# Processes, methods, and tools in model-based engineering—A qualitative multiple-case study



Citation for the original published paper (version of record):

Holtmann, J., Liebel, G., Steghöfer, J. (2024). Processes, methods, and tools in model-based engineering—A qualitative multiple-case study. Journal of Systems and Software, 210. http://dx.doi.org/10.1016/j.jss.2023.111943

The Journal of Systems and Software 210 (2024) 111943

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

# Processes, methods, and tools in model-based engineering—A qualitative multiple-case study☆

Jörg Holtmann [a,*], Grischa Liebel [b], Jan-Philipp Steghöfer [c]

[a] Department of Computer Science & Engineering, Chalmers | University of Gothenburg, Gothenburg, Sweden
[b] Department of Computer Science, School of Technology, Reykjavik University, Reykjavik, Iceland
[c] XITASO GmbH IT & Software Solutions, Augsburg, Germany

ARTICLE INFO

*Keywords:*
Model-based engineering
Development processes
Modeling methods

ABSTRACT

Research on model-based engineering (MBE) has occasionally touched upon the relationship between development processes and concrete MBE practices. However, the alignment of these elements has rarely been the central focus of these studies. As a result, important questions regarding the alignment of MBE and development processes, as well as the impact of development processes on the utilization and success of MBE, have remained unanswered.

To address this research gap, we conducted a multiple-case study involving 14 individuals from nine different companies, conducting a total of 12 interviews. Building upon seven propositions derived from existing literature, our investigation sought to understand how MBE is aligned with the development process and explore the application of MBE in this context. Additionally, we identified challenges and needs in this area.

Our findings challenge some previously reported results, such as the perceived conflicts between agile development processes and MBE. Furthermore, we unearthed previously unreported issues, like the importance of considering the perspectives of tool vendors in MBE discussions.

Overall, this paper makes a significant contribution by providing a comprehensive and up-to-date perspective on how MBE is integrated into development processes, along with an examination of the social and organizational aspects inherent to these processes.

*Editor's note: Open Science material was validated by the Journal of Systems and Software Open Science Board.*

## 1. Introduction

Rigorous development processes are crucial for the development of complex systems and software. There is a multitude of standards (e.g., for systems and software engineering (ISO et al., 2023, 2017)) and assessment frameworks (e.g., SPICE (ISO and IEC, 2021) or its automotive-specific derivative Automotive SPICE (Automotive SIG and VDA, 2023)) prescribing development process reference models, which aim at improving product quality through a high process quality. Such process reference models describe lifecycle phases and work products that companies should pass and create and tailor towards their specific needs. Independently of whether companies follow such reference models, they typically document their intended development processes through corresponding descriptions and guidelines. Even if a process is ad-hoc and undocumented, the development still follows this process.

Beyond development processes that are the basis for any development approach, model-based engineering (MBE) is one concrete development approach that aims at coping with the complexity of systems or software under development. It does so by building abstractions of the real world. Depending on the purposes of the MBE application, on the degree of its intensity and pervasiveness in a company, and on the engineering discipline and development lifecycle contexts, MBE promises to facilitate system understanding and communication through abstraction, to increase productivity through automation, or to improve the quality of the system under development, inter alia. In this study, we deliberately use the term model-*based* engineering to indicate that we consider any formality level of modeling (i.e., from whiteboard sketches to automatic model exploitation) and also independent of engineering disciplines.

☆ Editor: Heiko Koziolek.

* Corresponding author.
*E-mail addresses:* jorg.holtmann@gu.se (J. Holtmann), grischal@ru.is (G. Liebel), jan-philipp.steghoefer@xitaso.com (J.-P. Steghöfer).

https://doi.org/10.1016/j.jss.2023.111943
Received 7 August 2023; Received in revised form 2 November 2023; Accepted 22 December 2023
Available online 26 December 2023

However, the actual alignment of development processes with MBE is not yet sufficiently investigated. There are plenty of both qualitative and quantitative studies on MBE that touch process, organizational, or social aspects but do not focus on their actual alignment and interplay: Baker et al. (2005), Mohagheghi and Dehlen (2008), Mohagheghi et al. (2009), Hutchinson et al. (2011a,b, 2014), Whittle et al. (2014), Torchiano et al. (2011, 2013), Tomassetti et al. (2012), Whittle et al. (2013, 2017), Burden et al. (2014), Amorim et al. (2019), Liebel et al. (2018) and Liebel and Knauss (2023). Other MBE studies do not touch these aspects at all: Dobing and Parsons (2006), Reggio et al. (2013, 2014), Forward and Lethbridge (2008), Agner et al. (2013), Mohagheghi et al. (2013), Gorschek et al. (2010, 2014), Liebel et al. (2019) and David et al. (2023). Consequently, Hebig and Bendraou (2014) formulate in their literature review the need to study the impact of MBE on development processes.

In this paper, we contribute to this need and investigate the alignment of MBE with the development processes it is embedded in. To do so, we conducted a qualitative multiple-case study encompassing 12 interviews with 14 persons from nine companies.

We formulated three research questions (RQs) to guide the study:

**RQ1** How is MBE aligned with the development process?

**RQ2** How and why are which model(ing) kinds, languages, and tools applied in different phases of the development process?

**RQ3** What are practitioners' challenges and needs regarding the alignment of MBE with development processes?

In the context of the literature mentioned above, RQ1 and RQ3 have not been considered yet. In contrast, questions similar to RQ2 have been considered in the literature (e.g., in Torchiano et al. (2011), Tomassetti et al. (2012), Torchiano et al. (2013), Reggio et al. (2013), Dobing and Parsons (2006) and Liebel et al. (2019)). However, we included this question to understand the context of the MBE application in the studied cases. We also added the notion that the use of model kinds, languages, and tools can change during the development process.

Based on the research questions, we derived seven propositions from the literature (in particular w.r.t. RQ1). We then mapped these propositions to the questions in a semi-structured interview guide to systematically conduct the study and elicit answers for our RQs.

Overall, our contribution is an up-to-date empirically grounded overview of how practitioners see and apply model-based engineering and thus paints a picture of the current state of MBE and its use in modern development processes. Many of our findings provide a contrast to existing literature and provide deeper insights into the drivers and challenges of using MBE in state-of-the-art systems and software development. Based on our propositions, we identify that:

- whether the tool or the process is adapted when MBE is introduced depends on the size of the organization and its influence on the tool vendor;
- process-mature companies deliberately describe their development processes in a coarse-grained and tool-agnostic manner and complement these descriptions with tool-specific modeling method guidelines;
- agile development and MBE do not impede each other — but rather a lack of suitable tools and methods as well as a lack of knowledge might impede an agile way of working using MBE;
- safety standards and process standards are a major driver for adoption of MBE, and industrial de-facto standards influence the choice of modeling languages;
- the majority but not all companies in our sample apply expert teams to institutionalize MBE;
- the value of models changes across different phases of the development lifecycle, depending on their purpose and model maintenance over time;
- and the information exchange across organizational boundaries using standardized exchange formats plays an important role in MBE, and models of different kinds are involved in this.

The remainder of this paper is structured as follows. The upcoming section presents the related work mentioned above in a more detailed manner. Section 3 describes our research method, and Section 4 sketches the profiles of the interviewees' company affiliations. We then present the study results in Section 5 and discuss them in Section 6, before concluding in Section 7.

## 2. Related work

Various empirical studies have been published in the last two decades focusing on the adoption of models and MBE in industry. In the following, we summarize studies that cover the use of MBE in relation to process, organizational and social aspects. Additionally, we briefly review more general studies that investigate the adoption of MBE in industry.

### *2.1. Studies on the application of MBE that cover process, organizational, or social aspects*

Baker et al. (2005) conduct an early qualitative case study reporting experiences at Motorola over a period of 20 years, including successes and failures. Besides the well-known benefits of MBE like productivity and quality gains and a defect identification early in the development lifecycle, they observed aspects related to the development process. That is, Motorola initially struggled with the introduction of MBE due to the lack of a well-defined development process resulting in trial and error procedures, and due to an inflexibility regarding cultural changes. To tackle these issues, Motorola introduced a rigorous top-down development process accompanied by the development and deployment of technologies and tools. They also installed a technical advisory board supporting the application of MBE in several ways.

Mohagheghi and Dehlen (2008) conduct a systematic literature review on industrial experience reports about the application of model-driven engineering. Their findings indicate issues in the readiness of software development processes for model-driven engineering. For example, they state that model-driven engineering does not make any assumptions about the processes, and the application of model-driven engineering can be impeded by the lack of well-defined processes, required knowledge, and the ability to change the existing culture — also citing (Baker et al., 2005). However, Mohagheghi and Dehlen (2008) focus on model-driven engineering in the sense of using models only as primary artifacts and on models only in the discipline of software engineering.

In a follow-up paper, Mohagheghi et al. (2009) report on challenges in the introduction of MBE elicited from two companies as part of two research projects. They report that MBE "is a long-term investment and needs customization of environment, tools and processes, and training". Furthermore, they report on challenges regarding the integration of tool chains, and propose to build expert teams to provide support in adopting MBE in companies. However, a deeper investigation regarding development processes and MBE is not in their scope.

In a series of studies to empirically investigate the industrial application of MBE, Hutchinson et al. (2011a,b, 2014) and Whittle et al. (2014) combine data from qualitative, semi-structured interviews with practitioners from different companies and a quantitative questionnaire survey.

Hutchinson et al. (2011a) investigate the introduction and adoption of MBE through three qualitative case studies, selecting and aggregating interviews from three companies from different industry sectors each. Their results indicate that the success or failure of the introduction of MBE practices is connected to organizational aspects; for example, a company should be transparent about the goals of introducing MBE so that the employees consider it meaningful.

Hutchinson et al. (2011b) conduct a more general study on the industrial state-of-the-art of the MBE application. In this study, they do not only focus on the interviews from the three companies mentioned above, but consider more interviews and complement them with the results of an online questionnaire. Whereas the latter quantitative part yields typical data on the kinds of applied modeling languages, the reasons to apply MBE, etc., the former qualitative part yields organizational, social, and educational aspects in the application of MBE. Hutchinson et al. (2014) updates and aggregates the results from Hutchinson et al. (2011a,b).

In summary, the studies by Hutchinson et al. do not consider the alignment of the application of MBE with the development processes that MBE is embedded in. Nevertheless, they yield interesting results regarding organizational, social, and educational aspects of the application of MBE, which partially match with some results of our study.

In a different series of studies on MBE in the Italian industry, Torchiano et al. (2011, 2013) and Tomassetti et al. (2012) conduct a quantitative questionnaire survey with software professionals. The results presented by Torchiano et al. (2011) show that the Unified Modeling Language (UML) (Object Management Group, 2023c) is mostly used by the participating practitioners, whereas their process-related results show which roles perform the actual modeling activities (i.e., mostly architects, project managers, and software developers — but only few domain experts).

Tomassetti et al. (2012) focus on the maturity of MBE, that is, the application of more advanced MBE features like model interpretation and transformation as well as code generation. To investigate this aspect, they complement the results of the survey by interviews with three of the participating practitioners. Their results show that small companies are more mature in the application of MBE. Two of their three interviewees assume that this is due to smaller companies being more flexible regarding the application of new technologies, and that there is more resistance in larger companies against such new technologies due to threats to personal competence niches.

Torchiano et al. (2013) complement this interview result with the authors' assumption for a higher MBE maturity in small companies that such companies might have more competitive pressure. However, a deeper investigation of the alignment between the MBE application with the development processes is not in the authors' scope.

Re-analyzing the qualitative data of Hutchinson et al. (2011a, 2014) and adding new data to it, Whittle et al. (2013, 2017) conduct a qualitative study on the tooling aspects in the MBE application to conceive a taxonomy of tooling-related MBE issues. For this purpose, they select some of the interviews from Hutchinson et al. (2011a, 2014) and complement them with further interviews from two companies from different industry sectors. Whereas the study focuses on tooling aspects, it again also considers organizational and social factors. One key result is that the tools often impede following processes. However, the actual alignment with the development processes is not considered.

Re-considering the findings of Hutchinson et al. (2011a, 2014) in a different manner, Burden et al. (2014) conduct a qualitative multi-case study based on interviews in three different companies operating in partially different industry sectors. The authors validate or refute the particular findings of Hutchinson et al. (2011a, 2014). Relevant to our study is the finding that agile development and MBE can be applied together in a very successful way, although there are certain impediments. Again, the authors do not consider the actual alignment between MBE and development processes but touch some process and organizational aspects.

Hebig and Bendraou (2014) conduct a systematic literature review on the impact of MBE on software development processes. Their main findings regarding the surveyed approaches yield that in all approaches the processes make assumptions about the application of MBE, that in approximately half of the approaches the processes were tailored to the MBE applications, and that the general topic has to be studied more extensively — thereby also motivating our study. However, the surveyed approaches were not exclusively limited to industrial case studies, and the focus was on the software engineering discipline.

Amorim et al. (2019) conduct a qualitative interview study followed by validation through a questionnaire survey to elicit best practices in the introduction of Model-Based Systems Engineering in sectors developing embedded systems. They state that processes and tools in this context are strongly intertwined, and the interviews and best practices yield that open tool interfaces are important, tooling is costly so that it has to be planned thoroughly, processes should be rigorously documented so that the tool application is easier, and that training is important but should be conducted by technical personnel and not by sales people. However, their focus is on Model-based Systems Engineering and the embedded systems industry.

In a survey conducted with 113 industry practitioners in the embedded systems domain, we find that models are used to address various concerns in systems engineering (Liebel et al., 2018). Whereas the study focuses primarily of general benefits and shortcomings of models and MBE, several process-related concerns are revealed, such as difficulties in interoperability between modeling tools and difficulties integrating models into the development process.

In the context of requirements models, the second author conducted an in-depth case study in the telecommunications domain on the use of models in agile software development (Liebel and Knauss, 2023). The authors find that requirements models are used to effectively maintain system-level knowledge of the requirements, while agile teams focus on smaller increments in terms of user stories. The models thus help to connect different methodological islands in the company. The study further reveals a number of trade-off decisions with respect to fidelity of models and tooling, for example, to use automated layout to allow for the use of text-based models that can be maintained in traditional version control systems alongside with source code.

### 2.2. Further studies on the industrial application of MBE

After an initial literature review and preliminary qualitative interviews with practitioners, Dobing and Parsons (2006) conduct a quantitative survey on the usage of different model kinds provided by the UML in its version 1.5. The main results are that the model kinds of classes, use cases, and interactions are the most frequently used ones. Furthermore, the study raises historically quite early the issue that the UML might be too complex. However, process-related aspects are not in the authors' scope.

Similarly, Reggio et al. (2013, 2014) conduct a quantitative study on the usage of the different model kinds in the UML version 2.4.1. The authors gain similar results, but process-related aspects are again not considered.

Forward and Lethbridge (2008) conduct a quantitative questionnaire survey amongst practitioners with the aim of comparing their experiences regarding MBE as well as non-MBE approaches. Their findings show that MBE approaches are better for communication and understandability than non-MBE approaches, but it is harder for MBE approaches to keep models and generated code in synchronization and establish traceability between them. However, process-related aspects are not in their scope.

Agner et al. (2013) conduct a quantitative questionnaire survey amongst industrial practitioners in Brazil to investigate the usage of UML in the context of embedded software development. Their main results yield that UML in this context is considered complex and mainly used, if at all, for rather lightweight purposes like improving communication and understanding. Although stating the known MBE benefits of improving quality and productivity, the participants rarely use more advanced MBE features like model transformation or code generation. Besides the narrow focus on UML and embedded software development, process-related aspects are not in the authors' scope.

