How are Conceptual Models used in Industrial Software Development?: A Descriptive Survey

Conference Paper · June 2017

DOI: 10.1145/3084226.3084256

1 author:

Harald Störrle
QAware

How are Conceptual Models used in Industrial Software Development? A Descriptive Survey

Harald Störrle
QAware GmbH, Aschauer Str. 32, 81549 München, Germany
HStorrle@acm.org

ABSTRACT

Background: There is a controversy about the relevance, role, and utility of models, modeling, and modeling languages in industry. For instance, while some consider UML as the *"lingua franca of software engineering"*, others claim that "*the majority [of industry practitioners] simply do not use UML.*"

Objective: We aspire to evolve this debate to differentiate the circumstances of modeling, and the degrees of formality of models.

Method: We have conducted an online survey among industry practitioners and asked them how and for what purposes they use models. The raw (anonymized) survey data is published online.

Results: We find that models are widely used in industry, and UML is indeed the leading language. Three distinct usage modes of models are reported, the most frequent of which is informal usage for communication and cognition. MDE-style usage is rare, but does occur. Software architects are believed to benefit most from modeling.

Conclusions: Our study contrasts and complements existing studies, and offers explanations for some of the seeming contradictions of previous results. There might be cultural differences in modeling usage that are worth exploring in the future.

ACM Reference format:
Harald Störrle. 2017. How are Conceptual Models used in Industrial Software Development? A Descriptive Survey. In *Proceedings of EASE'17, Karlskrona, Sweden, June 15-16, 2017,* 10 pages.
https://doi.org/http://dx.doi.org/10.1145/3084226.3084256

1 INTRODUCTION

There has been considerable controversy regarding the extent to which industrial software engineering benefits from conceptual modeling. Those in favor maintain that *"Model-based approaches [...] hold out the promise of significantly improving the productivity of software developers and the quality of the products they generate"* [38, p. 525]. They particularly promote using the Unified Modeling Language (UML) as the *"lingua franca of software engineering"* [18, p. v] in the context of the Model Driven Engineering (MDE)[1] paradigm which, they believe, "*has the potential to greatly reduce development time*", and *"will help us develop better software systems faster."* [6, p. 8]. Furthermore, they claim that "*MDA is not a vision of some future [...] it has already proven itself many times over in diverse application domains*" (ibid). Note that this quote is from 2004.

However, Mohagheghi and Dehlen famously asked "*Where is the proof?*" in their 2008 landmark paper [29]. Even MDA-supporters had to admit that *"adoption of this approach has been surprisingly slow"* [38, p. 513]. Various contributions have since studied success cases and failure cases of MDA adoption in an attempt to explain exactly when and why MDA projects fail or succeed [20–22; 41; 42]). Others have outright denied that UML is used to any degree in industry [31; 32]. Where UML is used, Petre claims, it is used only informally, and *"if models end up merely as documentation, they are of limited value [...]. Consequently, a key premise behind MDD is that programs are automatically generated from their corresponding models"* [37, p. 20]. So, in order to assess the status and claim of MDD, it is quite relevant to understand whether conceptual modeling is used in industry, and what languages are actually in use. Thus, our first research question is:

RQ 1: Are conceptual modeling languages like UML or BPMN (widely) used in the (software) industry at all?

Biased by personal experience, the author believes this is true. If indeed it is, we ask our second research question:

RQ 2: When, for what purposes, and by whom are models used in industrial software development?

This question may yield a simple enumeration of distinct purposes, or we may discover a structure in this set. For instance, Fowler [17] postulates the three model-roles "sketch", "blueprint", and "program" with increasing degrees of formality: *fully formal* models that *are* system representations, i. e., the kind of models MDD purposes (see quote above), *semi-formal* models serve to plan or document a system, and *informal* models are ephemeral sketches on whiteboards or napkins serve to support *conception of and conversation about* systems. While this distinction has a strong intuitive appeal, there is no direct evidence for it. After all, there could be other usage modes, too. Thus our third question is:

RQ 3: Are there distinct usage modes for models, and if so, how many can be distinguished?

