - "*In one department: Low willingness to abstraction and the need to quickly get technical discussions*"

BP12_P08"*Face-to-Face Seminar with exercises*" "*Per tool training e.g. Doors*" - **Anxiety:Large training efforts** - "*Training necessary: How do I bring my engineers to the same level as the experts?*"

BP13: There should be a planned form of later evaluation to fill eventual gaps.

BP13_P01"*Continuous explanation of the methodology*" - **Inertia:Insufficient training** - "*Inadequate training*"

BP13_P05"*Accompanying technical monitoring of work results; is not happening systematically.*" - **Inertia:Insufficient training** - "*In one department: Low willingness to abstraction and the need to quickly get technical discussions*"

BP14: Make the advantages of MBSE clear.

BP14_P18"*Good examples show: project for new methods tools, then show that it works (utility and acceptance).*" - **Inertia:Doubts about improvements / Inertia:Learning new tools** "*As with any change a certain denial attitude. Especially with older developers who work textually - they are diffi-*

cult to convince of a new tool. The benefits should already be felt directly for them"

BP14_P13*"Every few years new product generation, e.g. because of Platform changes through technical advances (multicore ...) or new system approaches that affect many functions. Then there is a break in the system concepts, many have to be taken in hand, because you can introduce new methods." "Representation as a means of unique selling proposition. Works, but each developer must be convinced individually."*

- **Inertia:Doubts about improvements** - *"Problem still not felt, then little action readiness./Justification Benefits vs. Effort is difficult."*
- **Inertia:Changing the way of thinking** - *"Difficult to tell successful and experienced people: 'That's how you have to do it'. Psychological problem."*
- **Anxiety:ROI uncertainty/migration issues** - *"Architectural changes are heavy. Then high benefits must be demonstrated"*

BP14_P16*"Advantages of simulability were made visible"* - **Anxiety:ROI uncertainty / Anxiety:Skill of employees** - *"Benefit of the model based not quite clear, costs a lot. Alternatives unclear."*

BP16: Bring everyone to adoption (i.e., avoid creating castes).

BP16_P11*"Avoid living apart, everyone has to go!!!"* - **Anxiety:Role misunderstandings / Anxiety:Team competence loss** - *"Avoid division in the team (worker / modeler)"*

BP17: If you have good engineers let them do the work for you, it is cheaper, and they will engage more (i.e., empowering).

BP17_P08*"Information model developed itself"* - **Inertia:Insufficient training** - *"Standardization of the language between the MA necessary"*

BIBLIOGRAPHY

Afzal, W., R. Torkar, R. Feldt, and T. Gorschek (2013). "Prediction of Faults-Slip-Through in Large Software Projects: An Empirical Evaluation." In: *Software Quality Journal* 22.

Ågerfalk, P. J. and K. Wistrand (2003). "Systems Development Method Rationale: A Conceptual Framework for Analysis." In: *Proceedings of the 5th International Conference on Enterprise Information Systems (ICEIS'03)*.

Ameller, D. (2009). "Considering Non-Functional Requirements in Model-Driven Engineering." MA thesis. Llenguatges i Sistemes Informàtics (LSI).

Amorim, T., H. Martin, Z. Ma, C. Schmittner, D. Schneider, G. Macher, B. Winkler, M. Krammer, and C. Kreiner (2017). "Systematic Pattern Approach for Safety and Security Co-engineering in the Automotive Domain." In: *Proceedings of the 36th International Conference on Computer Safety, Reliability, and Security (SAFECOMP'17)*.

Amorim, T., A. Vogelsang, F. Pudlitz, P. Gersing, and J. Philipps (2019). "Strategies and Best Practices for Model-based Systems Engineering Adoption in Embedded Systems Industry." In: *Proceedings of the 41st ACM/IEEE International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP'19)*.

– (2020). "Strategies and Best Practices for MBSE Adoption in Embedded Systems Industry." In: *Proceedings of the Software Engineering 2020, Fachtagung des GI-Fachbereichs Softwaretechnik*.

Aranda, J., D. Damian, and A. Borici (2012). "Transition to model-driven engineering." In: *Proceedings of the 15th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS'12)*.

Asan, E., O. Albrecht, and S. Bilgen (2014). "Handling Complexity in System of Systems Projects – Lessons Learned from MBSE Efforts in Border Security Projects." In: *Complex Systems Design & Management*.