 

In the context of a joint research project, Mohagheghi et al. (2013) conduct three qualitative case studies with three companies from different industry sectors. The three cases are very different in their scopes and the research methods, and Mohagheghi et al. describe mainly technical results of positive and negative aspects in the application or introduction of model-driven engineering. An investigation of the alignment of MBE with the development processes is not in their scope.

Gorschek et al. (2014) re-analyze the data from their quantitative large-scale questionnaire survey (Gorschek et al., 2010) with a focus on the use of design models before coding. For this purpose, they relate the demographic data regarding roles, experience, etc. with the question whether the participants apply design models for guiding their actual software development activities. Their results yield that the sample of participants rarely or never uses design models in this context. Except for the demographic questions on the participants' roles, no process-related aspects are considered.

In a case study at two automotive companies conducted by the second author (Liebel et al., 2019), we find that models are used in a requirements engineering context to improve communication and to manage complexity. However, stakeholders prefer whiteboard sketches over formal modeling notations. The study focuses on models in requirements engineering only, and does not specifically consider processes or MBE.

A recent study by David et al. (2023) investigates the application of collaborative model-driven software engineering in the industry. For this purpose, they conduct qualitative interviews with seven practitioners as part of two focus groups and complement them with data from a quantitative questionnaire survey. However, the study focuses on technical features and needs on them regarding collaboration in the context of model-driven software development and does not consider process-related aspects.

## 3. Research method

Fig. 1 visualizes our research method, which we explain in this section. In a nutshell, we conducted a multiple-case study by conducting interviews with practitioners from different companies (which we consider individual cases) with the goal to support or refute a number of propositions that we derived from the related work.

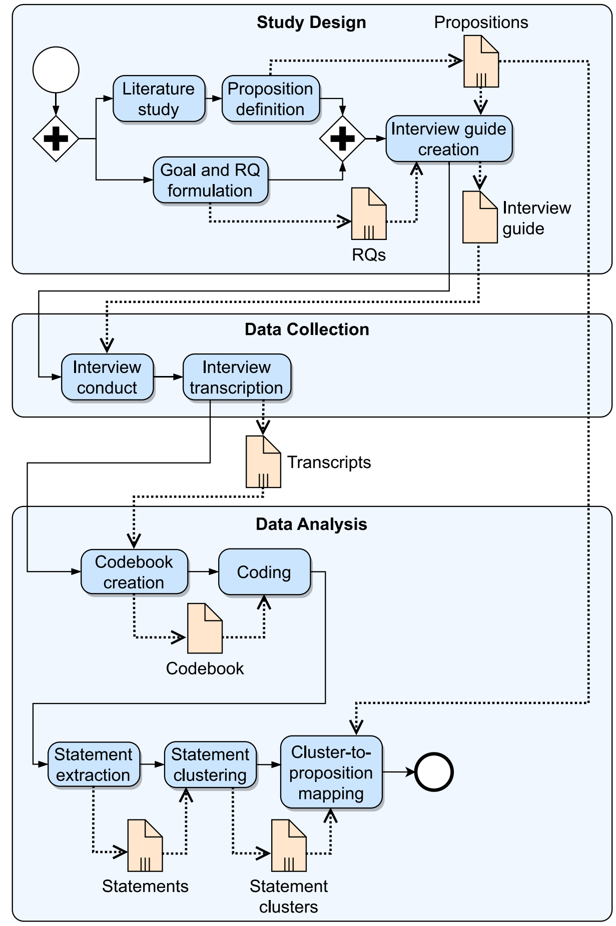


**Fig. 1.** Our research method: Based on a review of the literature, we created seven propositions that we then evaluated based on interviews in a multiple-case study.

### 3.1. Study Design

Since we are conducting an exploratory case study (Runeson et al., 2012), we use an inductive approach without a specified theory at the beginning of the study. As visualized through the super-activity Study Design in Fig. 1, we formulated the goal as well as the RQs and in parallel conducted a literature study from which we derived seven propositions for RQ1 and RQ2 (see Table 1). These propositions present a limited set of statements made in relation to our research questions and allow for the exploration of the study topic without limiting the scope. We deliberately did not derive propositions for RQ3, to not restrict the focus on challenges and needs.

We used a semi-structured interview guide for the data collection, consisting of 5 demographic questions and 19 questions. We created the interview guide based on our RQs and propositions, maintaining links between the interview questions and the RQs as well as hypotheses. We provide the interview guide including the mapping of the propositions to the particular questions as part of our supplementary material (Holtmann et al., 2023).

Note that during the interviews, we neither defined exactly what we understood as a model nor what we considered a process description or modeling method. We therefore gave the interviewees the freedom to use their own understanding of these concepts without having to map them to an unfamiliar definition.

### 3.2. Data Collection

The super-activity Data Collection in Fig. 1 visualizes our data collection process. We used purposive sampling, more specifically expert sampling (Etikan et al., 2016) to identify practitioners that are using MBE in the development of software systems. The population for the sampling was the pool of combined contacts of the authors. We created an initial list of about 50 candidates who were known for being familiar with MBE. From this list, we selected 18 which we found most suitable based on their prior experience. We also selected participants for diversity, considering their roles, the industry they work in as well as the size of the organization.

After contacting these 18 practitioners, we got agreements for interviewing 14. Table 2 provides an overview of these 14 interviewees and their affiliations. We interviewed three people from the same company in a focus group interview FG (see row 3 in Table 2). The remaining 11 persons were interviewed separately. We were able to interview two practitioners each from the automotive tool vendor/consulting company, the printing company, and the space/industry/innovation company (see rows 5–6, 9–10, and 11–12 in Table 2, respectively). The remaining 5 practitioners belong to individual companies. In summary, we conducted 12 interviews with 14 persons (i.e., one focus group with three persons and 11 single-person interviews) from 9 companies operating in heterogeneous industry sectors.

All interviews were conducted by the first author of this paper, with one of the other authors assisting in the majority of them. The interview time ranged between 48 and 107 min.

**Table 1**
Propositions from related work.

| No. | Description | RQ | References |
|---|---|---|---|
| P1 | Processes, modeling methods, and modeling tools have to fit to each other. When MBE is introduced, either the process has to be adapted, or the tools have to be customized or newly developed. | RQ1 | Hebig and Bendraou (2014) and Whittle et al. (2013, 2017) |
| P2 | Development processes are not tailored to MBE, and MBE does not make any assumptions on the development process. | RQ1 | Mohagheghi and Dehlen (2008) |
| P3 | Agile development processes and MBE impede each other. | RQ1 | Kuhrmann et al. (2022) and Bucchiarone et al. (2020) |
| P4 | External regulations or industrial de-facto standards foster and push MBE. | RQ1 | Whittle et al. (2013, 2017) |
| P5 | MBE should be supported through expert teams in order to be successful. | RQ1 | Baker et al. (2005) and Mohagheghi et al. (2009) |
| P6 | The value of different kinds of models changes throughout the different phases of the development lifecycle. | RQ2 | Stachowiak (1973) |
| P7 | Certain models are used as boundary objects to exchange information between methodological islands. | RQ2 | Liebel and Knauss (2023) |

**Table 2**
Overview interviewees and companies.

| Row | Type | Company | Role | ID |
|---|---|---|---|---|
| 1 | Interview[a] | Corporation A | Safety expert | Int1 |
| 2 | Interview | Electronics & tool vendor | Software developer | Int2 |
| 3 | Focus group | Multi-sector tool vendor / consulting 1 | Software architect<br>Scrum master / project manager<br>Development manager | FG |
| 4 | Interview | Multi-sector tool vendor / consulting 2 | Meta-automation consultant | Int3 |
| 5 | Interview[a] | Automotive tool vendor / consulting | Solution engineer | Int4 |
| 6 | Interview[a] | | Solution engineer | Int5 |
| 7 | Interview | Automotive OEM | Key account manager | Int6 |
| 8 | Interview[a] | Automotive tier-1 supplier | Technology advisor | Int7 |
| 9 | Interview | Printing | Performance modeling researcher | Int8 |
| 10 | Interview | | System architect | Int9 |
| 11 | Interview | Space / industry / innovation | CTO | Int10 |
| 12 | Interview | | Unit manager | Int11 |

[a] Throughout this paper, we translated the snippets from this interview from German.

One of the participants did not agree to recording the interview. We analyzed that interview based on our extensive notes. We transcribed the first recorded interview manually. For the remaining interviews, we used the automatic transcription feature of Microsoft Teams to get an initial transcript. Afterwards, we refined the initial transcript by correcting transcription errors, resolving technical terms, and anonymizing. The first author transcribed all interviews except one verbatim, anonymized, and sent them out for review to the interviewees who opted for a review.

### 3.3. Data Analysis

The super-activity Data Analysis in Fig. 1 visualizes the steps of this final part of our study. This activity follows thematic analysis (Cruzes and Dyba, 2011).

All authors initially coded a single transcript using open, descriptive coding (Saldaña, 2015), closely connected to our RQs. We then jointly agreed on a codebook that was used to code all transcripts. To ensure familiarization with the data, we assigned coders so that they would process transcripts of interviews they did not participate in.

After completing coding, we extracted statements into a virtual whiteboard using Miro (RealtimeBoard, 2023). To do so, we used the assigned codes to determine relevance to the RQs. In an iterative fashion, we then grouped statements into related clusters and discussed the resulting themes jointly.

### 3.4. Threats to validity

We structure the threats to validity according to the taxonomy of Runeson and Höst (2008) and Runeson et al. (2012).

#### 3.4.1. Construct validity

The first author of the paper created the interview guide, which he afterwards refined in multiple discussions with both co-authors. By doing so, we reduced the possibility that the interviewees understand terms and definitions in a different way than the authors. This procedure also helped us to identify questions that could be considered suggestive. One focus of these discussions was the alignment of the interview questions with the research questions and the seven propositions we had identified.

All interviewees participated voluntarily and did not receive any compensation. Furthermore, we sent a consent form at least one day before the interview to inform all interviewees about the purpose of the study, the interview and study procedures, and that we protect their anonymity and confidentiality. The consent form was signed by all interviewees. Moreover, the interviewees had the option of reviewing and correcting their transcript (i.e., *member checking* (Candela, 2019)). Thus, we expect that interviewees did not avoid certain topics and answered our questions truthfully.

Finally, it was always the case that at least one of the interviewing authors had not only a professional but also a personal connection with the interviewees. This ensured an open and friendly interview atmosphere, so that the interviewees did not hesitate to ask questions if understandability or other issues occurred. We used open questions exclusively to avoid biasing the participants and to avoid situations where they tried to guess what the interviewer wanted to hear.

#### 3.4.2. Internal validity

To reduce the possibility that a single interview has too much influence on the study results, we used data triangulation in different places. Where possible, we interviewed multiple persons and roles per company (see Table 2), so that single persons with strong opinions did not have too much impact. Furthermore, we interviewed persons from heterogeneous industry sectors and engineering disciplines to get a diverse picture. Finally, we extracted clusters of statements that were mentioned by multiple interviewees, so company affiliations did not have too much influence.

To establish continuity during data collection, the first author led all interviews with the same interview guide (Holtmann et al., 2023), in many interviews with the assistance of one of the other two authors on a rotating basis. However, semi-structured interviews allow freedom in the order of the questions, through follow-up questions and the general flow of the interview. We mitigated this threat through member checking by giving the interviewees the option to review their transcripts.

#### 3.4.3. External validity

The external validity of case studies is low by nature. Thus, we cannot and do not intend to claim that the study results are generalizable to other companies or to their respective or other industry sectors. However, current research on qualitative research shows that our number of interviews is well within the range sufficient to reach saturation (Hennink and Kaiser, 2022) and that case studies do have strengths in terms of theoretical generalizability (Tsang, 2014).

In addition, development processes, our study topic, are firmly connected to people, and we argue that a case study is a well-suited means to yield such interpersonal and thereby "softer" aspects. Nevertheless, we can mitigate the threat to a certain extent by triangulating statement clusters mentioned by multiple people from heterogeneous companies and industry clusters.

There exists a potential selection bias, as the first and third author approached contacts who they collaborated with as part of different research projects. While participation was voluntarily, this could lead to a biased set of interviewees, for example, being particularly positive towards MBE or connected to research topics.

#### 3.4.4. Reliability

In terms of the case study design, we reduced reliability threats particularly by plenty of internal discussions between all authors on the interview guide and the coding. Additionally, we conducted the data analysis jointly through discussion-intensive synchronization meetings.

For transparency and potential replication by other researchers, we provide all material that does not harm the anonymization and confidentiality of our interviewees in our supplemental material Holtmann et al. (2023). This encompasses the semi-structured interview guide, the codebook, the blank informed consent form that our interviewees signed to participate, and the blank mail for inviting the interviewees to the study participation.

To avoid subjective judgement during the transcription, the first author transcribed all but the first of the interviews with the assistance of the automatic transcription service of Microsoft Teams and subsequent manual refinement. However, the abstraction and categorization of the coded statements is, to some extent, subjective. As mentioned above, we mitigated this by a joint and discussion-intensive data analysis.

## 4. Case companies

Table 2 lists the interviewees as well as their company affiliations. For giving an impression of the particular company sizes, we use the categorization of the European Union based on employee numbers: micro-size enterprises employ < 10 persons, small enterprises employ 10–49 persons, medium-sized companies employ 50–249 persons, and large enterprises employ > 250 persons.

### 4.1. Corporation A

Corporation A is a Germany-based large enterprise with many national and international subsidiaries employing a six-digit number of people. The business areas are automation and digitalization in the industry, infrastructure for buildings, decentralized energy systems, mobility solutions in the railway and automotive sectors, and medical technology. The interviewee Int1 (see row 1 in Table 2) is an expert in the area of model-based safety assurance and chose to focus on their experiences from supporting a business unit in the railway sector in the interview.

### 4.2. Electronics & tool vendor

The electronics & tool vendor is a Germany-based large enterprise with international subsidiaries and employs a four-digit number of people. It particularly supports customers in the automotive sector, but also in avionics and industrial manufacturing. The company provides a heterogeneous portfolio of services ranging from testing, simulation, and validation via engineering and consulting to training and support. The interviewee Int2 (see row 2 in Table 2) reported about their experiences from their former position at this company, where they had the position of a software developer for MBE tools and applied MBE in the development of these tools.

### 4.3. Multi-sector tool vendor / consulting 1

This company is a Swedish branch of an India-based large enterprise with international subsidiaries. The overall enterprise employs a six-digit number of people. The branch develops and distributes different kinds of software and tools, particularly for MBE and for domain-specific languages. The group interview we conducted FG encompasses a software architect, a Scrum master/project manager, and a development manager (see row 3 in Table 2). They reported on their own experiences with customers and on insights from different customers, particularly a large telecommunication enterprise but also from customers in regulated medicine and military sectors.

### 4.4. Multi-sector tool vendor / consulting 2

The multi-sector tool vendor/consulting 2 company is a Netherlands-based micro-sized enterprise. It develops MBE tools particularly for domain experts (i.e., not for software engineers) based on domain-specific languages and provides corresponding consulting services. In this context, the meta-automation consultant Int3 (see row 4 in Table 2) particularly reported on their experiences on building MBE tooling for tax law and on the feedback from a tax office, but also on experiences with customers from other sectors.

### 4.5. Automotive tool vendor / consulting

The automotive tool vendor/consulting company is a Germany-based large enterprise employing persons in the four-digit range. It develops MBE and other software tools for application in the automotive embedded sector as well as automotive software and hardware components using their own tools. The team that both interviewees are affiliated with does not follow a strict development or MBE process, but rather adheres to the particular processes of the customers and project partners. The solution engineer Int4 (see row 5 in Table 2) works as part of research projects and to a lesser extent directly with customers, so that they chose to report on insights from industrial project partners and colleagues working for the industry. The solution engineer Int5 (see row 6 in Table 2) reports on their insights from different customers in the automotive domain.