The first three questions may appear trivial, yet there is disagreement about them. Answering the first three questions establishes a base line to ask the fourth, decisive question:

RQ 4: What is the relative frequency of the usage modes of models?

The answer to this question will be indicative of the state of MDE, and can inform tool builders as well as researchers.

[1] There are several formulations of essentially the same paradigm, e. g., Model Driven Architecture (MDA), and Model Driven Development (MDD). We use these terms interchangeably.

EASE'17, June 15-16, 2017, Karlskrona, Sweden

ACM ISBN 978-1-4503-4804-1/17/06…$15.00
https://doi.org/http://dx.doi.org/10.1145/3084226.3084256

2 RESEARCH METHODOLOGY

In this section we present the study design, describe the study execution and data analysis, and discuss threats to validity. An overview of the setup and major parameters of this survey is presented in Fig. 1. The live survey is available at http://tinyurl.com/MU-survey-2014, a dump of the full questionnaire as well as the (anonymized) results are available as [40].

2.1 Study design

Guidelines to create surveys are presented in [23] and [39], which our study conforms to. As our research goal is of primarily descriptive nature, a cross-sectional design is most appropriate. Given that Software Engineering is a discipline practiced all over the world, and informed by our hypothesis $\mathbf{H}_1$ that there are cultural and regional differences, we concluded that an online survey is the only viable implementation of our survey. Of course, it is also a very cost-effective way of conducting a survey.

2.2 Questionnaire construction

Due to space restrictions, we cannot include the complete questionnaire in this paper. They can be found in an archival link, however [40]. We designed the questionnaire based on our research questions and evaluated it in two rounds. The first round served to identify and resolve issues in the question wording. To this end, we presented the survey questions to three graduate students and asked them to paraphrase the questions. Whenever discrepancies between interpretations arose, we rephrased or clarified the questions. We also added explanatory texts to all questions as a result.

In the second round, we aimed at ensuring the relevance and completeness of the questionnaire. To this end, we presented the questionnaire to two acquainted industry professionals and asked for feedback that was duly processed. As an ongoing control for questionnaire validity, we encouraged participants to also provide meta comments about the questionnaire as such, that is, whether they considered it completeness and meaningful. Participants did actually use these answer fields, but with one exception, no problematic issues were discovered.

The questionnaire consists of an instruction and five main parts. In the introduction we explained our research objective, provided instructions, offered to leave a feedback address, and requested explicit consent to participating. In the first section we asked cultural background and participants demographic details such as their education, occupation, and current affiliation. In the second part, participants were asked about their professional experience. Based on the self-reported kind of affiliation, participants from academia and industry were asked slightly different questions.

The third and fourth parts of the questionnaire constitutes the main part of the survey and investigates experiences with modeling and modeling languages in general (ppart 3) and in an industrial context (part 4, again only presented to people from industry). The fifth and last part of the questionnaire asked the degree of agreement or disagreement to a set of opinions, and offered participants with free text fields for adding any other comments or opinions. Participants were encouraged to also reflect on the questionnaire itself, e.g., we asked "*Are there any questions that we haven't asked but that you expected?*".

2.3 Study execution

This study is conceived as a probe into the practice of modeling, and how opinions and experiences differ depending on professional maturity and type of affiliation, in particular industry and academia. Attracting academics and junior practitioners is fairly simple. However, attracting a sizable number of senior and very senior practitioners all over the world is quite a challenge: this population is in high demand and very likely does not respond to cold calling or anonymous campaigns on social media. Our recruiting strategy, thus, consisted of three elements.

- First, we used all the conventional advertisement channels like the social media (particularly, Twitter and LinkedIn), included the link to the survey in our email signature, and advertised it on our web page. We also distributed advertisement material at over 15 scientific workshops and conferences like MODELS.
- Second, we contacted our personal network in industry with the request to pass on word about the survey. We also asked our students to spread the word at their jobs, and distributed fliers at trade fairs like Cebit and industry-oriented conferences like OOP and EclipseWorld.
- Third, we advertised the survey in person when giving more than ten talks at evening functions arranged by industrial special interest groups, regional chapters of the ACM, professional bodies, and similar organizations. This strategy proved to be particularly effective in attracting contributions: spikes in the participant numbers inevitably followed after a presentation.