Baars, T., F. Mijnhardt, K. Vlaanderen, and M. Spruit (2016). "An analytics approach to adaptive maturity models using organizational characteristics." In: *Decision Analytics* 3.

Badewi, A. (2016). "The impact of project management (PM) and benefits management (BM) practices on project success: Towards developing a project benefits governance framework." In: *International Journal of Project Management* 34.4.

Basili, V. R. (1992). *Software modeling and measurement: the Goal/Question/Metric paradigm*. Tech. rep. University of Maryland.

Basili, V. R., G. Caldiera, and H. D. Rombach (1994). "The Goal Question Metric Approach." In: *Encyclopedia of Software Engineering*. Wiley.

Basili, V., J. Heidrich, M. Lindvall, J. Munch, M. Regardie, and A. Trendowicz (2007). "GQM+Strategies – Aligning Business Strategies with Software Measurement." In: *Proceedings of the 1st International Symposium on Empirical Software Engineering and Measurement (ESEM'07).*

Batejat, F. (2019). *Build a car with graphs*. Graph Tour 2019 Volvo Cars. URL: `https://neo4j.com/videos/graph-tour-2019-volvo-cars/` (visited on 05/22/2021).

Bauer, V. and A. Vetrò (2016). "Comparing reuse practices in two large software-producing companies." In: *Journal of Systems and Software* 117.

Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. Addison-Wesley Longman Publishing Co., Inc.

Beck, K. and C. Andres (2004). *Extreme Programming Explained: Embrace Change (2nd Edition)*. Addison-Wesley Professional.

Becker, J., M. Rosemann, and C. von Uthmann (2000). "Guidelines of Business Process Modeling." In: *Business Process Management: Models, Techniques, and Empirical Studies*. Springer Berlin Heidelberg.

Bekkers, W. and M. Spruit (2010). "The Situational Assessment Method put to the test: Improvements based on case studies." In: *Proceedings of the 4th International Workshop on Software Product Management (IWSPM'10).*

Bekkers, W., I. v. d. Weerd, S. Brinkkemper, and A. Mahieu (2008). "The Influence of Situational Factors in Software Product Management: An Empirical Study." In: *Proceedings of the 2nd International Workshop on Software Product Management (IWSPM'08).*

Bekkers, W., M. Spruit, I. D. Weerd, R. V. Vliet, and A. Mahieu (2010). "A Situational Assessment Method for Software Product Management." In: *Proceeding of the European Conference on Information Systems (ECIS'10).*

Bennett, P. D. and P. F. Anderson (1995). *Dictionary of Marketing Terms.* NTC Business Books.

Bharadwaj, A. S. (2000). "A Resource-Based Perspective on Information Technology Capability and Firm Performance: An Empirical Investigation." In: *MIS Quarterly* 24.1.

Boehm, B. W., T. E. Gray, and T. Seewaldt (1984). "Prototyping vs. Specifying: A Multi-project Experiment." In: *Proceedings of the 7th International Conference on Software Engineering (ICSE'84).*

Boehm, B. and R. Turner (2003). "Using risk to balance agile and plan-driven methods." In: *Computer* 36.6.

Böhm, W., M. Junker, A. Vogelsang, S. Teufl, R. Pinger, and K. Rahn (2014). "A Formal Systems Engineering Approach in Practice: An Experience Report." In: *Proceedings of the 1st International Workshop on Software Engineering Research and Industrial Practices (SER&IPs'14).*

Boll, A., F. Brokhausen, T. Amorim, T. Kehrer, and A. Vogelsang (2021). "Characteristics, Potentials, and Limitations of Open Source Simulink Projects for Empirical Research." In: *Software and Systems Modeling.*

Bone, M. and R. Cloutier (2010). "The current state of model based systems engineering: Results from the OMG SysML request for information 2009." In: *Proceedings of the 8th Conference on Systems Engineering Research (CSER'10).*

Booch, G., J. Rumbaugh, and I. Jacobson (2005). *Unified Modeling Language User Guide, The (2nd Edition) (Addison-Wesley Object Technology Series).* Addison-Wesley Professional.

Bradley, Gerald (2010). *Fundamentals of Benefit Realization.* Tso.