### 4.6. Automotive OEM

This company is a Sweden-based large enterprise employing persons in the five-digit range and acts as an automotive original equipment manufacturer (OEM). The key account manager Int6 (see row 7 in Table 2) has a good overview of the development processes and MBE application within the different departments and reported about these insights.

### 4.7. Automotive tier-1 supplier

This company is a Germany-based large enterprise employing persons in the five-digit range and acts as an automotive tier-1 supplier. The technology advisor Int7 (see row 8 in Table 2) works in a methods and tools department defining standards for and supporting the company's particular business units and hence reported on the units' development processes and MBE application.

### 4.8. Printing

The printing company is a Netherlands-based large enterprise employing persons in the four-digit range. The company is one of the leading companies for the development of products in digital printing and document management. Furthermore, the whole company is a strong advocate of MBE across all engineering disciplines for the development of their products. Interviewee Int8 (see row 9 in Table 2) is currently a researcher in performance modeling, but also had different roles like department head and project leader in the same company for more than 20 years. Thus, they report on all of their company-specific insights. The system architect Int9 (see row 10 in Table 2) works in a MBE method and tools department that supports the company's business units directly within their projects, and they report on their insights from this activity.

### 4.9. Space / industry / innovation

The space / industry / innovation company is a Sweden-based small enterprise. Its business areas focus mainly on computer systems for space missions (where the computer systems are not very safety-critical) and on products in the industrial manufacturing sector. Beyond its small size, the company is quite young and considers itself a startup. Thus, they do not yet have strict development process descriptions in place, and see MBE rather skeptical. They currently use models in the form of informal drawings. The interviewee Int10 (see second to last row in Table 2) is the chief technical officer of the company and reports on their experiences as well as on their skepticism regarding MBE. Similarly, the unit manager Int11 (see last row in Table 2) does not only report about their processes and skepticism but also few insights about their former affiliation.

## 5. Results

In the following, we present the results of our interviews derived during the data analysis part of our methodology structured by the different research questions.

### 5.1. RQ1: How is MBE aligned with the development process?

In the following, we cluster statements of our interviewees regarding the fit of MBE to development processes, the mentions and anchoring of MBE in process descriptions, the institutionalization of MBE, tool and process tailoring, agile development and MBE, and the motivation for the application of MBE.

#### 5.1.1. Fit of MBE to development processes

The approaches applied in the development have to fit to the processes that guide the development (cf. proposition P1 in Table 1). No interviewee reported that MBE does not fit or fits poorly with the particular development processes of the companies. In contrast, most interviewees stated that MBE fits well or very well with the processes. Particularly, the interviewees working in the automotive sector reported that the sector-specific de facto MBE standard AUTOSAR (AUTOSAR GbR, 2023) fits well to the sector-specific process and safety standards Automotive SPICE (Automotive SIG and VDA, 2023) and ISO 26262 (ISO, 2018). Standing out, one interviewee from the printing company emphasized the good fit:

> *"Yeah, there's no question about that [MBE fits to the development process]. I mean, that is for everybody here clear. It's not just for a few, let's say, advocates of MBE clear, it is basically for everybody clear."* — Int9 (printing)

#### 5.1.2. Mentions of MBE or tools in process and method descriptions

Proposition P2 in Table 1 stemming from Mohagheghi and Dehlen (2008) states that development processes and the actual application of MBE are independent of each other. We wanted to investigate this aspect more deeply to also understand how MBE is anchored in the development processes. For this purpose, we asked the interviewees whether and how MBE is mentioned or even prescribed in the process descriptions.

Most interviewees denied that MBE or modeling tools are mentioned in the process descriptions. Particularly, the interviewees from the automotive sector reported that the process descriptions rather prescribe the inputs and outputs of the process steps in a coarse-grained manner instead of concrete recommendations how to apply MBE:

> *"[MBE is mentioned] not in the process, because the process describes* what *has to be delivered but not* how. *That is, it describes that requirements have to be delivered — whether they are specified in a model or not, that is not of interest to the process."* — Int7 (automotive tier-1 supplier)

> *"I'm thinking that process descriptions are not always detailed enough to talk about what to do in which tool and so on. [...] And typically because MBE contains the less firm information, the process descriptions are not so clear in that area. They are more clear when it comes to the PDM systems and factory systems and product documentation of the final product. So therefore the MBE documentation in process descriptions is rather weak, I would say."* — Int6 (automotive OEM)

When it comes to mentions or prescriptions of tools in process descriptions, one interviewee from the multi-sector tool vendor / consulting 1 company had the opinion that they should not be mixed up:

> *"I don't think [using a modeling tool or prescribing the usage of it] belongs to the process. It's more to the development environment that exists. So the process I think is more like way of working, than the tools that you use to work."* — FG (multi-sector tool vendor / consulting 1)

However, there are typically tool-specific modeling method descriptions applicable for the corresponding process steps, and these method descriptions contain concrete recommendations how to specify and exploit the particular model kinds in certain tools (e.g., tool-specific modeling guidelines):

> *"There is a process description and then there is a method description for the respective tool. And then it's a match, so to say, but there is not the process description that says 'you shall use this tool like that', typically."* — Int6 (automotive OEM)

*"One comes from the process to the method, and in the method it is specified for which process steps it is applied. That is, one has a bidirectional traceability between both. But you can use the methods multiple times. For example, you can apply behavior diagrams in requirements engineering, in the architectural design, as well as in the detailed design— then they are referenced multiple times."* — Int7 (automotive tier-1 supplier)

In this context, Int7 also mentioned the advantage of this deliberate distinction: The process descriptions are quite stable as they do not need to change often (i.e., they have fixed long-term release cycles), whereas the method and tool descriptions can be updated on demand.

#### 5.1.3. Institutionalizing MBE

Proposition P5 in Table 1 states that MBE should be supported by expert teams in order to be successful. In this context, our interviewees reported on several approaches to institutionalize MBE in the different companies, which we partitioned into three different clusters described in the following.

*Examples.* Companies provide examples of models that have been proven in practice or are considered otherwise adequate. However, concrete modeling support or documentation is not in the scope of such examples.

*"We use models that we look at when we do implementation, we have that in sort of more of a checklist-oriented process descriptions with a few examples on how it can look like, like design examples from various projects and checklists on what sort of aspects and areas should be covered within the design description."* — Int11 (space / industry / innovation)

*"We have a large server, on which our <CompanyD_ToolA> is hosted, to which basically all employees of the department have access. And mostly, one imitates. Mostly, one looks at modeling projects to get inspiration."* — Int5 (automotive tool vendor / consulting)

*Dedicated modeling support teams.* There are dedicated modeling support teams or departments, who help in applying MBE in concrete projects or create model libraries:

*"We introduced a <CompanyC_DepartmentA>, which is able to help out in the projects to bring the modeling to a higher level, especially when models go across disciplines."* — Int8 (printing)

*"For example, they have libraries, model libraries that they share between different departments. So they have created libraries, for example, containing types of different kinds that are more or less standardized things that everyone needs to use. And they have a special team that manages those models and they, of course, have to be a bit careful when uplifting or changing them, since that will affect many other downstream components."* — FG (multi-sector tool vendor / consulting 1)

*Documentation and training.* The companies explicitly document how to apply a tool to specify certain kinds of models and sometimes also offer training:

*"There are guidelines, of course. These typical modeling handbooks or guidelines, which emerge as soon as one models with multiple people. [...] And, of course, for MBSE and MDE there are corresponding training offers."* — Int1 (corporation A)

*"There are some guidelines how to approach the translation from the <CountryA> law to the formalization in <CompanyM_ToolA>."* — Int3 (multi-sector tool vendor / consulting 2)

*"They have lots of Wiki pages on a Confluence Wiki, where they have articles about many things that kind of should remind people how they should work. So I think I'm pretty sure that in those pages you will find lots of best practices and these guidelines, I mean how to do things. [...] And since they have many people, I guess it helps to be able to point people at, if some less experienced developer has some problems they can point to a page saying 'please read this and follow these steps'. And those steps can of course involve product steps, how to use the product in some way, the modeling product."* — FG (multi-sector tool vendor / consulting 1)

#### 5.1.4. Tailoring of tools towards processes and methods or vice versa

When it comes to the question whether the tools are tailored towards the processes and methods or vice versa (cf. propositions P1 and P2 in Table 1 and Whittle et al. (2013, 2017)), the interviewees answered heterogeneously.

*Tools have no impact on the process.* One portion of the interviewees answered that their organizations never tailored a process or a method towards the tool.

*"I can't think of any cases where the process was tailored for the tool. [...] Typically, it's not so much the process that needs [adaptations], it's more like we need to capture certain data and then we ask for adaptations in the tool so that they can manage it."* — Int6 (automotive OEM)

*"The tool is [tailored towards the process]."* — Int1 (corporation A)

*Process is adapted to a tool when required.* Another portion of interviewees answered that they also try to tailor the processes and methods towards the tools, but this is not always possible.

*"I would say it is fifty/fifty, that is, you try to establish the method first and subsequently the tool. But as you do not create the tool yourself, you have to draw conclusions about the method with all the restrictions that the tools bring with them."* — Int7 (automotive tier-1 supplier)

*"I think we always try to have the process in the first place, agree on that, and then fit the tools to that. But there is some interaction between the two, in the sense that if the tool adaptation is too difficult, then sometimes the process is also slightly altered."* — Int8 (printing)

*Tools are context-sensitive and need to be combined consciously.* Orthogonal to the two kinds of answers above, other interviewees reflected in a more nuanced way about the correspondence between tools and the processes and methods in which they are embedded.

One interviewee working for a tool vendor and consulting company mentioned that they adapt their tools for larger customers, but that such adaptations are typically not generalized:

*"It is rather the way in the automotive industry that a large OEM has its tier-1's, who work for the OEM. And likewise it was like that earlier, from my point of view, that many OEMs bought the particular tools of automotive tool vendor/consulting and worked together with the tool manager for certain problem solutions to apply and adapt the tool correctly. However, there never was someone at automotive tool vendor/consulting, who abstracted and said: 'Our tools are applied by all customers in this and that order—let's provide this in a generic way!'"* — Int5 (automotive tool vendor / consulting)

A different interviewee highlighted the aspect that every tool brings its own method, and that such a method has to fit the context it is applied in:

*"What most people do, they first think about tools. Instead of the other things which are actually in my view bit more important. Tool is the, let's say, the most changeable part of this story. [. . . ] A tool tends to implement or adhere to a certain method. [. . . ] But it's not always a good method for the problem that you are trying to address. So that means that the tool or the method is not suitable for your problem and you may need a different tool or method. [. . . ] So there's no, uh, there is not one method and I also don't believe that there can be one. But of course, there is an overarching, let's say, way of working or on how you combine these things. [. . . ] The company where I am working now, we may use very much the same tools [. . . ] as a different company. But we may use some of them in a different way, because that fits better to our context."* — Int9 (printing)

### 5.1.5. Agile development and MBE

Bucchiarone et al. (2020) state that combining agile development and MBE is a foundational challenge. However, Kuhrmann et al. (2022) state that this combination is traditionally considered incompatible, but their study results are in line with current research that agile development and MBE do not impede each other (cf. proposition P3 in Table 1). The latter aspect is also underpinned by Burden et al. (2014).

In our study, none of the interviewees whose organizations apply an agile or hybrid way of working mentioned contradictions. For example, FG reflected about the perceived contradiction of agile development and MBE:

*"I don't see any contradiction between model-based and Agile. I know there are some people that think there is. Sometimes it is because they have this perception that with models you need to do lots of specification work upfront. Like you're doing waterfall and then you hand over to development. But there is really not something saying that you have to work like that. [...] I think it's a matter of how you use the tool more than that the tool can put some restrictions on the way you can work. So the way I view with this that a model is just like a program source file in some sense."* — FG (multi-sector tool vendor / consulting 1)

The last sentence of this statement was mentioned by the same focus group in different flavors but also by other interviewees who perceive the combination of agile development and MBE in a positive way.

*"I think you can say that in all cases that [our customers] don't treat [...] models very much differently than other artifacts that go into their development process."* — FG1 (multi-sector tool vendor / consulting 1)

*"I think MBE is just a way to capture data. And Agile is a way to organize work."* — Int6 (automotive OEM)

*"I'm always a little bit surprised if model-driven-ness is regarded as something completely different than ordinary software engineering, because it's not. It's only a language on a higher level."* — Int3 (multi-sector tool vendor / consulting 2)

However, FG also stated that in certain cases tools have to be adapted to fit an agile way of working:

*"There was a big effort and that's where we were involved, [. . . ] where [the customer] actually went from a more traditional waterfall process into agile, and that had lots of consequences. [. . . ] it also means that all the tools that they use have to support this workflow. [. . . ] compare/merge is, I think, the best example. There have been lots of requirements on us to ensure that comparing and merging models works as smoothly as comparing and merging source files. [. . . ] because then kind of the whole idea is that you can work in parallel on features without disturbing other teams. And if the modeling tool doesn't support that, then their whole idea breaks."* — FG (multi-sector tool vendor / consulting 1)

Nevertheless, one interviewee even reported that the reason for applying MBE was the need of being more agile:

*"They were not able to change the execution environments based on the law changes, so they have to redraw some of the law changes because they were not able to follow them without a lot of problems. So the whole reason why the model-driven approach was chosen is to increase the agility. And it's literally the agility to change the law."* — Int3 (multi-sector tool vendor / consulting 2)

### 5.1.6. Motivation for the application of MBE

Whittle et al. (2013, 2017) state that external regulations or industrial de facto standards foster and push MBE (cf. proposition P4 in Table 1). In our study, this was confirmed by several interviewees.

In the context of external regulations, the interviewees mention that safety, cybersecurity, and process standards do not prescribe MBE but are demanding when it comes to traceability and other means to provide assessment arguments, and that MBE helps in this regard:

*"Safety standards are relatively strict in what they prescribe regarding traceability. This also results a bit in the things that one has to provide evidence for, that is, what one can do in a tool-supported way. Nobody wants to prescribe Model-based Software Engineering — one can write software by hand in the same way. That is rather a best practice, which has been established over time."* — Int1 (corporation A)

*"Some of the standards are heavy on keeping track of information. Like the functional safety standard or cybersecurity standard. And without the MBE, it's a pain to keep track of the information and capture it in a proper way."* — Int6 (automotive OEM)

Furthermore, safety standards prescribe semi-formal verification for highly safety-critical system parts, which is supported by MBE capabilities like simulation:

*"We came from the document-based world, and have partially transitioned to models. But it will be more and more now, because we get more and more functionality to be implemented. We get more and more safety-critical things, the standards ISO 26262 second edition, the new standard for cyber security, et cetera. They set certain constraints, like a semi-formal verification for ASIL C/D projects, and that we also have to conduct verification, simulation, et cetera. And we will manage this only if we work with models."* — Int7 (automotive tier-1 supplier)

Finally, one interviewee also emphasizes the importance of a better establishment of the comprehension of the system under development through models:

*"Everything is handy that helps to explain someone external (which you have on the side of the assessors) why a system is safe. I establish an understanding internally, where MBE supports me. And I can better communicate it externally, because I have standardized diagrams."* — Int1 (corporation A)

In the context of industrial de facto standards, the high availability of tools and exchange formats due to the sheer prevalence of certain modeling languages motivates the application of MBE. Several interviewees mentioned the application of the wide-spread general-purpose modeling languages UML (Object Management Group, 2023c) and SysML (Object Management Group, 2023b) due to tool availability, although the data exchange possibilities are limited since different tools implement the languages differently. This problem of limited data exchange does not exist for AUTOSAR (AUTOSAR GbR, 2023), which defines an exchange format that the tool vendors adhere to. All interviewees from this industry sector stated that this language is used particularly because of its import and export possibilities. We look at these modeling language aspects in more detail as part of the results for RQ2 in Section 5.2.