All of these activities were conducted in a 18 month period, from September 2014 to April 2016. We chose Google Forms as a platform to implement our questionnaire. We attracted a total of 96 completed answers (Google Forms does not store incomplete answers). Many of the arguments against online questionnaires do not apply to the audience addressed in this survey (e.g., limited computer literacy, restricted internet access), and we agree that *"the advantages of Internet surveys outweigh many of the disadvantages."* [39, p. 153].

We then copied the data over to a local spreadsheet, and manually separated and regrouped the data columns. The data was then encoded, where necessary (e. g., free-form questions, "other" choices of closed questions), and exported to R [33]. We used the `Likert` package to analyze the data [10]. No coding occurred for the prose answers.

2.4 Threats to validity

Since our data analysis does neither aspire to uncover causal relationships nor use inferential statistics, internal validity is of no concern to us [36]. For a survey like this, presenting primarily qualitative results, the primary concerns are thus external and construct validity.

2.4.1 External validity. Obviously, there are significant threats to external validity, that is, whether the results obtained in this survey allow generalization and replication. First, the population studied in this survey is not necessarily representative for the global community of practitioners in the software industry. First, we have

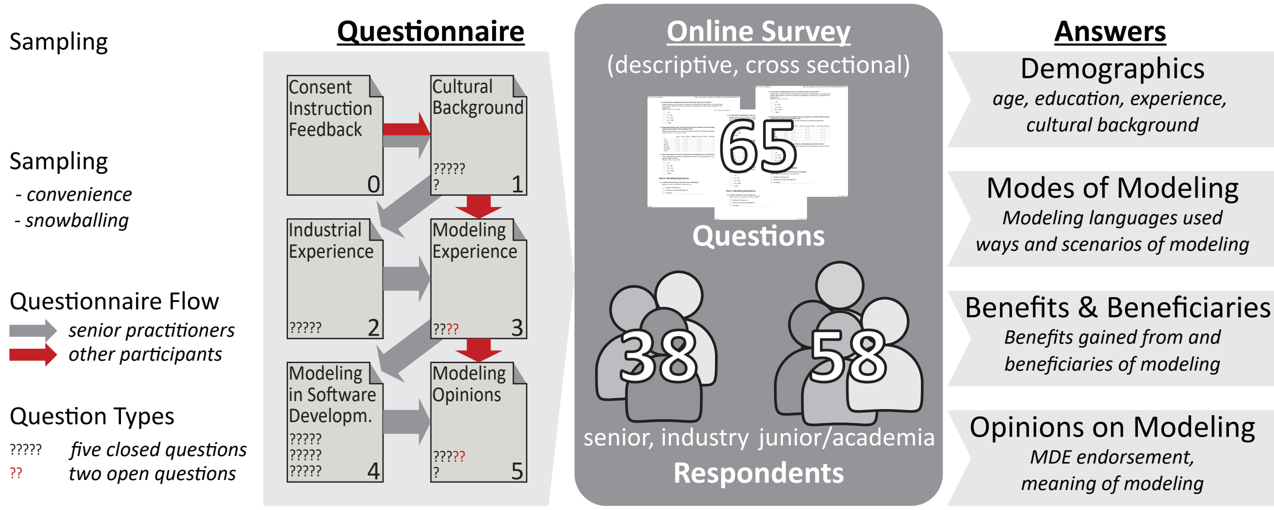

Figure 1: The setup and major parameters of the survey reported here: the different types of arrow on the left indicate different sets of questions administered to practitioners (solid blue arrows) and academics (dashed black arrows). The shapes on the right indicate the kinds of answers discussed in Section Section 3.

used *convenience sampling* rather than probability sampling resulting in uneven representations of, in particular, Germany and the US. So while it perfectly possible that our findings are an effect of cultural differences (as some people in the community will maintain), we have no proof. Second, with self-administered online questionnaires such as Google Forms, we have no control over the completion rate and so this survey may suffer from response rate bias – only completed forms are saved. Since recruitment depended to a notable degree of the author's personal appearances, there is also a risk of experimenter bias. In defense of this research, we would like to highlight that this is a general open methodological problem [24, p. 105] and no comparable survey in the area has used probability sampling.