Breese, R., S. Jenner, C. Serra, J. Thorp, and A. Badewi (2016). *A unified view of benefits management/benefits realization management to be integrated into PMI standards*. Tech. rep. Sheffield Hallam University.

Brinkkemper, S. (1996). "Method engineering: engineering of information systems development methods and tools." In: *Information and Software Technology* 38.4.

Brinkkemper, S., M. Saeki, and F. Harmsen (2001). "A Method Engineering Language for the Description of Systems Development Methods." In: *Proceedings of the 23rd International Conference on Advanced Information Systems Engineering (CAiSE'01)*.

Broy, M. (2006). "Challenges in automotive software engineering." In: *Proceedings of the 28th International Conference on Software Engineering (ICSE'06)*.

Broy, M., W. Damm, S. Henkler, K. Pohl, A. Vogelsang, and T. Weyer (2012). "Introduction to the SPES Modeling Framework." In: *Model-Based Engineering of Embedded Systems*. Springer Berlin Heidelberg.

Bryman, A. (2015). *Social research methods*. Oxford university press.

Bubenko, J. A. and M. Kirikova (1999). "Improving the Quality of Requirements Specifications by Enterprise Modelling." In: *Perspectives on Business Modelling: Understanding and Changing Organisations*. Springer Berlin Heidelberg.

Cambridge University Press (2020). *Cambridge Dictionary*. URL: `https://dictionary.cambridge.org/dictionary/` (visited on 10/20/2020).

Carroll, E. R. and R. J. Malins (2016). *Systematic Literature Review: How is Model-Based Systems Engineering Justified?* Tech. rep. Sandia National Laboratories.

Chaudron, M. R., W. Heijstek, and A. Nugroho (2012). "How Effective is UML Modeling?" In: *Software and Systems Modeling* 11.

Clarke, P. and R. V. O'Connor (May 2012). "The Situational Factors That Affect the Software Development Process: Towards a Comprehensive Reference Framework." In: *Information and Software Technology* 54.5.

Conboy, K. and B. Fitzgerald (2010). “Method and Developer Characteristics for Effective Agile Method Tailoring: A Study of XP Expert Opinion.” In: *ACM Transactions on Software Engineering and Methodology* 20.1.

Conner, D. R. (1993). *Managing at the Speed of Change*. Random House.

Copi, I.M. (2003). *Essentials of Logic*. Pearson Prentice Hall.

Dabkowski, M., J. Estrada, B. Reidy, and R. Valerdi (2013). “Network Science Enabled Cost Estimation in Support of MBSE.” In: *Procedia Computer Science* 16.

Dardenne, A., A. van Lamsweerde, and S. Fickas (1993). “Goal-directed requirements acquisition.” In: *Science of Computer Programming* 20.1.

DeCuir-Gunby, J., P. Marshall, and A. Mcculloch (May 2011). “Developing and Using a Codebook for the Analysis of Interview Data: An Example from a Professional Development Research Project.” In: *Field Methods Journal* 23.

Denzin, N.K. (2006). *Sociological Methods: A Sourcebook*. Methodological perspectives. Aldine Transaction.

Diebold, P., A. Vetrò, and D. M. Fernández (2015). “An Exploratory Study on Technology Transfer in Software Engineering.” In: *Proceedings of the 9th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM’15).*

Dowson, M. (1997). “The Ariane 5 Software Failure.” In: *ACM SIGSOFT Software Engineering Notes* 22.2.

Dresch, A., D. P. Lacerda, and J. A. V. Antunes (2015). “Design Science Research.” In: *Design Science Research: A Method for Science and Technology Advancement*. Springer International Publishing.

Foo, G. and D. J. Friedman (1992). “Variability and Capability: The Foundation of Competitive Operations Performance.” In: *AT&T Technical Journal* 71.4.

Fosse, E. and T. Bayer (2016). *Model-based Systems Engineering MBSE 101*. Incose International Workshop. URL: `https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:2016_iw-mbse_101.pdf` (visited on 05/22/2021).

Franco, L. A. and G. Montibeller (2011). “Problem Structuring for Multicriteria Decision Analysis Interventions.” In: *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc.

Frederiksz, H., B. Hedeman, and G. V. van Heemst (2010). *Project Management Based on PRINCE2 2009 Edition*. Van Haren Publishing.