Finally, the interviewees mentioned MBE benefits and promises that are already known from the literature. This encompasses a structured and uniform representation, graphical representations, static checks, the establishment of an interdisciplinary understanding of the system under development, communication means, automation, increased efficiency and flexibility, rapid prototyping, investigation and study means, and the exploratory identification of risks early in the development lifecycle.

### 5.2. RQ2: How and why are which model(ing) kinds, languages, and tools applied in different phases of the development process?

In the following, we cluster interviewee statements regarding the drivers for adopting MBE, the scope of modeling, the use of models in the development lifecycle, the selection of modeling languages, and the tooling.

#### 5.2.1. Drivers for adopting model-based engineering

Our interviewees mention a number of drivers for adopting MBE other than external regulations and standards as proposed in P4 (see Table 1). One important one is the increasing complexity of the products that are being developed and, as a consequence, a move towards integration and connections between domains:

> *"We have an increasing number of customers who move into a process where they no longer deliver a single device, but rather have to integrate software from different suppliers onto one device. They do not necessarily know these suppliers and therefore cannot communicate directly, but instead need a way to store [relevant information]."* — Int5 (automotive tool vendor / consulting)

Even if an organization does not need to coordinate several suppliers, exchange of information between teams was also mentioned as a driver. Related to this, models are heralded as a maintainable form of documentation that can be used during the entire lifecycle of the system.

In many other cases, the drivers for adopting MBE are directly coupled with the *purposes of the models* that are being chosen. When interviewees reported on the purpose for modeling, a focus was on analysis of the system, for example, for timing behavior ("check whether the software timings can work out"), validation of requirements, and replacing physical testing with testing of a model.

A second broader focus was on generating artifacts from models. Our interviewees explicitly mentioned generation of documentation, code generation, and generation of variants in a product line.

Apart from these larger aims that often encompass a number of different kinds of models and corresponding languages, we also found the usual requirements models, system architecture models, software architecture and design models, testing models, simulation models as well as models used specifically in safety analysis and for specific domains such as CAD models for hardware. However, these kinds of models were mostly mentioned as supporting one of the other purposes as described above and as part of the larger scope of modeling and the modeling lifecycle as described further below in Section 5.2.3.

#### 5.2.2. Scope of modeling

In systems engineering companies, modeling is not limited to software but also includes the detailed hardware including mechanical and electrical parts. That means that CAD and circuit layout are part of the scope of modeling as well and often driven by the same high-level system structure and requirements, albeit at different steps in the development process (see P6 and P7 in Table 1). In some cases, this even includes disciplines such as chemistry, even though models there are mostly used to describe chemical processes or their outcome rather than used to design them.

Interestingly, this broad scope for modeling does not mean our interviewees consider end-to-end modeling in which the entire scope is captured in one model as particularly useful. As one respondent put it:

> *"You have to take care that you can isolate some aspects [in the models]."* — Int8 (printing)

Part of the reason is that there are specific modeling languages for specific parts of the system (e.g., chemistry, CAD, and PCB design). The other part is that isolation of models also allows mixing informal modeling for specific aspects of a system (such as the system architecture) with more formal modeling for others (e.g., electronics). One respondent considered this distinction useful as it allows them to build an architecture that is "good enough" rather rapidly and with very simple tools, such as online diagram sketching tools. As the system architecture is translated into code manually by a group of developers, the models do not need to be particularly rigorous and thus can be informal. A design for a circuit board, however, needs to be crafted in much more detail so that it can be manufactured from the specification and thus needs to be formal.

In contrast to this, another respondent emphasized that models are "**the** means to develop our products" (emphasis by the authors). That also means that models are generally executable, in particular when it comes to models that describe system behavior. Whereas there are some models that capture, for example, specification of product variability, which cannot be directly executed, there is no distinction between model and code anymore. Thus, the models become the single source of truth in the development process.

At the same time, this means that such models are necessarily very formal, with their semantics provided by the execution engine. Likewise, code generators which are often used to derive source code directly for models similarly impose a semantics and make models formal. Interestingly, one respondent from a tool vendor discussed the role of code generators: In contrast to the original idea of MBE in which platform-specific models are generated from platform-independent models, their experience with customers shows that it is not necessary to generate code in different target languages or for different execution platforms. Their customers therefore usually do not distinguish platform-specific and platform-independent models.

The scope of modeling also includes a notion of completeness. This aspect was particularly emphasized by one of the respondents:

> *"The main thing is [...] to see that we have sort of captured everything that is in the requirements. So everything that we should do is actually captured within somewhere, so you can trace the requirements to the design in some way."* — Int11 (space / industry / innovation)

Traceability is therefore an important part of modeling and needs to be ensured during the entire product lifecycle.

#### 5.2.3. Models in the development lifecycle

Interviewees from organizations that use model-driven engineering extensively, for example, in the automotive domain or when code generation is an integral part of development, describe a back and forth between different languages and tools as the system under construction grows. In these cases and similarly to traceability (see previous section), models are used throughout the entire product lifecycle, including service and after-market:

> *"[The models] should be maintained throughout the product and after the product as well, if there are any changes to be done with the product through aftermarket sales."* — Int11 (space / industry / innovation)

This obligation in certain sectors to maintain and ensure the possibility to access models for long periods of time imposes interesting challenges, which we describe in more detail as part of RQ3 (see Section 5.3.2).

Several of our interviewees report that domain-specific languages are used in different areas that roughly correspond to different lifecycle phases, including, but not limited to requirements, state machines (belonging to design), and tests (belonging to validation and verification). Systems engineering organizations also involve all engineering disciplines throughout the development lifecycles as described above. Function models in MATLAB/Simulink (The MathWorks, Inc, 2023) are also very common, both for code generation and for analysis.

In some cases, standard languages such as SysML are used in different parts of the lifecycle. In some cases, they are augmented, for example, to capture specific requirements patterns as language constructs:

> *"We almost have a domain-specific language, you could say, since we have extended SysML and requirement patterns so that we can capture a lot of the specific information that we need there. And this continues to grow as we more and more require additional aspects."* — Int7 (automotive tier-1 supplier)

Interestingly, this interviewee's organization introduced architectural models not directly with the adoption of MBE. They originally started with function and unit models in MATLAB, then added capturing higher-level architectural aspects based on SysML and UML, and, after using a controlled natural language for specifying more formal requirements, now start to also apply models in the requirements phases.

This bottom-up approach is in contrast to what many of our interviewees from less mature organizations reported, who currently tend to focus on models for the design part of the development lifecycle. In particular, they focus on software and systems architecture, mostly with a low level of formality. In the future, they intend to investigate more detailed models for low-level aspects in later lifecycle phases for purposes like simulation and code generation — thereby following rather a top-down approach for the MBE adoption. One interviewee also mentioned that new model kinds are first explored in internal development and research projects before being applied in customer projects.

This also addresses proposition P6 from Table 1.

#### 5.2.4. Drivers of choosing a specific modeling language

While our interviewees mentioned a whole host of different modeling languages and the tools they use to create and maintain the models, there were fewer concrete drivers for the selection of these languages and, to a lesser degree, tools. In some cases, there just is not a lot of choice: One interviewee mentioned the language KBL (VDA, 2014), a language that allows modeling wiring harnesses. There do not seem to be any alternatives for this particular domain.

An additional strong driver is to use standards: UML, SysML and MATLAB/Simulink all fall under this category and several interviewees mentioned that these languages are used since they provide the benefits of standards such as a good knowledge base amongst stakeholders and a variety of tools to choose from. Standards are also interesting since they enable reuse across projects and make an organization independent of a specific toolchain.

Another kind of standardization also plays a role: Specific formats are used as standardized exchange formats between organizations, supporting proposition P7 (see Table 1). Our interviewees specifically mentioned AUTOSAR in this capacity as well as the Requirements Interchange Format (ReqIF) (Object Management Group, 2023a). Such modeling languages allow suppliers and OEMs/integrators to exchange information without the need to worry that information is lost or that tools cannot read them. As mentioned before, this is not always guaranteed for SysML or UML as different tool vendors implement the languages differently.

However, using such languages can also have disadvantages, as one of our interviewee reports:

> *"One of the problems of AUTOSAR is that you need to decide early on to either use the 'classic' variant, which targets microcontrollers, or the 'adaptive' variant which runs on a microprocessor. [...] I consider this a small weakness. UML has an advantage there since it is possible to start modeling the software without making this decision yet."* — Int4 (automotive tool vendor / consulting)

In this case, the disadvantage of having to make a rather consequential design decision early is a driver to switch to a different, less constraining modeling language.

One of the interviewees from a tool vendor also report on a customer that specifically chose the combination of modeling language and tooling to enforce constraints:

> *"[Our customers] have a guideline saying that you cannot use entry/exit actions on states. [...] And then they have implemented their own validation rules in the tool that will let the user quickly know if they violate that rule. And I think they look at that as one important benefit of model-based development, because with modeling it's kind of possible and quite easy to enforce such organization-specific constraints or industry-specific constraints that you want everyone in your organization to adhere to."* — FG (multi-sector tool vendor / consulting 1)

Finally, one driver that several interviewees report on is to select languages that allow to directly execute models. While there are some standardized languages for these purposes, including MATLAB/Simulink to a degree, this is also a driver for using DSLs:

> *"And actually for [models that can be executed] we tend to use domain-specific languages."* — Int9 (printing)

#### 5.2.5. Tooling

When it comes to the tooling used, our interviewees report a spectrum of solutions. On the one end of the spectrum, there are lots of informal tools: Diagramming tools, presentation tools, and mindmapping tools, for instance. In many cases in which such tools are used, the notations are similar to UML. But since such tools do not enforce syntactical correctness, diagrams are often not according to the specification. In general, however, modeling tools need to capture the semantics of the modeling language.

On the other end of the spectrum are highly integrated toolchains for end-to-end modeling. Such modeling toolchains can consist of "a plethora of different modeling environments", as one of our interviewees put it. In these cases, our interviewees report that they need to develop their own tools since some of it cannot be bought, especially when using highly specific or proprietary domain-specific languages. It is also possible that a company uses separate toolchains for different aspects (e.g., specification and simulation) and with differing levels of maturity.

In general, the different MBE tools for the different modeling languages in such an environment need to be integrated. However, one of our interviewees describes that integration as being ad hoc rather than designed:

> *"From my point of view, people always try to solve [the integration of MBE tools into a heterogeneous development landscape] a bit ad hoc. That means that the people who are responsible for the respective (sub-)disciplines look at how this is integrated accordingly. I don't yet know of any PLM approach where these models have been well integrated. It's always a question of whether top-down or bottom-up; sometimes I don't know whether top-down is always the right approach. As a rule, it should always be a mixture, probably. In practice, it's often more top-down. And that's not always easy, because the individual disciplines differ and the individual development teams have different priorities. Otherwise, you would have a relatively long requirements list if you tried to set something up top-down. That's why I always believe that you should rather proceed bottom-up, and then try to tackle things piece by piece."* — Int1 (corporation A)

The interviewee also mentions that customization is usually necessary for the individual tools since it is rare that languages are used 'as is'.

One significant challenge in integrated tool chains is the exchange of information between the tools. One interviewee mentions exchange formats, but these can have limitations since they might not cover the whole expressiveness of the involved languages. A complementary option is therefore to constrain the languages, for example, by using defined building blocks to make it easier to integrate across modeling languages.

A different approach to integration is championed by another interviewee:

> *"All tools that are now being added to the toolchain should be OSLC-capable so that they can somehow be linked. So that you can really achieve a complete linking of the tools, so that you at least have traceability, so that you can exchange certain data."* — Int7 (automotive tier-1 supplier)

Apart from its capabilities to support traceability and information exchange, the interviewee also mentions that using Open Services for Lifecycle Collaboration (OSLC) (OASIS Open, 2023) allows to exchange the tools in the toolchain at any point in time, thus increasing flexibility and reducing vendor lock-in.

Finally, one of the interviewees from a tool vendor mentions that they make large investments in documentation, courses, and examples:

> *"There's a whole part of the organization (about 12 people) maintaining [our tool] including language design and that kind of stuff. But there is also a group of seven people that develops course materials and modeling courses and uses of [our tool], test or example projects."* — Int3 (multi-sector tool vendor / consulting 2)

### 5.3. RQ3: What are practitioners' challenges and needs regarding the alignment of MBE with development processes?

Our interviewees report a number of challenges and corresponding needs with respect to MBE and processes. Whereas some of these challenges are well-known from existing studies, we describe them nevertheless as our data adds new perspectives or new depth to existing work.

As discussed in Section 3.1, we deliberately did not use the propositions derived from the literature to discuss the practitioners' needs and challenges in an exploratory manner.

#### 5.3.1. Tool-related challenges

Challenges relating to modeling tools have been reported many times in existing work, for example, in Hutchinson et al. (2011a) and Whittle et al. (2013, 2017).

*Tool zoos and interoperability.* Companies applying MBE typically have various modeling tools to cover different types of models, abstraction levels, or model purposes. This combination of tools poses problems, as models cannot easily be exchanged between the tools, usually due to lack of defined exchange formats or other interoperability solutions.

> *"Well, on the one hand it's the tools that are sometimes not compatible among each other. Then, there are different standards, like the AUTOSAR standard."* — Int7 (automotive tier-1 supplier)

> *"There's a lack of standardization and rigor in the various tools. They are appropriate for their narrow use, so to say. Simulink is good at generating code. But it's not clear how to relate Simulink to other downstream tools or related tools for software development."* — Int6 (automotive OEM)

Partially, this issue also stems from different parts of a company building custom solutions, which then need to be integrated when models should be connected across these parts:

> *"There are very many customers who are using very different processes — also many in-house solutions, which they built themselves and which do not have standard exchange formats. And, therefore, they often buy single products from us and integrate them in their own tool and process landscape."* — Int5 (automotive tool vendor / consulting)

Finally, the effort required to integrate or customize tools leads to a trade-off between doing so and building tools in-house:

> *"So let's say take an example: Architecting tools, right? You can find lots of them, like SysML tools. You can probably come up with some examples. You could buy one and then what you for sure have to do, you have to customize it to your needs. [..] The other option is that, or one other option is that you actually develop exactly what you want. For example, DSLs using Jetbrains MPS. And yeah, we are for this particular example, we are actually currently having discussion of what to choose."* — Int9 (printing)

*Challenging model management.* Several interviewees mention the challenging nature of models when it comes to managing them over time. The large number of models causes problems when trying to find information:

> *"We have such an enormous amount of models stored in many different places that we more or less created a very big chaos. That doesn't help. So: How to make sure that people can find models? How to make sure that they are accessible for those who need them? Yeah, that's also not trivial, and we also partly miss infrastructure to do proper model lifecycle management."* — Int9 (printing)

However, it also applies to the agile way of working. Specifically, iterative work in a cross-functional fashion makes it necessary to merge model versions more often compared to a traditional plan-driven process. Similarly, building models in a modular fashion needs to be explored:

> *"[...] since you have many people working in parallel on many features, it is very important to be able to compare and merge models. That's usually a difference when it comes to text files, they are easy to compare and merge. Model files are not always necessarily easy. There can be challenges involved in merging/changes done by many by different people."* — FG (multi-sector tool vendor / consulting 1)

> *"Since years people have been working also on languages that allow for this modularity in the modeling language itself. So you can have a larger single model but focus on certain aspects in that model."* — Int8 (printing)

*Suitable tool environments.* In modern software and systems engineering, individual modeling tools are not disconnected from the general way of working. Therefore, interviewees pointed out that models and modeling tools need to be supported by general-purpose software engineering tools, just as in regular, code-based development:

> *"If you do not have [automation, in particular translation from models to code and an automated build pipeline], then you cannot scale up with model-driven-ness, is my experience. You need to have the full stack on software engineering tools surround."* — Int3 (multi-sector tool vendor / consulting 2)