Clearly, our study suffers from selection bias in the sense that only those people are likely to answer our survey about modeling that are interested in modeling. Therefore, we can not support a conclusions about IT professionals that do not model. We can, however, claim that there is a significant number of people that do model, and those that do exhibit certain usage patterns. Thus, our conclusions are not compromised by the validity threats explained here.

2.4.2 Construct validity. While this questionnaire was constructed with great care and was piloted repeatedly, there is a risk of biasing the participants by the questions and the answer options. For instance, in the list of usage profiles of models, we might have omitted some, or misrepresented others. To mitigate this problem, we asked participants to describe further scenarios in prose. Many participants took advantage of such text fields, and we could indeed identify one usage profile we had omitted.

Also, we specifically asked whether participants felt that any important questions were missing. While the respondents consistently offered opinions in other text fields, there was little feedback in this category. The questions that were suggested pointed to aspects beyond the scope of this study, such as tooling, particular modeling languages, and visual vs. textual notations for modeling. Some participants also suggested to drill deeper into application conditions (system size, availability of formal training), and causes of modeling failures. There was no comment suggesting that the questions were wrong, irrelevant, or biased, though there were two participants that understood "model" in the sense of mathematical model rather than conceptual model in software engineering. We take this as a confirmation that the questions were fair and balanced.

3 OBSERVATIONS

This whole study hinges on attracting the "right" participants: we need to survey a broad spectrum of suitably qualified and experienced respondents in order to achieve ecological validity. Thus, we first analyze the demographic structure of the participants in greater than usual depth. As we go along we shall point out potential biases that will have to be considered when deriving conclusions from our observations.

3.1 Participant demographics

For the given purpose of this survey, we believe the answers of experienced professionals to be more valuable than those of students, academics, or junior developers, say. Thus, our first goal in recruiting was to attract senior people with more overview and insight. Fig. 2 shows the demographic structure of the participants of our survey. Obviously, the vast majority has an education in Computer Science, a significant portion up to the level of PhD. Similarly, a majority of the participants come from industry (49 of 96), and here, most are senior people (38 of 49). The survey also attracted a number of academics, most of them junior (23 of 43). While this is not our targeted audience, it allows us to contrast the opinions held in academia and industry as well as differences between junior and senior people in the area. The gender distribution is largely representative of the field. Most participants are in their 30s, with a self-assessed median modeling proficiency 7 out of 10.