Garmus, D. and D. Herron (2001). *Function Point Analysis: Measurement Practices for Successful Software Projects*. Addison-Wesley Longman Publishing Co., Inc.

Hammer, M. (2007). “The Process Audit.” In: *Harvard Business Review*.

Harvey, D. and T. Liddy (2014). “Architecting the Problem Space – Model-Based User Needs Analysis.” In: *Proceedings of the 4th Asia Pacific Conference on Research in Industrial and Systems Engineering (APCORISE’14)*.

Hayes, R. H. and S. C. Weelwright (1984). *Restoring Our Competitive Edge: Competing Through Manufacturing*. John Wiley & Sons, Inc.

Helle, P. (2012). “Automatic SysML-based Safety Analysis.” In: *Proceedings of the 5th International Workshop on Model Based Architecting and Construction of Embedded Systems*.

Henderson-Sellers, B. and J. Ralyté (2010). “Situational Method Engineering: State-of-the-Art Review.” In: *Journal of Universal Computer Science* 16.3.

Henderson-Sellers, B., J. Ralyte, P. J. Agerfalk, and M. Rossi (2014). *Situational Method Engineering*. Springer Publishing Company Inc.

Hoffmann, H. (2012). “Deploying model-based systems engineering with IBM Rational; solutions for systems and software engineering.” In: *Proceedings of the 31st IEEE/AIAA Digital Avionics Systems Conference (DASC’12)*.

Hohl, P., J. M, K. Schneider, and M. Stupperich (2016). “Forces that prevent agile adoption in the automotive domain.” In: *Proceedings of the 17th International Conference of Product-Focused Software Process Improvement (PROFES’16)*.

Horkoff, J. and E. K. Yu (2011). “Analyzing goal models: different approaches and how to choose among them.” In: *Proceedings of the ACM Symposium on Applied Computing (SAC’11)*.

Horkoff, J., D. Barone, L. Jiang, E. Yu, D. Amyot, A. Borgida, and J. Mylopoulos (2014). "Strategic Business Modeling: Representation and Reasoning." In: *Software and Systems Modeling* 13.

Huang, S. and W. Han (2006). "Selection priority of process areas based on CMMI continuous representation." In: *Information & Management* 43.3.

Hutchinson, J., M. Rouncefield, and J. Whittle (2011). "Model-driven Engineering Practices in Industry." In: *Proceedings of the 33rd International Conference on Software Engineering (ICSE'11).*

Hutchinson, J., J. Whittle, and M. Rouncefield (2014). "Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure." In: *Science of Computer Programming* 89.

Hutchinson, J., J. Whittle, M. Rouncefield, and S. Kristoffersen (2011). "Empirical assessment of MDE in industry." In: *Proceedings of the 33rd International Conference on Software Engineering (ICSE'11).*

INCOSE (2007). *Systems engineering vision 2020*. Tech. rep. International Council on Systems Engineering.

ISO/IEC (2003). *ISO/IEC 15504:2004 Information technology – Process assessment – Part 4: Guidance on use for process improvement and process capability determination*. Tech. rep. International Organization for Standardization, Geneva, Switzerland.

ISO/IEC (2010). *ISO/IEC 25010 System and software quality models*. Tech. rep. International Organization for Standardization, Geneva, Switzerland.

Jitpaiboon, T. (2014). "The Study of Competitive Priorities and Information Technology Selection: Exploring Buyer and Supplier Performance." In: *Journal of International Technology and Information Management* 23.

Jolak, R., E. Umuhoza, T. Ho-Quang, M. R. V. Chaudron, and M. Brambilla (2017). "Dissecting Design Effort and Drawing Effort in UML Modeling." In: *Proceedings of the 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA'17).*

Kambil, A., A. Ginsberg, and M. Bloch (1996). "Re-Inventing Value Propositions." In: *NYU Stern School of Business Research Paper Series.*

Karban, R., T. Weilkiens, R. Hauber, and R. Diekmann (2010). *MBSE Practices in Telescope Modeling*. Tech. rep.

Kaslow, D., B. Ayres, P. T. Cahill, L. Hart, and R. Yntema (2017). "A Model-Based Systems Engineering (MBSE) approach for defining the behaviors of CubeSats." In: *Proceedings of the IEEE Aerospace Conference*.