*Tool users vs. Tool vendors.* In our sample, we have both interviewees working at tool vendors and on client side, that is, tool users. As a result, several statements relate to the dynamics between tool vendors and users. Two interviewees mention that tool vendors do not always know how their tools are used, either because the vendors simply lack awareness, or because the customers do not want to share their way of using a tool:

*"A take away is that, I mean, we sit a bit in the dark, right? We see, but what we see is either when we meet directly with customers or through the support cases, right. And we sometimes see really strange things coming in via support. [..] So it's really hard to say what and how the customers are using the tools."* — FG (multi-sector tool vendor / consulting 1)

In a slightly different direction, one interviewee stated that tool vendors might not be interested in standardization of data or exchange formats, as this potentially affects their position in a software ecosystem:

*"The problem is that all the different point tools, they have their own business case and ecosystem. But it's not in their interest to invite others into that ecosystem. They want to keep the cake for themselves, so to say. So there's this tension between standardization, that is, in the interest of users and development companies on the one hand, and the tool vendors on the other hand."* — Int6 (automotive OEM)

An interviewee working at a tool vendor added to this aspect that there is a fear on customer side to become locked in on a specific tool, and therefore customers look for open data formats:

*"Because one concern maybe [which] many customers have, at least initially, is that if I start using a certain modeling tool, don't I become locked in to that vendor? And that this particular modeling tool, I mean how easy will it be in the future to take my models and go to something else? That's always a concern that you may not have as much as you if you're just coding."* — FG (multi-sector tool vendor / consulting 1)

*Money vs. Innovation.* As tool integration is challenging and as modeling and MBE are considered heavyweight, tool vendors would benefit from showing example cases of successful integrations across tools of different vendors. However, lacking adequate licenses prevents such demos:

*"Currently, the problem is that you have to write to every single tool vendor and ask 'I would need a license'. The interoperability between the tools is not given, since you do not know yourself what you are doing."* — Int5 (automotive tool vendor / consulting)

Similarly, in industrial use, successful integrations might not work across the value chain, as subcontractors might not have the same tool licenses as the OEMs (or other contractors):

*"Not every supplier has the tool licenses. For example, from Polarion requirement PDF documents are generated and sent by mail."* — Int2 (electronics & tool vendor)

In a similar direction, the push for a direct return on investment prevents innovating when new tools are needed. That is, economic discussions prevent that tools and other modeling solutions can be trialled.

*"If you, at some point, cannot work anymore, because it is only about licenses, for example. I often see this general trend that everything needs to grow, everything needs to get bigger, always stay economical. That you [don't] say 'We will now try something, which maybe does not directly pay off, but instead works and might become established [..]'."* — Int5 (automotive tool vendor / consulting)

### 5.3.2. Long-term model and tool maintenance

As many artifacts in software and systems engineering, models need to be maintained over time. Depending on the nature of the models, this also implies that the tool environment needs to be maintained so that models remain readable and potentially editable over time.

*Model and tool maintenance requirements.* In regulated domains, suppliers and OEMs face requirements for long maintenance of models. This, in turn, requires long lifetime for modeling tools, as model file formats are not always interoperable or as tools might offer specific analysis capabilities. Both aspects are considered challenging by some of our interviewees:

*"There is always a second challenge: the long-time [archiving of information]. [..] I think we only just start asking the question how this looks like after 30 years."* — Int1 (corporation A)

*"This is the big challenge, because as suppliers we have times of up to 20 years where we have to maintain things, so that you can still use them. If that always works is another question."* — Int7 (automotive tier-1 supplier)

*"[Some customers in regulated industry sectors] have requirements to be able to recreate old environments so that even 25 or 30 years after they have shipped something they need to make sure that those tools that were used back then still are available, still can load these models and generate code from them. So that they can patch things and work with even very old things."* — FG (multi-sector tool vendor / consulting 1)

*Risk of outdated models.* In addition to regulations, models need to be maintained as they otherwise become unreliable and therefore lose their purpose. Interviewees face the continuous challenge of keeping models up to date. In particular, sometimes changes are made in derived artifacts, such as code, without updating the models. This then causes the models to slowly become outdated until there is no longer a connection with the derived artifacts:

*"[...] when people start working with the code, they changed some small things, and if they didn't also update the model, the model was not useful because it was incorrect. So you couldn't trust the model after you have done a few changes in the code."* — Int11 (space / industry / innovation)

Related to the process, one interviewee stated that the danger lies in having models disconnected from the development process:

*"If you are not able to connect the modeling to let's say the development process itself by the artifacts themselves, then they get outdated. You have the same problem as with documentation in general."* — Int8 (printing)

### 5.3.3. External pressure: Time, complexity, and criticality

Companies are under constant time pressure and try to take shortcuts. Also, complexity and criticality are ever increasing, leading to models as the only solution:

*"On the one hand there is agility. On the other hand there is an increasing development of safety-critical functions [..]. OEMs increasingly require faster development cycles."* — Int7 (automotive tier-1 supplier)

*"The challenge is essentially [..]: Increasing complexity and decreasing time to market."* — Int4 (automotive tool vendor / consulting)

In some cases, the needs are not strong enough, yet, for modeling to be adopted:

*"There are many things that can be done, where I know that, in practise, they are not very much automated. [..] There are many [research-based] solutions, but not yet many industrial tools. Simply because the need in industry is not yet so strong. The pain is not yet strong enough [..]."* — Int4 (automotive tool vendor / consulting)

Instead of applying modeling, and doing so correctly, a common approach is to use models and modeling tools in an incorrect way, for example, by taking shortcuts. Later, these shortcuts cause even more effort:

> *"There again the issue is time. Currently, everything is one large server, where you could distribute, maintain everything, but for that we lack the time. Therefore, you only model and model."* — Int5 (automotive tool vendor / consulting)

> *"Much easier to use what you already have. Even if it's not the best fit, because if you can customize it, you can take it and get productive with it anyway."* — FG (multi-sector tool vendor / consulting 1)

#### 5.3.4. Resistance to modeling

There are various resistances to adoption of models or change in general. We discuss the different resistances our interviewees reported in the following paragraphs.

*Resistance to top-down change.* Changes are often imposed by management decisions, for example, to change from one process model to another. This can cause strong resistance among engineers, especially when they feel that their voices have not been heard, or their specifics not considered.

> *"Well, if it is a push, especially if it is a push from management, that does not necessarily work, and I know cases and they are not from this company, but from other companies where I've been, where that has worked out in a extremely bad way."* — Int9 (printing)

*Resistance to external change.* Changes can also be introduced externally, for example, through new trends, tools, or standards. These types of changes can cause resistance within a company, as the company culture and established ways of working are not in line with the proposed changes.

> *"'Not invented here' is a strong [psychological resistance factor]."* — Int6 (automotive OEM)

> *"It is change on the one hand, that applies for all large and established industries: Change takes time and there is a certain restraint and you know your way, how you used to do it before, how you did it up until now. It all works OK. [..] But yes, if the pain is not strong enough, then the established ways of working remain the same."* — Int4 (automotive tool vendor / consulting)

*Resistance to change between companies/units.* A specific form of resistance to external change are resistances to changes introduced by collaborators (i.e., by OEMs and suppliers) and by other units/departments in the same company. Here, organizations that cooperate along the value chain have to align on responsibilities, ways of working, and information exchange. In case this alignment requires adaptation, there is the tendency within companies to try resisting and instead requiring the counterpart to change:

> *"'Whose fault is it, when it doesn't work?', instead of 'How do we solve the problem, so that it works?'. [..] And yes, the big-picture person, who basically knows the entire car on customer side, does not exist anymore for the large OEMs."* — Int5 (automotive tool vendor / consulting)

> *"I see a very large challenge in improving the interfaces between the individual departments. There is always the issue that everyone [..] thinks their unit is currently the most important one. That you basically always have problems between the units to exchange data or finding the one to blame."* — Int5 (automotive tool vendor / consulting)

*Resistance from standardization bodies.* Whereas models can be beneficial in showing compliance to standards and/or regulations, standardization bodies are slow to change and accept models as evidence. Instead, they tend to require classical textual documents. This therefore hampers model adoption at the companies, as they cannot directly be used for audits.

> *"So I think there's [..] still, let's say, a challenge in discussions with regulatory bodies to get models accepted for at least some form or some part of the tests."* — Int9 (printing)

*Personal resistance to change.* Finally, individuals tend to resist changes for various reasons. Our interviewees mention several reasons for this resistance, for example, increased workload for one role for the benefit of others, resistance to acquire new skills, or varying performance indicators for individuals.

> *"Up to very human things, where you have someone saying 'Doesn't matter, I will retire in 1.5 years — I don't want to do something new'."* — Int1 (corporation A)

> *"Getting the individual engineers and domains to accept some inconvenience for the better of the whole. So even if my favorite editor is X I should use editor Y if it helps the overall integration and optimization. But optimizing the whole is difficult when each of the domain, for example, have their own key performance indices and their own instructions on what to optimize on."* — Int6 (automotive OEM)

#### 5.3.5. Modeling education and skills

Several challenges relate to educating engineers in modeling, and the skills they acquire at university and throughout their careers. We discuss these challenges in the following paragraphs.

*Lack of modeling skills for domain experts.* For most engineering tasks, domain expertise is required. However, many domain experts are not modeling experts and lack the skills to create high-quality models. Educating domain experts in modeling and onboarding engineers on existing modeling methods is therefore a challenge.

> *"How do I create a modeling process so that I can relatively easy onboard people, especially also when you have new people. That it does not have to be modeling experts, but so that domain experts can relatively easy create these models."* — Int1 (corporation A)

> *"It's very seldom that you have somebody that is a domain expert AND is good in modeling."* — Int3 (multi-sector tool vendor / consulting 2)

> *"I have given up the hope that there will ever be someone who understands everything. But that would be my dream: Having someone who combines all this [knowledge about different domains]."* — Int5 (automotive tool vendor / consulting)

*Lack of real-world modeling skills.* Even if engineers have the right background and education to be knowledgeable in modeling, they might lack skills to do so in industrial, real-world scenarios. That is, university knowledge might not prepare them to create models of industry-scale systems. Instead, knowledge is usually acquired and practiced on toy problems in university.

> *"Schools teach techniques on how to code something, or how to use [...] object-oriented design, how [a] compiler works. But it doesn't really tell you how a project works and how... What is a good design? What's a good requirement? How should you test things and stuff like that? And so I think the need for education in proper projects is fairly huge after someone who has ended an engineering education at school."* — Int11 (space / industry / innovation)

> *"I think the biggest challenge is that people cannot model. What I mean by that, is that they can do that unconsciously. Very good at that actually. But they don't have a clue how to make it explicit."* — Int3 (multi-sector tool vendor / consulting 2)

*Lack of knowledge for specifying understandable models.* In an industrial setting, most models need to be read, and possibly modified, by engineers different from the author who created the model. As such, they need to be specified in an understandable way that allows for reuse. The knowledge to do so is lacking, according to one interviewee.

> *"How do you get to people along? How do you make sure that the models are considered trustworthy and that especially if you have a different person creating the model than the one who's supposed to be using it. How to decide on the maturity of a model? There are a lot of questions."* — Int9 (printing)

> *"Then you need to go a bit further than just making the model for yourself. You have to make sure that this model is also usable for other people and that other people can understand this model. And they are not necessarily interested in how this model is, let's say, implemented. [..] No, these other people, they just want to know what the parameters that they can play with mean, so that they can actually change the values of those parameters and understand the outcome of the model, what that means. So that's a different kind of user."* — Int9 (printing)

*5.3.6. Uniform way of working*

Modeling is relatively heavyweight, as it requires introducing new tools, and adapting methods and processes. As such, several challenges relate to how a uniform way of working can be achieved.

*Too much tool choice.* Two interviewees mention that the variety of tools is large. Therefore, there is a risk of introducing too much freedom in choosing tools or too many tools at the same time.

> *"I think [being less strict in prescribing modeling tools] imposes challenges, mainly. In the sense that we are developing multiple parallel development pipelines. On the other hand, the resource that is mostly constrained is engineering competence and engineering bandwidth. So if we allow a little bit of inconvenience in the tool chains in order to make the most of the engineers, maybe it's worth it."* — Int6 (automotive OEM)

> *"Especially in software development I think that it is a challenge not to have too many environments in which you work."* — Int8 (printing)

*Aligning way of working takes time.* When using models heavily, many changes need to be introduced that affect processes, methods, but also the mindset of engineers. Making these changes takes a lot of time, according to one of our interviewees:

> *"Then we are talking about the 'how', but that immediately gives a million of options, let's say, of how you could do it. And everybody, of course, has an opinion about that. So the question is, what is then the... What are then the methods that you want to...? We all are sort of agreeing to that are the right way to go. And that is just a very yeah, time-consuming process."* — Int9 (printing)

> *"And it is also important that we do that ourselves, because you need to know, to a large extent you need to have the knowledge of existing ways of working and a domain. So it's about printers, and you need to understand sufficiently of that to be able to know what works — and what does not work. Or at least to have some initial idea about that."* — Int9 (printing)

*Automation requires harmonization.* Automation is often the goal when introducing MBE. However, to successfully automate activities using models, these models need to be precise. Achieving this precision in order to automate requires making processes and methods uniform and harmonized. This was stated by one interviewee:

> *"I guess the challenge and opportunity I had is to increase the amount of automation in the engineering processes. And that requires more alignment. Because of automation and corresponding machine readability of your content, then you must be more precise. And the good thing with it is that correctness will increase. And many manual steps might be omitted."* — Int6 (automotive OEM)

*5.3.7. Unclear motivation and no pressing problems*

As mentioned earlier, some interviewees see modeling and MBE as the only way to cope with increasing competition, need for speed, and safety criticality. However, some companies do not yet see pressing problems that would motivate the use of models. Instead, they see a larger need for improvement in other areas:

> *"And you know it's of course it's worth paying for a good tool if it helps you. But if you're sort of slightly bit skeptical on what you get, it's hard to motivate that and you don't really feel the pull to get to it. That seems like the sort of biggest problem to solve right now, I suppose."* — Int10 (space / industry / innovation)

> *"Of course, it should invest in tooling that makes you increase your velocity so they can make more and better stuff. But I'm not sure if... I mean that's probably my notion that sort of modeling tools is not the first thing I look at. But it's rather other things, sort of you, know, better testing tools, you know, better tools for implementations, that's probably where I feel there we lack the most."* — Int10 (space / industry / innovation)

> *"It seems like the way how we want to introduce model-based is... It's not easy to see that helps this process in a certain way, so it's more... It feels like an add-on than solving these type of problems."* — Int8 (printing)

## 6. Discussion

Our interviews yield many different topics for discussion, but many have already been discussed in the literature (e.g., Baker et al. (2005), Mohagheghi and Dehlen (2008), Mohagheghi et al. (2009, 2013), Whittle et al. (2013, 2017), Liebel et al. (2019) and Amorim et al. (2019) report on the lack of integrated tools or challenges in tool chain integrations). We therefore focus on those aspects we identified in our results that relate to our propositions (see Table 1), which have not been treated thoroughly in previous work, or where our data adds more depth.

### 6.1. Tailoring tools to processes and methods or vice versa

Proposition P1 in Table 1 addresses the question whether tools are tailored toward processes and methods or vice versa. In their literature review, Hebig and Bendraou (2014) indicate that approximately half of the approaches that introduce MBE into an organization require tailoring the engineering process. Amorim et al. (2019) report that tool vendors often enforce a tool-specific workflow with their tools in the context of Model-Based Systems Engineering, enforcing the tailoring of processes when introducing the tool. Whittle et al. (2013, 2017) investigate whether the most common practice is to tailor tools to an existing engineering process, to tailor the process to the tool, or to develop a dedicated tool that naturally fits the process. They state that the latter approach is the most promising one and that some tool vendors actively forbid tailoring a tool for business reasons.