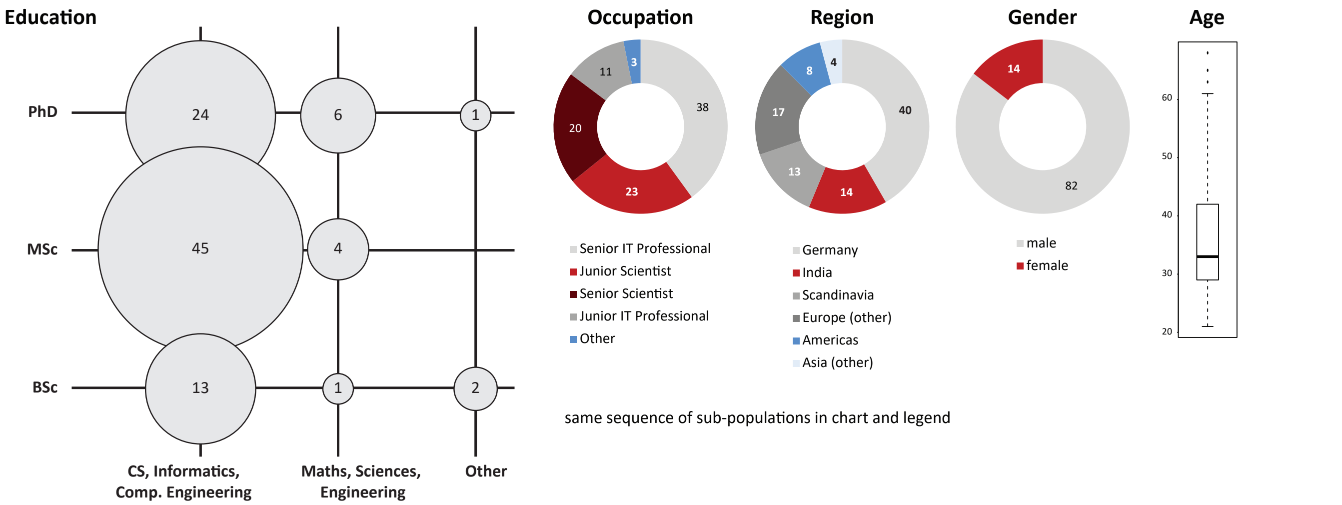

Figure 2: Demographic structure of the participant population: level and area of education, occupation, region, gender, age, and years of professional experience (left to right). Gender is reported for completeness only.

We also tried to include practitioners from all over the world, assuming that regional differences reflect cultural differences, which we suspect might have an impact too—this sentiment has been repeatedly expressed by renowned figures in the MDE community (e. g., "*people in the US just do it in a less structured way*", Bran Selic, personal communication). Clearly, the regional distribution of the study participants strongly favors Europe, in particular Germany and, to a lesser degree, Scandinavia. This is clearly a consequence of the recruitment process, and the author's personal network and affiliation. Unlike previous surveys, however, there is also a sizable number of participants from Asia, mostly from India. About half way through the survey we observed the absence of participants from the UK and North America. We reached out to contacts in those regions, and specifically tried to attract respondents from these regions, but without much success. It is not clear whether this is a weakness of this study or a finding, indicating less model usage in those regions.

From the 38 senior respondents, we asked additional details about the kind of projects on which they have worked, see Fig. 3. Given the conventional career path in industry, we expected a clear dominance of managers, consultants, and software architects, but among the respondents of our survey, there is also a large number of developers and testers. As expected, the vast majority of senior professionals work in fairly large commercial corporations, and for even larger client organizations.

We have also asked the senior respondents to indicate the size of projects they have witnessed, and the absolute frequency in which these occur. Unsurprisingly, very few participants report having been part of very large projects. Interestingly, though, a fair number of the participants report that they have never or rarely been part of small (less than 10 people) and very small projects (less than 5 people). The results of this survey are thus biased towards large projects.

We conclude that overall, the participant population matches our expectations, and allows us to draw conclusions about the opinions held by both junior and senior IT professionals in the software development industry. Note that our analysis exhibits a likely source of bias: large organizations and large projects are known to tend towards more formalized development processes, which may favor the use of modeling over more ad-hoc, non-modeling software development methods.

3.2 Usages modes of modeling

We asked participants to indicate in which contexts they use modeling, what modeling languages they use, and how they assess their level of modeling expertise (see Fig. 5). Clearly, using models in the context of software development is the dominating usage context (76%)(Question C1), in line with the intended audience of this survey. To a lesser degree (18%), participants stated more domain-oriented contexts, such as business process or enterprise architecture modeling. Clearly, the UML is the most used modeling language among the respondents (83 out of 96), with Domain Specific Languages (DSLs) coming in second (31 of 96).

Then, we asked for specific usages of models and offered 16 predefined items with a five-point Likert scale each to indicate the frequency of these usages. Despite our best efforts in constructing this questionnaire, there is a risk of biasing the participants by these 16 items. To mitigate this problem, we asked participants to describe further scenarios in prose, and 16 (17%) of them used this option. Seven of these alternate scenarios were minor variations of the categories initially offered, six referred to testing and verification, and three showed a different understanding of "model" altogether. So, we conclude that apart from the usage category "Testing and Verification" the suggested set of model usages is adequate. The results are presented in Fig. 6.