Klement, A. (2018). *When Coffee and Kale Compete: Become Great at Making Products People Will Buy*. CreateSpace Independent Publishing Platform.

Kneuper, R. (2008). *CMMI: Improving Software and Systems Development Processes Using Capability Maturity Model Integration*. Rocky Nook.

Krajewski, L. and L. Ritzman (2018). *Operations management : strategy and analysis*. Addison-Wesley.

Kuhn, A., G. C. Murphy, and C. A. Thompson (2012). "An exploratory study of forces and frictions affecting large-scale model-driven development." In: *Proceedings of the ACM/IEEE 15th International Conference on Model Driven Engineering Languages and Systems (MODELS'12)*.

Likert, R. (1932). "A technique for the measurement of attitudes." In: *Archives of Psychology* 22.140.

Little, J. D. C. (1970). "Models and Managers: The Concept of a Decision Calculus." In: *Management Science* 16.8.

Lorenz, W. and M. B. Brasil (2014). "Activity-Based Software Process Lines Tailoring." In: *International Journal of Software Engineering and Knowledge Engineering* 24.

MBSE and SysML. Accessed: 17.05.2021. URL: `https://www.visual-paradigm.com/guide/sysml/mbse-and-sysml/`.

Mandić, V. and N. Gvozdenović (2017). "An extension of the GQM+Strategies approach with formal causal reasoning." In: *Information and Software Technology* 88.

Martin, H., Z. Ma, C. Schmittner, B. Winkler, M. Krammer, D. Schneider, T. Amorim, G. Macher, and C. Kreiner (2020). "Combined automotive safety and security pattern engineering approach." In: *Reliability Engineering and System Safety* 198.

Maxwell, J. A. (2012). *Qualitative research design: An interactive approach*. Vol. 41. Sage publications.

McConnell, S. (2004). *Code Complete, Second Edition*. Microsoft Press.

Merriam-Webster (2021). "Context." In: *Merriam-Webster.com dictionary*. URL: `https://www.merriam-webster.com/dictionary/context` (visited on 05/24/2021).

Mettler, T. and R. Winter (2010). "Towards a Classification of Maturity Models in Information Systems." In: *Management of the Interconnected World*. Physica-Verlag HD.

Mijnhardt, F., T. Baars, and M. Spruit (2016). "Organizational characteristics influencing sme information security maturity." In: *Journal of Computer Information Systems* 56.2.

Miller, J. and J. Mukerji (2003). *MDA Guide Version 1.0.1*. Tech. rep. Object Management Group (OMG).

Minh, V. T., A. Moustafa, and M. Tamre (2017). "Design and simulations of dual clutch transmission for hybrid electric vehicles." In: *International Journal of Electric and Hybrid Vehicles* 9.4.

Motamedian, B. (2013). "MBSE Applicability Analysis." In: *International Journal of Scientific and Engineering Research*.

Neuman, W. L. (2010). *Social Research Methods: Qualitative and Quantitative Approaches*. Alpha Books.

Northrop, L. M. (2006). "Software product lines: reuse that makes business sense." In: *Proceedings of the 15th Australian Software Engineering Conference (ASWEC'06)*.

OMG (2010). *Business Motivation Model - Version 1.1*. Object Management Group.

PMI (2017). *PMI's Pulse of the Profession*. Project Management Institute.

Pan, S. Q., M. Vega, A. J. Vella, B. H. Archer, and G. R. Parlett (1996). "A mini-Delphi approach: An improvement on single round techniques." In: *Progress in Tourism and Hospitality Research* 2.1.

Peppard, J. (2017). *A Tool to Map Your Next Digital Initiative*. Harvard Business Review.

Peppard, J., J. Ward, and E. Daniel (2007). “Managing the realization of business benefits from IT investments.” In: *MIS Quarterly Executive* 6.1.

Peterson, M. (2017). *An introduction to decision theory*. Second edition. Cambridge introductions to philosophy. Cambridge University Press.

Pohl, K., M. Broy, H. Daembkes, and H. Hönninger (2016). *Advanced Model-Based Engineering of Embedded Systems, Extensions of the SPES 2020 Methodology*. Springer.

Pöppelbuß, J. and M. Röglinger (2011). “What makes a useful maturity model? a framework of general design principles for maturity models and its demonstration in business process management.” In: *Proceeding of the European Conference on Information Systems (ECIS’11)*.