Regarding this aspect, our interviewees answered heterogeneously (see Section 5.1.4). In this context, we observe a relationship between the size of the companies and the resulting influence they might have on tool vendors. In cases where the interviewees answered that the tools have no impact on the process at all, the corresponding companies are large corporations that might have a certain level of influence on the tool vendors. For example, Int1 from corporation A and Int6 from the automotive OEM reported this setting. In contrast, in cases where the interviewees answered that the processes are altered if absolutely required, the corresponding companies are also large but might not have as much influence on tool vendors. For example, Int7 from the automotive tier-1 supplier and Int8 from the printing company mentioned these circumstances. This is in line with our industrial experiences, where we observed that large corporations have enough money to order tool adaptions or, with a large user basis, threaten to switch to a different tool by a competitive vendor or to open-source software.

Regarding the application of self-developed tools, the interviewees whose organizations sometimes tailor their process to the tools mentioned this approach in the context of applying domain-specific modeling languages as one of many building blocks in an overall tool chain (see Sections 5.2.3 and 5.2.4). For example, Int7 from the automotive tier-1 supplier mentions a DSL for requirements in a self-developed tool together with model transformations synchronizing with a SysML extension, and Int8 and Int9 from the printing company mention an executable DSL for state engines in a self-developed tool. In contrast, Int3 from multi-sector tool vendor / consulting 2, who build their own tooling ecosystem for the specification of tax laws, mentions a plethora of self-developed DSL tools based on JetBrains MPS (JetBrains s.r.o, 2023).

Several challenges relate to P1 and might explain the obtained results. The fact that most companies use several tools (see Section 5.3.1) leads to interoperability challenges. However, it might also mean that tools are used for more specialized purposes, requiring less adaptation to the process, or vice versa. Another important aspect is that companies prioritize a direct return-on-investment over the potential to innovate (see Section 5.3.1). This can result in re-use of existing tools for purposes they are not suited for, which can require either adapting the tool or the process.

Regarding **P1**, we find evidence for the statement that either the process or the tools have to be adapted to fit to each other. More concretely, we observe that whether the tool or the process is adapted when MBE is introduced depends on the size of the organization and its influence on the tool vendor. Larger organizations have the ability to influence tool development or buy customizations that adapts tools to their process.

### 6.2. Tailoring processes to MBE

As captured in proposition P2 (see Table 1), Mohagheghi and Dehlen (2008) state that development processes are not tailored to MBE and MBE does not make any assumptions on the development process. While there is some relationship to P1, the premise is slightly different: while Whittle et al. (2013, 2017) argue that the tailoring centers around the tools, Mohagheghi and Dehlen (2008) state that MBE can be applied in any development process and that the process does not need to change to accommodate it.

Our participants reported that, indeed, process descriptions do not mention MBE explicitly, partially because they are designed this way (as stated by, e.g., Int7, see Section 5.1.2). The respondents also emphasize the distinction between process description which should be agnostic to concrete tools and techniques and the development environment or the method descriptions which contain these details. These two kinds of descriptions have different lifecycles.

Challenges: While not mentioned by our interviewees, we believe several of the reported challenges relate to this proposition. To address interoperability issues arising from the use of multiple tools, as well as difficulties in model management (see Section 5.3.1), MBE and tool-specific activities might have to be lifted to process descriptions. That is, concrete steps to make models interoperable or maintain them over time might have to be described at process level. Similarly, in order to achieve automation using MBE, different process steps and activities might have to harmonized (see Section 5.3.6), again requiring more details regarding MBE in process descriptions.

Regarding **P2**, we find evidence for the statement by Mohagheghi and Dehlen (2008) that development processes are not tailored to MBE and MBE does not make any assumptions on the development process. Beyond that, our data yields that process-mature companies deliberately describe their development processes in a coarse-grained and tool-agnostic manner and complement these descriptions with tool-specific modeling method guidelines, which can be applied in different process steps.

### 6.3. Agile development processes and MBE

Bucchiarone et al. (2020) argue that agility and MBE impede each other. Kuhrmann et al. (2022) formulate the same hypothesis but refute it, and also Burden et al. (2014) yield opposite results. Our interviewees argue for the latter side of that coin (see Section 5.1.5), namely that there are no contradictions between an agile way of working and MBE. Specifically, several interviewees note that MBE is different compared to traditional development in the sense that different artifacts are produced, that might be at a different level of abstraction. However, this does not affect the way of working, and therefore it does not conflict with agility. That is, models should be treated no differently than other software engineering artifacts in any software development process.

One interviewee went as far as stating that the only way to be agile in a given domain is to apply MBE, as it enables a faster way of making changes. This is an interesting take compared to Bucchiarone et al. (2020), in the sense that it highlights the possibility for MBE to push the boundaries of agility beyond its current limitations. For instance, challenges of scaling agile development to large systems, for example, reported in Dikert et al. (2016), Dingsøyr and Moe (2013), Conboy and Carroll (2019) and Kasauli et al. (2021), could be addressed to some extent by introduction of MBE.

However and in line with the findings of Burden et al. (2014), we find several statements indicating that the path to using MBE in an agile development process successfully is not straightforward. First, using MBE successfully requires standard software engineering tooling that surrounds modeling, such as version control (see Section 5.3.1). While such tooling is mature for a code-focused development process, this is not the case for MBE. Second, several interviewees report that time, criticality of the developed components, and system complexity lead to shortcuts when using MBE, for example, by re-using tools for purposes they are not suited for (see Section 5.3.3), or due to lack of knowledge how to make models re-usable (see Section 5.3.5). Shortcuts can increase the risk for outdated models, which is already challenging (see Section 5.3.2). This challenge might manifest particularly strong in an agile context, as every iteration/sprint should lead to working software. Finally, many agile methods advocate for empowered teams that can choose, amongst other things, their own tools and methods. This conflicts with requirements on the harmonization in MBE (see Section 5.3.6).

We find no direct evidence for **P3**, that is, that agile development processes and MBE impede each other. However, a lack of suitable tools and methods, as well as a lack of knowledge, might impede an agile way of working using MBE.

### 6.4. Drivers for MBE

Our results provide evidence for Whittle et al. (2013, 2017)'s results that external regulations and industrial de facto standards foster adoption of MBE (cf. P4 in Table 1). In particular, safety standards such as ISO 26262 and process standards such as Automotive SPICE are drivers for the adoption of MBE. Our respondents state that they make it possible to track information across the development lifecycle (a question of traceability, see, for example, Steghöfer et al. (2021)) and are critical in verification and simulation as seen, e.g., in the automotive industry (Mayer and Spieckermann, 2010; Weissnegger et al., 2016) or for medical devices (Lee et al., 2006).

Overall, many of the benefits of MBE mentioned by participants in Section 5.1.6 are particularly relevant for safety-critical systems. This view is supported by Maurya and Kumar (2020) who find a large number of model-based techniques used to ensure reliability of safety-critical systems. Likewise, Bolbot et al. (2019) describe a number of methods to assure safety in cyber–physical systems which are based on models (e.g., Failure Logic modeling or System-Theoretic Process Analysis) and are used to show that systems adhere to safety standards.

Our participants also refer to de facto standards when discussing their choice of modeling language. A prominent example is AUTOSAR which is a de facto standard in the automotive industry. But even the choice of MATLAB/Simulink as mentioned in Section 5.2.3 is based on the same rationale. Likewise, UML and SysML are often chosen based on the availability of tools and their state as a de facto standard in the industry as discussed in Section 5.2.4.

However, there are a number of additional factors that are mentioned by our interviewees which they cite as important drivers to adopt MBE. This includes the growing complexity of the products and the need to support specific activities for which models are a suitable choice such as analysis and code generation. These factors have already been widely reported in the literature, for example, by Mohagheghi et al. (2009).

While several challenges slow down or hinder adoption of MBE, we consider a few of them particularly detrimental. First, while external regulations and standards are a reason to adopt MBE, as discussed above, these external factors also cause internal resistance (see Section 5.3.4). Second, focusing too much on a direct return on investment can hinder innovation and prevent adoption of MBE (see Section 5.3.1), as the return on investment for MBE is not always immediately evident. Third, domain experts might lack the necessary skills to adopt MBE, thus requiring costly training and/or cause resistance (see Section 5.3.5). Finally, many organizations might not (yet) face problems that directly motivate the use of MBE (see Section 5.3.7).

Regarding **P4**, a major driver for adoption of MBE are safety standards and process standards. In addition, industrial de facto standards influence the choice of modeling languages.

### *6.5. Different maturity levels and approaches for the institutionalization of MBE*

Proposition P5 in Table 1 states that the institutionalization of MBE through dedicated expert teams is a critical success factor. Based on the three different clusters of answers that we identified regarding the institutionalization of MBE and modeling methods (see Section 5.1.3), we identify different maturity levels regarding this aspect:

**Level 0: No approach** On the lowest maturity level, there is no approach at all to institutionalize MBE or a modeling method — none of our practitioners reported on that, indicating a higher maturity in our company sample.

**Level 1: Examples** Whittle et al. (2013, 2017) elicit from their interviewees that, in contrast to code-level examples and case studies, open-access and tool-specific model examples can rarely be found online. Consequently, on the next higher maturity level, there are well-proven examples accessible in the company that can guide the modeling activities or serve as inspiration for new modeling projects. For example, Int5 from the automotive tool vendor / consulting company as well as Int10 and Int11 from the space / industry / innovation company mention this approach.

**Level 2: Expert teams** Mohagheghi et al. (2009) propose to build expert teams as part of the company to provide support in the adoption of MBE. In this context, Baker et al. (2005) report on such teams being a success factor in the application of MBE, but they lack a description of how the teams provide support. Consequently, on the highest maturity level, there are dedicated modeling expert teams or departments. We observe that they seem to follow two different approaches, depending on the company culture:

- **Active modeling support** In the first approach, the expert teams provide dedicated and active modeling support in the particular development projects. For example, FG from multi-sector tool vendor / consulting 1 reporting about one of their customers as well as Int8 and Int9 from the printing company mention this approach. This approach might be followed in rather collaborative or agile companies.
- **Documentation and trainings** In the second approach, such expert teams comprehensively document modeling methods with, for example, modeling guidelines or handbooks and complement such documentation with offers for modeling trainings. For example, Int1 from corporation A, FG from multi-sector tool vendor / consulting 1 reporting about a different customer, Int3 from multi-sector tool vendor / consulting 2, Int6 from the automotive OEM, and Int7 from the automotive tier-1 supplier mention this approach. The distribution of answers from our interviewees indicates that this approach seems to be used in regulated sectors. We believe that this might be related to the fact that such sectors are in general heavy on documentation.

  Regarding trainings, Amorim et al. (2019) report that they are absolutely essential. Furthermore, their interviewees propose that domain-specific training examples should be created to help the engineers understand modeling concepts in their domain, and that technically prepared people should conduct the trainings instead of sales people from tool vendors. Hutchinson et al. (2011a, 2014) and Whittle et al. (2013, 2017) report that MBE requires dedicated effort for the corresponding trainings.

Another factor influencing which approach to follow on this maturity level could be scalability. That is, in large companies with many parallel active development projects, the documentation and training approach might scale better than the active modeling support approach. However, the latter one might be more effective during the solving of concrete modeling challenges due to the direct collaboration.

For P5, several challenges indeed motivate an increased use of expert teams or at least well-documented modeling examples. Regarding tool-specific challenges (see Section 5.3.1), the use of tool zoos, difficulties in model management, and long-term maintenance of models all require mature modeling support. Similarly, lack of real-world modeling skills, lack of modeling skills in domain experts, and lack of knowledge on how to create understandable models (see Section 5.3.5) call for sophisticated support. Finally, modeling support is a way to align the way of working over time, which relates to all three challenges in Section 5.3.6.

Regarding **P5**, we find evidence but also deviations from the statement that MBE should be supported by expert teams. To be more precise, we observe that the majority but not all companies in our sample apply expert teams to institutionalize MBE. However, the way how these teams provide their expertise can be distinguished into active support and passive documentation plus trainings. The companies that do not have expert teams at least share internally best-practice examples, whereas we did not observe a company that had no MBE institutionalization approach. We aggregate these observations to different maturity levels for the institutionalization of MBE.

### *6.6. Long-term model maintenance*

Proposition P6 states that the value of different kinds of models changes throughout the different phases of the development lifecycle. In our interview data, we find support for P6 regarding two dimensions, a purpose and a temporal dimension. In terms of purpose, different models are used at different points in time, for different reasons (see Sections 5.2.3 and 5.2.2). As such, the value of different models naturally relates to the purpose and its connection to the development lifecycle. For instance, an architectural model of the system under development is valuable early on to plan work and decide on quality attribute trade-offs. The same model can be used for change impact assessment during software and system maintenance.

The temporal dimension relates to the value of the model as the system evolves. That is, a model's value might decrease if not updated (see Section 5.3.2). Furthermore, changes in tooling might affect the value of a model in case it can no longer be read, or if new tools do no longer support certain tasks (such as code generation) (see Section 5.3.1). This aspect is particularly important in case of long maintenance times (see Section 5.3.1). Finally, external pressure might lead to shortcuts, thus decreasing the value of models (see Section 5.3.3).

Both the purpose for which individual models are used, and how they are maintained over time relate heavily to the development process. Specifically, for an agile development process, organizations need to decide whether models serve only a single purpose, or if they need to be created in a more general fashion so that they might be re-used for other purposes. Here, the Agile Manifesto advocates for maximizing "the work not done" (Beck et al., 2001). Similarly, for the temporal dimension, models might be treated as temporary artifacts that are discarded after use, similar to user stories. However, this clashes with requirements for maintenance and with known issues in maintaining product knowledge over time (Kasauli et al., 2021).

Regarding **P6**, we find evidence that the value of models changes across different phases of the development lifecycle, depending on their purpose and model maintenance over time.

### *6.7. Standardized exchange formats allow using models as boundary objects*

Forward and Lethbridge (2008) report that many of their practitioners complain that models cannot be exchanged between different tools, and Amorim et al. (2019) consequently emphasize the importance of import and export features. Mohagheghi et al. (2013) report on issues that emerge when tool vendors do not strictly adhere to actually standardized exchange formats, and Mohagheghi and Dehlen (2008) report that one motivation for MBE is the existence of standardized exchange formats.

In our study, many interviewees motivate the usage of certain modeling languages based on the possibilities to use them as exchange formats for importing and exporting data, which stems from the (de facto) standardization of some of these modeling languages (see Sections 5.1.6, 5.2.4 and 5.2.5). One of these languages is the AUTOSAR architecture description language, which originates from the research project EAST-EEA that ran from 2001 to 2004 (EAST-EEA Project Consortium, 2018). This language also defines a standardized exchange format, one of the main goals of the AUTOSAR initiative from its inception (Pagel and Brörkens, 2006). In fact, many exchange formats are outcomes of research projects (e.g., AMALTHEA data models from the projects AMALTHEA(4public) AMALTHEA(4public) Project Consortium, 2018 or the Functional Mock-up Interface standard from the MODELISAR project MODELISAR Project Consortium, 2017) that are very important for the industry and consequently one reason why such projects get funded. Particularly, exchange formats are one building block for the consolidation of tool chains, which we discuss in the upcoming subsection.

Liebel and Knauss (2023) argued that requirements models in a case company they investigated were used as boundary objects (Wohlrab et al., 2019), that is, artifacts that can be used to communicate and coordinate across different teams, disciplines, or even organizations, collectively described as "methodological islands". We argue that the use of languages that enables information exchange as described by our participants enables the use of models for information exchange and that the evidence we find in this study goes beyond the requirements models described in the case by Liebel and Knauss (2023). As described above, industrial de facto standards enable the exchange of information and collaboration across organizational boundaries. Current developments such as digital twins (see, e.g., VanDerHorn and Mahadevan (2021)) and asset administration shell (see, e.g., Ye et al. (2021)) – both evolutions of MBE – are specifically designed around such information exchange.