The most popular usage scenarios by far (scenarios 1-3) center around communicative and cognitive processes: between 70 and 79% of all participants use models for such activities "often" or "always", while only between 4 and 8% do this "rarely" or "never". In contrast, about half the population in our survey used models "rarely" or "never" for generating code or a DSL, while only a quarter to a third of the population did this "often" or "always". Also,

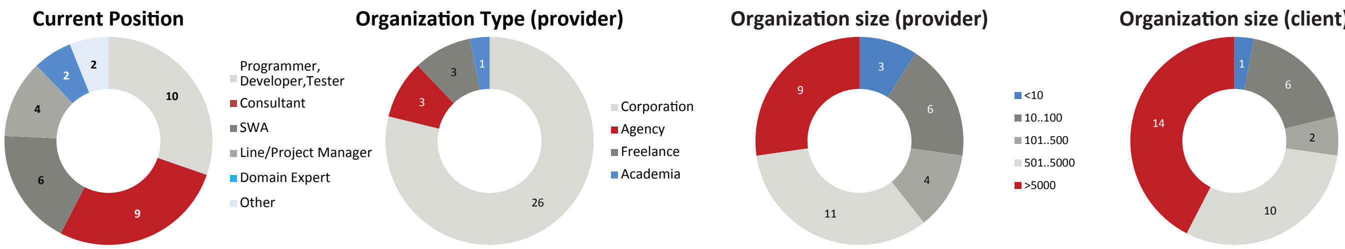

Figure 3: Details about the professional experience of study participants: Current position, type and size of the organization they are affiliated with, and years of professional experience (left to right). Only the results from senior professionals are reported here (n=38), with some participants skipping these questions.

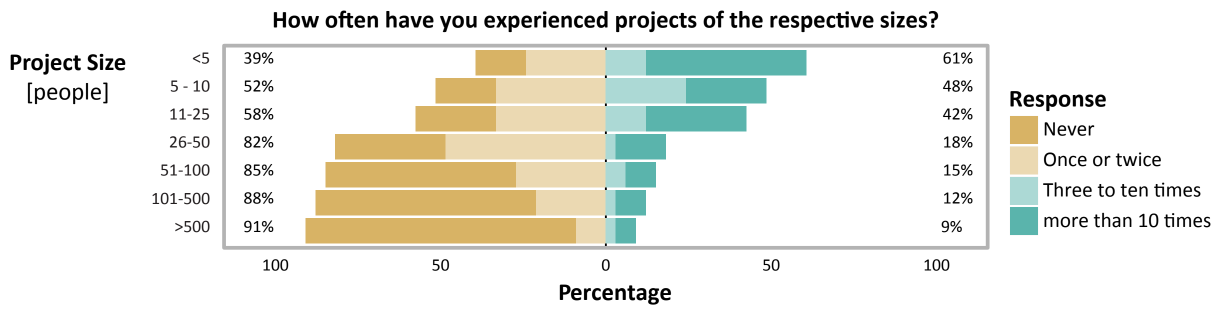

Figure 4: Few participants have witnessed ultra large projects, but many have also only been part in mid-sized or large projects. Only the results from senior professionals are reported here (n=38).

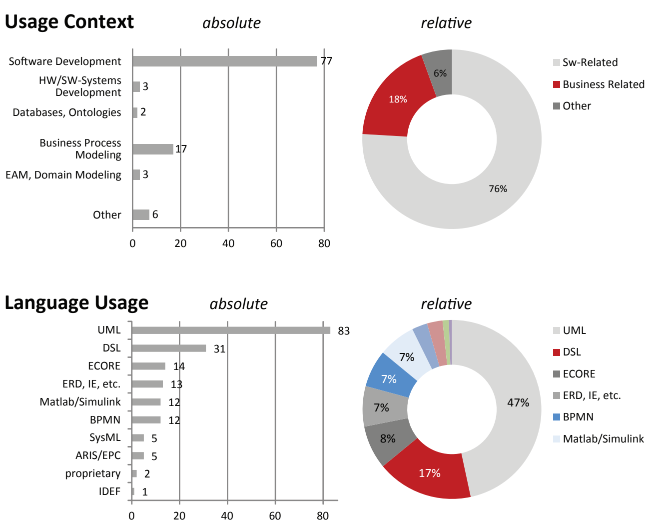

Figure 5: More details about the modeling experience of study participants: Usage context of models in absolute and relative numbers (top), modeling languages used in absolute and relative numbers (bottom).

using models for domain- and requirements-oriented tasks (scenarios 4, 6, 9) is apparently more common than using models for technology-specific tasks (scenarios 5, 7, 10, 11). This is in line with the findings of [31] who concludes that UML is mostly used as a "thought tool" and to facilitate communication with stakeholders. Usage scenario 2 ("Visualize an idea or concept") is the one that is used at least "sometimes" by most participants (96%). This supports the commonly found sentiment that the visual nature of modeling languages like UML or BPMN is of prime importance.