Rao, L., G. Mansingh, and K. Osei-Bryson (2012). “Building Ontology Based Knowledge Maps to Assist Business Process Re-Engineering.” In: *Decision Support Systems* 52.

Riungu-Kalliosaari, L., S. Mäkinen, L. E. Lwakatare, J. Tiihonen, and T. Männistö (2016). “DevOps Adoption Benefits and Challenges in Practice: A Case Study.” In: *Proceedings of the 17th International Conference of Product-Focused Software Process Improvement (PROFES’16)*.

Runeson, P. and M. Höst (2008). “Guidelines for conducting and reporting case study research in software engineering.” In: *Empirical Software Engineering* 14.2.

Saaty, T. L. (1980). *The analytic hierarchy process : planning, priority setting, resource allocation*. McGraw-Hill International Book Co.

Sales, T. P., F. A. Baião, G. Guizzardi, J. P. A. Almeida, N. Guarino, and J. Mylopoulos (2018). “The Common Ontology of Value and Risk.” In: *Proceedings of the 37th International Conference on Conceptual Modeling (ER’18)*.

Salimi, F. and F. Salimi (2017). *A Systems Approach to Managing the Complexities of Process Industries*. Elsevier Science.

Saltelli, A. (2008). *Global sensitivity analysis: the primer*. John Wiley & Sons, Inc.

Sanchez-Puchol, F. and J. A. Pastor-Collado (2017). "Focus Area Maturity Models: A Comparative Review." In: *Information Systems*.

Schmitt, A. and P. Diebold (2016). "Why Do We Do Software Process Improvement?" In: *Proceedings of the 17th International Conference of Product-Focused Software Process Improvement (PROFES'16)*.

Serra, C. E. M. and M. Kunc (2015). "Benefits Realisation Management and its influence on project success and on the execution of business strategies." In: *International Journal of Project Management* 33.1.

Shields, P. and N. Rangarjan (2013). *A Playbook for Research Methods: Integrating Conceptual Frameworks and Project Management*. New Forums.

Simmons, J. P., R. A. LeBoeuf, and L. D. Nelson (2010). "The effect of accuracy motivation on anchoring and adjustment: do people adjust from provided anchors?" In: *Journal of Personality and Social Psychology* 99.6.

Sowden, R. (2011). *Managing Successful Programmes*. TSO.

Stol, K. and B. Fitzgerald (2015). "A Holistic Overview of Software Engineering Research Strategies." In: *Proceeding of the 3rd IEEE/ACM International Workshop on Conducting Empirical Studies in Industry (CESI'15)*.

Torpp, J. (1999). *The informaton paradox*. McGraw-Hill Inc.,US.

Trendowicz, A., J. Heidrich, and K. Shintani (2011). "Aligning Software Projects with Business Objectives." In: *Proceedings of the 21st International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA'11)*.

Van Lamsweerde, A. (2001). "Goal-Oriented Requirements Engineering: A Guided Tour." In: *Proceedings of the 5th IEEE International Symposium on Requirements Engineering (RE'01)*.

Vetschera, R. (2006). "Preference-Based Decision Support in Software Engineering." In: *Value-Based Software Engineering*. Springer Berlin Heidelberg.

Voelter, M., J. Siegmund, T. Berger, and B. Kolb (2014). "Towards User-Friendly Projectional Editors." In: *Software Language Engineering*. Springer International Publishing.

Vogelsang, A. (2019). "Feature Dependencies in Automotive Software Systems: Extent, Awareness, and Refactoring." In: *Journal of Systems and Software*.

Vogelsang, A., H. Femmer, and C. Winkler (2015). "Systematic Elicitation of Mode Models for Multifunctional Systems." In: *Proceedings of the 21st International Working Conference on Requirement Engineering: Foundation for Software Quality (REFSQ'15)*.

Vogelsang, A., S. Eder, G. Hackenberg, M. Junker, and S. Teufl (2014). "Supporting Concurrent Development of Requirements and Architecture: A Model-based Approach." In: *Proceedings of the 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD'14)*.