However, boundary objects as originally defined by Star and Griesemer (1989) and as used by Zaitsev et al. (2016) and Wohlrab et al. (2020) to describe collaboration in software engineering require more than the exchange of information: They also include a governance structure where the parties involved in the collaboration agree on a concrete way of maintaining and using the artifacts. Our interviewees did not mention such governance structures. It seems like they are using them mostly for unidirectional information exchange within a limited scope where concepts are clearly defined. This is a somewhat more reductive view of boundary objects and does not correspond fully to the understanding expressed by Liebel and Knauss (2023).

Indeed, establishing such governance structures in particular in situations where suppliers which are in competition need to collaborate in a product for an OEM remain an open question. Schmelter et al. (2023) address this by transferring the concepts of *information sharing management system*, in particular of "Trusted Information Communication Entities" and "Warning, Advice and Reporting Points" from ISO 27001 (ISO and IEC, 2022) to collaborative engineering. We believe that more such work is needed to transition from pure information exchange to the use of models as boundary objects on a broader scale.

In terms of challenges, several of our findings motivate the use of models as boundary objects. The existence of tool zoos and the risk of outdated models (see Section 5.3.1) would both benefit from defined boundary objects, including established governance structures. Similarly, defining artifacts across organizational boundaries could help to address a uniform way of working (i.e., the challenges that too much tool choice exists), and automation requires a harmonized way of working (see Section 5.3.6).

Regarding **P7**, we find evidence in our interviews that information exchange across organizational boundaries using standardized exchange formats plays an important role in MBE and models of different kinds are involved in this. However, we do not observe the governance structures associated with boundary objects.

### 6.8. Additional propositions

Our original propositions were derived from the literature and guided our research for RQ1 and RQ2. Thanks to the explorative nature of our study, our data yielded two new propositions that were not considered in related work and are hence not linked to our propositions.

#### 6.8.1. Tool users vs. Tool vendors

As reported in Section 5.3.1, our interviews yield different points of view contradicting each other when looking at companies that use MBE tools and when looking at companies that develop such tools.

On the one hand, tool vendors face the difficulty of anticipating how their tools are used. This is either due to the fact that they observe only at certain synchronization points that their tools are used in a different way than expected, or due to the fact that the tool users do not want to share their way of working to protect their intellectual property. We observed the latter issue also in different industrial settings. Furthermore, Whittle et al. (2013, 2017) report that some tool vendors do not open certain parts of their tools for customization due to business reasons, and Amorim et al. (2019) report that tool vendors often enforce tool-specific workflows with their tools.

On the other hand, tool users partially indicate a lack of trust in tool vendors. That is, our interviewees do not trust that the tools are open enough to be integrated in heterogeneous tool chains or are maintained for long enough, and hence look for tools supporting standardized exchange formats but also non-proprietary storage formats to be able to access the data if tool support is discontinued. In this context, Forward and Lethbridge (2008) report about tool users' lack of trust in the evolution of the modeling tools and in the long-term maintenance support of the tool vendors. A switch to open-source solutions would enable full control over such aspects (Đurković et al., 2008), but none of our interviewees mentioned the possibility of such a transition.

Thus, in our opinion, tool vendors and tool users should approach one another and establish a more trustful relationship. A more collaborative and agile development approach as well as sharing data could help in such an increase of trust.

> **P8**: The relationship between tool vendors and their customers is a key factor in whether tools are perceived as a challenge by practitioners.

#### 6.8.2. Influence of the modeling language design on processes or methods

In our analysis, we were struck by the statement made by Int4 and reported in Section 5.2.4 that a modeling language can enforce an early design decision. In the concrete case, Int4 stated that the use of AUTOSAR enforces specific design decisions very early on. AUTOSAR supports two variants: The classical and the adaptive variant (Fürst and Bechter, 2016). Since both variants have significant differences in terms of which aspects can be modeled (e.g., only Adaptive AUTOSAR supports Ethernet for communication, but only Classic AUTOSAR supports hard real-time constraints), the choice of concrete variant and therefore modeling language is dependent on many design choices that might not have been made when the first models need to be created. According to Int4, they therefore prefer UML as it allows focusing on hardware-independent software application functionality.

This shows that modeling language design can influence the process and when certain design decisions need to be made. Interestingly, in literature, the language itself is rarely the focus. Instead, the influence of tools on the process is discussed (see, e.g., Whittle et al. (2013, 2017)). There is very little work on how to choose an appropriate modeling language. Giraldo et al. (2021) is an exception to this. The authors describe several language quality aspects and include "suitability of the approach regarding the application domain". This might fit the problem discussed here, but does not explicitly include when design decisions need to be made. Other literature focuses on choosing a suitable notation (Meliá et al., 2016) or on the user experience of modeling with the language (Abrahão et al., 2017).

We believe that this aspect has not been a major concern in previous works since the situation where there is a choice between two different languages that offer different features and that impact the degrees of freedom in the design is rare. In most cases, we posit that engineers have no choice at all (e.g., because a standard or a supplier prescribes the language), or that possible choices are not considered (e.g., because of company policy, existing tooling, familiarity with a language).

> **P9**: The constraints imposed by a modeling language can have a direct impact on the freedom of engineers to make design decisions.

### 6.9. Answering the RQs

To answer RQ1 and RQ2, we regarded the seven propositions in Table 1, where P1 to P5 addressed RQ1 and P6 and P7 regarded RQ1. Since RQ3 was more exploratory, we did not use propositions to address it. In the following, we will summarize our findings and answer the research questions.

*RQ1: How is MBE aligned with the development process?* Overall, we did not find evidence that there is a misalignment between MBE and established development processes. Our respondents stated that MBE is often not mentioned in higher-level process descriptions (P2) and that MBE practices are successfully used in agile development processes (P3). The alignment of tools to the development process occurs in both directions, with the bargaining power of the organization towards the tool vendors being a decisive factor (P1). Safety and process standards are the main driver for adopting MBE (P4). Expert teams support the institutionalization of MBE practices, but there are widely varying degrees of how this is handled in practice (P5).

*RQ2: How and why are which model(ing) kinds, languages, and tools applied in different phases of the development process?* In general, many of our interviewees mention the usage of de facto standard modeling languages like UML and SysML due to tool availability and sector-specific languages like AUTOSAR due to standardized exchange formats. With regard to languages and tools, we observe that the value of different kinds of models and therefore also of languages and tools evolves during the development lifecycle (P6). One driver for selecting different types of languages is the use of models for information exchange across organizational boundaries (P7).

*RQ3: What are practitioners' challenges and needs regarding the alignment of MBE with development processes?* We report several challenges that relate to aligning MBE and development processes. The overall categories we find resonate well with existing work, for example, on tool-specific modeling challenges, resistance to adopt MBE, or education-related challenges. However, we add depth to these categories. For instance, the angle of distinguishing tool-related challenges of users and vendors of modeling tools is novel. Similarly, while existing work has already reported that MBE and agile development are not conflicting, our challenges allow for a nuanced view, by highlighting that time pressure, lack of sophisticated tools supporting MBE, and a lack of knowledge in creating understandable models can hinder an agile way of working.

## 7. Conclusion and future work

In this paper, we presented a qualitative multiple-case study on the alignment and interplay of MBE and the development processes its application is embedded in. We conducted the study with nine companies operating in heterogeneous industry sectors and collected data from 12 interviews with 14 persons. Our goal was to understand how MBE is anchored in the development processes and to investigate the social and organizational aspects that processes intrinsically entail. In this context, we complement related work that partially touches such aspects but does not focus on them.

Our data yields one confirmation of related work results: The value of certain models changes over time (cf. P6 in Table 1).

Moreover, our data partially confirmed some further related work results but added more details that were not considered by related studies:

1. Processes or tools have to be adapted to fit each other (cf. P1), but larger companies have more leverage and money to influence and steer tool adaptations than smaller companies.
2. The application of MBE and development processes are independent of each other (cf. P2), and process-mature companies deliberately exploit this proposition to separate tool-specific and thereby often updated modeling method guidelines from tool-agnostic and therefore long-living process descriptions.
3. Regulations and de facto standards push MBE (cf. P4), but industrial de facto standards also influence the choice of modeling languages.
4. Not every company applying MBE employs expert teams, but if they do, the expert teams support either actively in projects or passively through documentation and training (cf. P5).
5. Whereas exchange formats play an important role in MBE, the governance of the exchanged artifacts does not seem to be a focus of our respondents (cf. P7).

Finally, our data contradicts the proposition P3: Agile development processes and MBE does not impede each other for our company sample, although the way toward a successful combination is not always straightforward. In this context, one company even stated that Agile was the reason for introducing MBE.

Beyond that, we yielded new findings thanks to the exploratory nature of our study:

1. Tool vendors face the difficulty of anticipating how their tools are used, and tool users partially indicate a lack of trust in tool vendors — resulting in a dysfunctional mutual relationship.
2. The design of modeling languages can influence the processes or methods by requiring design decisions that are potentially enforced at the wrong point in time in certain process or method steps.

We propose that future work should explore how MBE integrates into contemporary development processes. As our findings indicate, various critical factors exist that do not directly relate to models and modeling notations themselves, but rather to organizational factors, support tools such as version control or static (model) analysis, and sociotechnical factors such as modeling education and training. Furthermore, as many decisions require trade-offs in practice, we encourage exploring these factors in a collaborative manner with industry.

## CRediT authorship contribution statement

**Jörg Holtmann:** Conceptualization, Methodology, Validation, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Grischa Liebel:** Conceptualization, Methodology, Validation, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Jan-Philipp Steghöfer:** Conceptualization, Methodology, Validation, Investigation, Data curation, Writing – original draft, Writing – review & editing, Funding acquisition.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Joerg Holtmann reports financial support was provided by Sweden's Innovation Agency. Jan-Philipp Steghoefer reports financial support was provided by Sweden's Innovation Agency. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The interviews, their transcripts, and the interviewee data is confidential. Nevertheless, we shared our interview guide, the codebook, the blank consent form that our interviewees signed to participate, and the blank mail that we used to invite potential interviewees for the study participation as part of our supplemental material Holtmann et al. (2023).

## Acknowledgments

Parts of this work were sponsored by Vinnova under grant number 2019-02382 as part of the ITEA4 project *BUMBLE*.

## References

Abrahão, S., Bourdeleau, F., Cheng, B., Kokaly, S., Paige, R., Stöerrle, H., Whittle, J., 2017. User experience for model-driven engineering: Challenges and future directions. In: Proceedings of the 2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS). IEEE, pp. 229–236. http://dx.doi.org/10.1109/MODELS.2017.5.

Agner, L.T.W., Soares, I.W., Stadzisz, P.C., Simão, J.M., 2013. A Brazilian survey on UML and model-driven practices for embedded software development. J. Syst. Softw. (JSS) 86 (4), 997–1005. http://dx.doi.org/10.1016/j.jss.2012.11.023.

AMALTHEA(4public) Project Consortium, 2018. ITEA impact story—AMALTHEA: From individual approaches to a widely accepted open platform. URL: https://itea4.org/project/impact-stream/amalthea-impact-story.html, accessed November 2023.

Amorim, T., Vogelsang, A., Pudlitz, F., Gersing, P., Philipps, J., 2019. Strategies and best practices for model-based systems engineering adoption in embedded systems industry. In: Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). pp. 203–212. http://dx.doi.org/10.1109/ICSE-SEIP.2019.00030.

Automotive SIG, VDA, 2023. Automotive SPICE. URL: https://www.automotivespice.com/, accessed November 2023.

AUTOSAR GbR, 2023. Automotive open system architecture (AUTOSAR). URL: https://www.autosar.org/, accessed November 2023.

Baker, P., Loh, S., Weil, F., 2005. Model-driven engineering in a large industrial context—motorola case study. In: Proceedings of the 8th International Conference on Model Driven Engineering Languages and Systems (MODELS). In: LNCS, vol. 3713, Springer, pp. 476–491. http://dx.doi.org/10.1007/11557432_36.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D., 2001. Principles behind the Agile Manifesto. URL: https://agilemanifesto.org/principles.html, accessed November 2023.

Bolbot, V., Theotokatos, G., Bujorianu, L.M., Boulougouris, E., Vassalos, D., 2019. Vulnerabilities and safety assurance methods in cyber-physical systems: A comprehensive review. Reliab. Eng. Syst. Saf. 182, 179–193. http://dx.doi.org/10.1016/j.ress.2018.09.004.

Bucchiarone, A., Cabot, J., Paige, R.F., Pierantonio, A., 2020. Grand challenges in model-driven engineering: an analysis of the state of the research. J. Softw. Syst. Model. (SoSyM) 19, 5–13. http://dx.doi.org/10.1007/s10270-019-00773-6.

Burden, H., Heldal, R., Whittle, J., 2014. Comparing and contrasting model-driven engineering at three large companies. In: Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). ACM, http://dx.doi.org/10.1145/2652524.2652527.

Candela, A.G., 2019. Exploring the function of member checking. Qual. Rep. 24 (3), 619–628. http://dx.doi.org/10.46743/2160-3715/2019.3726.

Conboy, K., Carroll, N., 2019. Implementing large-scale agile frameworks: challenges and recommendations. IEEE Softw. 36 (2), 44–50. http://dx.doi.org/10.1109/MS.2018.2884865.

Cruzes, D.S., Dyba, T., 2011. Recommended steps for thematic synthesis in software engineering. In: 2011 International Symposium on Empirical Software Engineering and Measurement. pp. 275–284. http://dx.doi.org/10.1109/ESEM.2011.36.

David, I., Aslam, K., Malavolta, I., Lago, P., 2023. Collaborative model-driven software engineering—A systematic survey of practices and needs in industry. J. Syst. Softw. (JSS) 199, http://dx.doi.org/10.1016/j.jss.2023.111626.

Dikert, K., Paasivaara, M., Lassenius, C., 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review. J. Syst. Softw. (JSS) 119, 87–108. http://dx.doi.org/10.1016/j.jss.2016.06.013.

Dingsøyr, T., Moe, N.B., 2013. Research challenges in large-scale agile software development. ACM SIGSOFT Softw. Eng. Not. 38 (5), 38–39. http://dx.doi.org/10.1145/2507288.2507322.

Dobing, B., Parsons, J., 2006. How UML is used. Commun. ACM 49 (5), 109–113. http://dx.doi.org/10.1145/1125944.1125949.

EAST-EEA Project Consortium, 2018. ITEA impact story—EAST-EEA: Paving the way towards revolutionary automotive software development. URL: https://itea4.org/project/impact-stream/east-eea-impact-story.html, accessed November 2023.

Etikan, I., Musa, S.A., Alkassim, R.S., 2016. Comparison of convenience sampling and purposive sampling. Amer. J. Theor. Appl. Statist. 5 (1), 1–4. http://dx.doi.org/10.11648/j.ajtas.20160501.11.

Forward, A., Lethbridge, T.C., 2008. Problems and opportunities for model-centric versus code-centric software development—A survey of software professionals. In: Proceedings of the 2008 International Workshop on Models in Software Engineering. ACM, pp. 27–32. http://dx.doi.org/10.1145/1370731.1370738.

Fürst, S., Bechter, M., 2016. AUTOSAR for connected and autonomous vehicles: The AUTOSAR adaptive platform. In: Proceedings of the 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W). IEEE, pp. 215–217. http://dx.doi.org/10.1109/DSN-W.2016.24.

Giraldo, F.D., Chicaiza, A.J., España, S., Pastor, O., 2021. Empirical validation of a quality framework for evaluating modelling languages in MDE environments. Softw. Qual. J. 29, 275–307. http://dx.doi.org/10.1007/s11219-021-09554-1.

Gorschek, T., Tempero, E., Angelis, L., 2010. A large-scale empirical study of practitioners' use of object-oriented concepts. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE). ACM, pp. 115–124. http://dx.doi.org/10.1145/1806799.1806820.

Gorschek, T., Tempero, E., Angelis, L., 2014. On the use of software design models in software development practice: An empirical investigation. J. Syst. Softw. (JSS) 95, 176–193. http://dx.doi.org/10.1016/j.jss.2014.03.082.

Hebig, R., Bendraou, R., 2014. On the need to study the impact of model driven engineering on software processes. In: Proceedings of the 2014 International Conference on Software and System Process (ICSSP). ACM, pp. 164–168. http://dx.doi.org/10.1145/2600821.2600846.

Hennink, M., Kaiser, B.N., 2022. Sample sizes for saturation in qualitative research: A systematic review of empirical tests. Soc. Sci. Med. 292, 114523. http://dx.doi.org/10.1016/j.socscimed.2021.114523.

Holtmann, J., Liebel, G., Stehöfer, J.-P., 2023. Supplementary material on "processes, methods, and tools in model-based engineering—A qualitative multiple-case study". http://dx.doi.org/10.5281/zenodo.8217258.

Hutchinson, J., Rouncefield, M., Whittle, J., 2011a. Model-driven engineering practices in industry. In: Proceedings of the 33rd International Conference on Software Engineering (ICSE). ACM, pp. 633–642. http://dx.doi.org/10.1145/1985793.1985882.

Hutchinson, J., Whittle, J., Rouncefield, M., 2014. Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure. Sci. Comput. Program. 89, 144–161. http://dx.doi.org/10.1016/j.scico.2013.03.017, Special issue on Success Stories in Model Driven Engineering.

Hutchinson, J., Whittle, J., Rouncefield, M., Kristoffersen, S., 2011b. Empirical assessment of MDE in industry. In: Proceedings of the 33rd International Conference on Software Engineering (ICSE). ACM, pp. 471–480. http://dx.doi.org/10.1145/1985793.1985858.

ISO, 2018. ISO 26262: Road vehicles—functional safety.

ISO, IEC, 2021. ISO/IEC TS 33061:2021: Information technology—process assessment—process assessment model for software life cycle processes.

ISO, IEC, 2022. ISO/IEC 27001:2022: Information technology—security techniques—information security management systems—requirements.

ISO, IEC, IEEE, 2017. ISO/IEC/IEEE 12207:2017: Systems and software engineering—software lifecycle processes. http://dx.doi.org/10.1109/IEEESTD.2017.8100771.

ISO, IEC, IEEE, 2023. ISO/IEC/IEEE 15288:2023: Systems and software engineering—system lifecycle processes. http://dx.doi.org/10.1109/IEEESTD.2023.10123367.

JetBrains s.r.o, 2023. JetBrains meta programming system (MPS). URL: https://www.jetbrains.com/mps/, accessed November 2023.

Kasauli, R., Knauss, E., Horkoff, J., Liebel, G., de Oliveira Neto, F.G., 2021. Requirements engineering challenges and practices in large-scale agile system development. J. Syst. Softw. (JSS) 172, 110851. http://dx.doi.org/10.1016/j.jss.2020.110851.

Kuhrmann, M., Tell, P., Hebig, R., Klünder, J., Münch, J., Linssen, O., Pfahl, D., Felderer, M., Prause, C.R., MacDonell, S.G., Nakatumba-Nabende, J., Raffo, D., Beecham, S., Tüzün, E., López, G., Paez, N., Fontdevila, D., Licorish, S.A., Küpper, S., Ruhe, G., Knauss, E., Özcan-Top, Ö., Clarke, P., McCaffery, F., Genero, M., Vizcaino, A., Piattini, M., Kalinowski, M., Conte, T., Prikladnicki, R., Krusche, S., Coşkunçay, A., Scott, E., Calefato, F., Pimonova, S., Pfeiffer, R.-H., Schultz, U.P., Heldal, R., Fazal-Baqaie, M., Anslow, C., Nayebi, M., Schneider, K., Sauer, S., Winkler, D., Biffl, S., Bastarrica, M.C., Richardson, I., 2022. What makes agile software development agile? IEEE Trans. Softw. Eng. 48 (9), 3523–3539. http://dx.doi.org/10.1109/TSE.2021.3099532.

Lee, I., Pappas, G.J., Cleaveland, R., Hatcliff, J., Krogh, B.H., Lee, P., Rubin, H., Sha, L., 2006. High-confidence medical device software and systems. Computer 39 (4), 33–38. http://dx.doi.org/10.1109/MC.2006.127.

Liebel, G., Knauss, E., 2023. Aspects of modelling requirements in very-large agile systems engineering. J. Syst. Softw. (JSS) 199, http://dx.doi.org/10.1016/j.jss.2023.111628.

Liebel, G., Marko, N., Tichy, M., Leitner, A., Hansson, J., 2018. Model-based engineering in the embedded systems domain: An industrial survey on the state-of-practice. Softw. Syst. Model. (SoSyM) 17, 91–113. http://dx.doi.org/10.1007/s10270-016-0523-3.

Liebel, G., Tichy, M., Knauss, E., 2019. Use, potential, and showstoppers of models in automotive requirements engineering. J. Softw. Syst. Model. (SoSyM) 18, 2587–2607. http://dx.doi.org/10.1007/s10270-018-0683-4.

Maurya, A., Kumar, D., 2020. Reliability of safety-critical systems: A state-of-the-art review. Qual. Reliab. Eng. Int. 36 (7), 2547–2568. http://dx.doi.org/10.1002/qre.2715.

Mayer, G., Spieckermann, S., 2010. Life-cycle of simulation models: requirements and case studies in the automotive industry. J. Simul. 4, 255–259. http://dx.doi.org/10.1057/jos.2010.10.

Meliá, S., Cachero, C., Hermida, J.M., Aparicio, E., 2016. Comparison of a textual versus a graphical notation for the maintainability of MDE domain models: an empirical pilot study. Softw. Qual. J. 24, 709–735. http://dx.doi.org/10.1007/s11219-015-9299-x.

MODELISAR Project Consortium, 2017. ITEA impact story—MODELISAR: An international standard for systems and embedded software design in vehicles. URL: https://itea4.org/project/impact-stream/modelisar-impact-story.html, accessed November 2023.

Mohagheghi, P., Dehlen, V., 2008. Where is the proof?—A review of experiences from applying MDE in industry. In: Proceedings of the European Conference on Model Driven Architecture–Foundations and Applications (ECMDA-FA). In: LNCS, vol. 5095, Springer, pp. 432–443. http://dx.doi.org/10.1007/978-3-540-69100-6_31.

Mohagheghi, P., Fernandez, M.A., Martell, J.A., Fritzsche, M., Gilani, W., 2009. MDE adoption in industry: Challenges and success criteria. In: Models in Software Engineering: Reports and Revised Selected Papers of the Workshops and Symposia At MODELS 2008. In: LNCS, vol. 5421, Springer, pp. 54–59. http://dx.doi.org/10.1007/978-3-642-01648-6_6.

Mohagheghi, P., Gilani, W., Stefanescu, A., Fernandez, M.A., Nordmoen, B., Fritzsche, M., 2013. Where does model-driven engineering help? Experiences from three industrial cases. J. Softw. Syst. Model. (SoSyM) 12, 619–639. http://dx.doi.org/10.1007/s10270-011-0219-7.

OASIS Open, 2023. Open services for lifecycle collaboration (OSLC). URL: https://open-services.net, accessed November 2023.

Object Management Group, 2023a. Requirements interchange format (ReqIF). URL: https://www.omg.org/spec/ReqIF, accessed November 2023.

Object Management Group, 2023b. Systems modeling language (SysML). URL: https://www.omg.org/spec/SysML, accessed November 2023.

Object Management Group, 2023c. Unified modeling language (UML). URL: https://www.omg.org/spec/UML, accessed November 2023.

Pagel, M., Brörkens, M., 2006. Definition and generation of data exchange formats in AUTOSAR. In: Proceedings of the 2nd European Conference on Model Driven Architecture – Foundations and Applications (ECMDA-FA). In: LNCS, vol. 4066, Springer, pp. 52–65. http://dx.doi.org/10.1007/11787044_5.

RealtimeBoard, I., 2023. Miro virtual whiteboard. URL: https://www.miro.com/, accessed November 2023.

Reggio, G., Leotta, M., Ricca, F., 2014. Who knows/uses what of the UML: A personal opinion survey. In: Proceedings of the 17th International Conference on Model-Driven Engineering Languages and Systems (MODELS). In: LNCS, vol. 8767, Springer, pp. 149–165. http://dx.doi.org/10.1007/978-3-319-11653-2.

Reggio, G., Leotta, M., Ricca, F., Clerissi, D., 2013. What are the used UML diagrams? A preliminary survey. In: Proceedings of the 3rd International Workshop on Experiences and Empirical Studies in Software Modeling (EESSMod). pp. 3–12.

Runeson, P., Höst, M., 2008. Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. 14 (2), 131–164. http://dx.doi.org/10.1007/s10664-008-9102-8.

Runeson, P., Höst, M., Austen, R., Regnell, B., 2012. Case Study Research in Software Engineering—Guidelines and Examples, first ed. Wiley.

Saldaña, J., 2015. The Coding Manual for Qualitative Researchers. SAGE Publications.

Schmelter, D., Steghöfer, J.-P., Albers, K., Ekman, M., Tessmer, J., Weber, R., 2023. Trustful model-based information exchange in collaborative engineering. In: Proceedings of the 30th European Conference on Systems, Software and Services Process Improvement (EuroSPI). In: Communications in Computer and Information Science (CCIS), vol. 1890, Springer, http://dx.doi.org/10.1007/978-3-031-42307-9_12.

Stachowiak, H., 1973. Allgemeine Modelltheorie. Springer.

Star, S.L., Griesemer, J.R., 1989. Institutional ecology, 'translations' and boundary objects: Amateurs and professionals in berkeley's museum of vertebrate zoology, 1907-39. Soc. Stud. Sci. 19 (3), 387–420. http://dx.doi.org/10.1177/030631289019003001.

Steghöfer, J.-P., Koopmann, B., Becker, J.S., Törnlund, M., Ibrahim, Y., Mohamad, M., 2021. Design decisions in the construction of traceability information models for safe automotive systems. In: Proceedings of the 29th IEEE International Requirements Engineering Conference (RE). IEEE, pp. 185–196. http://dx.doi.org/10.1109/RE51729.2021.00024.

The MathWorks, Inc, 2023. Matrix laboratory (MATLAB®). URL: https://mathworks.com/products/matlab.html, accessed November 2023.

Tomassetti, F., Torchiano, M., Tiso, A., Ricca, F., Reggio, G., 2012. Maturity of software modelling and model driven engineering: A survey in the Italian industry. In: Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering (EASE). IET, pp. 91–100. http://dx.doi.org/10.1049/ic.2012.0012.

Torchiano, M., Tomassetti, F., Ricca, F., Tiso, A., Reggio, G., 2011. Preliminary findings from a survey on the MD* state of the practice. In: Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement (ESEM). IEEE, pp. 372–375. http://dx.doi.org/10.1109/ESEM.2011.51.

Torchiano, M., Tomassetti, F., Ricca, F., Tiso, A., Reggio, G., 2013. Relevance, benefits, and problems of software modelling and model driven techniques—A survey in the Italian industry. J. Syst. Softw. (JSS) 86 (8), 2110–2126. http://dx.doi.org/10.1016/j.jss.2013.03.084.

Tsang, E.W., 2014. Generalizing from research findings: The merits of case studies. Int. J. Manage. Rev. 16 (4), 369–383. http://dx.doi.org/10.1111/ijmr.12024.

Đurković, J., Vuković, V., Raković, L., 2008. Open source approach in software development—Advantages and disadvantages. Manage. Inf. Syst. 3, 29–33.

VanDerHorn, E., Mahadevan, S., 2021. Digital twin: Generalization, characterization and implementation. Decis. Support Syst. 145, 113524. http://dx.doi.org/10.1016/j.dss.2021.113524.

VDA, 2014. Harness description list (KBL)—recommendation, 2nd edition (KBL 2.4). URL: https://www.vda.de/dam/jcr:73be8037-abbf-400d-a91c-636b8ff1e1ff/harness-description-list-kbl-4964.pdf, accessed November 2023.

Weissnegger, R., Schuss, M., Kreiner, C., Pistauer, M., Römer, K., Steger, C., 2016. Simulation-based verification of automotive safety-critical systems based on EAST-ADL. Procedia Comput. Sci. 83, 245–252. http://dx.doi.org/10.1016/j.procs.2016.04.122.

Whittle, J., Hutchinson, J., Rouncefield, M., 2014. The state of practice in model-driven engineering. IEEE Softw. 31 (3), 79–85. http://dx.doi.org/10.1109/MS.2013.65.

Whittle, J., Hutchinson, J., Rouncefield, M., Burden, H., Heldal, R., 2013. Industrial adoption of model-driven engineering: Are the tools really the problem? In: Proceedings of the 16th International Conference on Model-Driven Engineering Languages and Systems (MODELS). In: LNCS, vol. 8107, Springer, pp. 1–17. http://dx.doi.org/10.1007/978-3-642-41533-3_1.

Whittle, J., Hutchinson, J., Rouncefield, M., Burden, H., Heldal, R., 2017. A taxonomy of tool-related issues affecting the adoption of model-driven engineering. Softw. Syst. Model. (SoSyM) 16, 313–331. http://dx.doi.org/10.1007/s10270-015-0487-8.

Wohlrab, R., Horkoff, J., Kasauli, R., Maro, S., Steghöfer, J.-P., Knauss, E., 2020. Modeling and analysis of boundary objects and methodological islands in large-scale systems development. In: Proceedings of the 39th International Conference on Conceptual Modeling (ER). In: LNCS, vol. 12400, Springer, pp. 575–589. http://dx.doi.org/10.1007/978-3-030-62522-1_42.

Wohlrab, R., Pelliccione, P., Knauss, E., Larsson, M., 2019. Boundary objects and their use in agile systems engineering. J. Softw.: Evol. Process (JSEP) 31 (5), e2166. http://dx.doi.org/10.1002/smr.2166.

Ye, X., Yu, M., Song, W.S., Hong, S.H., 2021. An asset administration shell method for data exchange between manufacturing software applications. IEEE Access 9, 144171–144178. http://dx.doi.org/10.1109/ACCESS.2021.3122175.

Zaitsev, A., Tan, B., Gal, U., 2016. Collaboration amidst volatility: The evolving nature of boundary objects in agile software development. In: Proceedings of the European Conference on Information Systems (ECIS). https://aisel.aisnet.org/ecis2016_rp/172.

**Jörg Holtmann** independently conducts research while working in the German railway industry. Formerly, he was a PostDoc at Chalmers University of Technology | University of Gothenburg in Sweden, after being a senior expert at the Fraunhofer IEM in Germany. His research interests center around model-based requirements/systems/software engineering for software-intensive systems.

**Grischa Liebel** is an Assistant Professor in Software Engineering and the director of the CRESS research centre at Reykjavik University, Iceland. He holds a Ph.D. degree from Chalmers University of Technology, Sweden. His research focuses on human factors in software engineering, typically in areas such modeling and model-based engineering, requirements engineering, processes, and education. Much of Grischa's research is done in collaboration with industry (and with humans).

**Jan-Philipp Steghöfer** is a senior researcher at XITASO and formerly an associate professor at the Interaction Design & Software Engineering Division of Chalmers University of Technology and the University of Gothenburg. His main research interests are software traceability, model-driven engineering, security assurance, and agile requirements engineering.