Vogelsang, A., T. Amorim, F. Pudlitz, P. Gersing, and J. Philipps (2017). "Should I Stay or Should I Go? On Forces that Drive and Prevent MBSE Adoption in the Embedded Systems Industry." In: *Proceedings of the 18th International Conference of Product-Focused Software Process Improvement (PROFES'17)*.

Wagenaar, Gerard, Sietse Overbeek, Garm Lucassen, Sjaak Brinkkemper, and Kurt Schneider (2018). "Working software over comprehensive documentation – Rationales of agile teams for artefacts usage." In: *Journal of Software Engineering Research and Development* 6.1, p. 7.

Ward, P. T., J. K. McCreery, L. P. Ritzman, and D. Sharma (2007). "Competitive Priorities in Operations Management." In: *Decision Sciences* 29.4.

Weston, C., T. Gandell, J. Beauchamp, L. McAlpine, C. Wiseman, and C. Beauchamp (2001). "Analyzing Interview Data: The Development and Evolution of a Coding System." In: *Qualitative Sociology* 24.3.

Whittle, J., J. E. Hutchinson, and M. Rouncefield (2014). "The State of Practice in Model-Driven Engineering." In: *IEEE Software* 31.3.

Whittle, J., J. Hutchinson, M. Rouncefield, H. Burden, and R. Heldal (2017). "A Taxonomy of Tool-related Issues Affecting the Adoption of Model-driven Engineering." In: *Software and System Modeling* 16.2.

Xu, P. and B. Ramesh (2007). "Software Process Tailoring: An Empirical Investigation." In: *Journal of Management Information Systems*.

Yu, E., P. Giorgini, N. Maiden, and J. Mylopoulos (2011). *Social Modeling for Requirements Engineering*. The MIT Press.

de Bruin, T., M. Rosemann, R. Freeze, and U. Kulkarni (2005). "Understanding the main phases of developing a maturity assessment model." In: *Proceedings of the 16th Australasian Conference on Information Systems (ACIS'05)*.

van Steenbergen, M., R. Bos, S. Brinkkemper, I. van de Weerd, and W. Bekkers (2010). "The Design of Focus Area Maturity Models." In: *Proceedings of the 5th International Conference on Global Perspectives on Design Science Research (DESRIST'10)*.

– (2013). "Improving IS Functions Step by Step: the Use of Focus Area Maturity Models." In: *Scandinavian Journal of Information Systems* 25.2.

van de Weerd, I., W. Bekkers, and S. Brinkkemper (2010). "Developing a Maturity Matrix for Software Product Management." In: *Software Business*. Springer Berlin Heidelberg.

ERKLÄRUNG ZUR DISSERTATION

„Hiermit versichere ich an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel und Literatur angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Werken dem Wortlaut oder dem Sinn nach entnommen wurden, sind als solche kenntlich gemacht. Ich versichere an Eides statt, dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie - abgesehen von unten angegebenen Teilpublikationen und eingebundenen Artikeln und Manuskripten - noch nicht veröffentlicht worden ist sowie, dass ich eine Veröffentlichung der Dissertation vor Abschluss der Promotion nicht ohne Genehmigung des Promotionsausschusses vornehmen werde. Die Bestimmungen dieser Ordnung sind mir bekannt. Darüber hinaus erkläre ich hiermit, dass ich die Ordnung zur Sicherung guter wissenschaftlicher Praxis und zum Umgang mit wissenschaftlichem Fehlverhalten der Universität zu Köln gelesen und sie bei der Durchführung der Dissertation zugrundeliegenden Arbeiten und der schriftlich verfassten Dissertation beachtet habe und verpflichte mich hiermit, die dort genannten Vorgaben bei allen wissenschaftlichen Tätigkeiten zu beachten und umzusetzen. Ich versichere, dass die eingereichte elektronische Fassung der eingereichten Druckfassung vollständig entspricht."

Teilpublikationen:

- Tiago Amorim et al. (2019). "Strategies and Best Practices for Model-based Systems Engineering Adoption in Embedded Systems Industry." In: *Proceedings of of the 41st ACM/IEEE International Conference on Software Engineering (ICSE'19)*
- Andreas Vogelsang et al. (2017). "Should I Stay or Should I Go? On Forces that Drive and Prevent MBSE Adoption in the Embedded Systems Industry." In: *Proceedings of of the 18th International Conference of Product-Focused Software Process (PROFES'17)*

Köln, May 2021