We find it interesting that while 45% answered they use models to document software "often" or "always", only 20% answered they use models to look up information "often" or "always". A plausible explanation could be that models are in principle useful to find information about a given system, but may often be out of date or contain not the right level of detail to be of use. It could also mean that it is difficult to find the appropriate model, or relevant information *inside* a model. All of these phenomena have previously been described by [25].

As we have seen, models are widely used in eliciting knowledge in early life-cycle phases. Models are also mentioned as a versatile tool in "Software Archeology" [19], that is, (re)discovering of knowledge in the (late) maintenance stage. Judging by our results, though, this is not a common practice as most participants reported doing this never or rarely (55%), and only a fifth did it often or always. This is consistent with the finding that using models to look up information about a product is a relatively uncommon practice, as mentioned above. The only model-related less common than these two is using models as (part of) contractual documents: only 16%

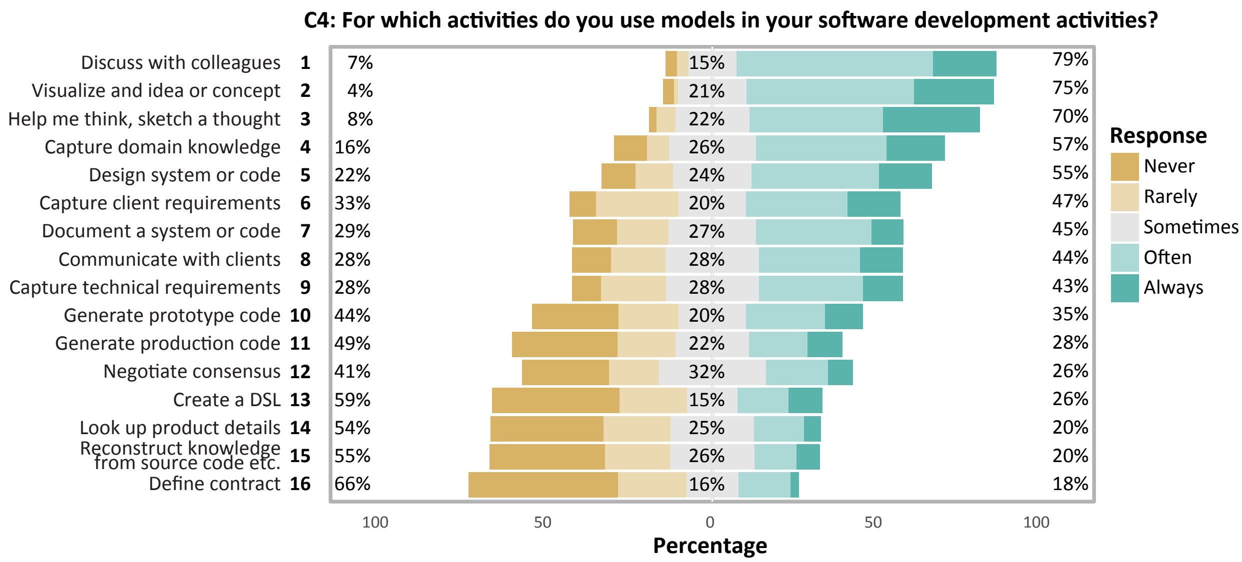

Figure 6: Frequency of model usage scenarios sorted by decreasing frequency. The numbers behind the model usage scenario names are used as shorthands to identify them. Answers from all respondents are reported here.

use models this way often or always, while two thirds never or rarely do this.

Based on our own experience, we distinguish the following three kinds of models with specific usage scenarios.

- **Informal models** support communication and cognition, utilizing rich information implicit in the situational context.
- **Semi-formal models** support design and documentation activities.
- **Fully formal models** are to be taken literal and binding, so as to allow the analysis of system properties, simulation, and generation of code and test cases. Fully formal models can also be used like legal documents such as contracts, or other formalized agreements.

Observe that this grouping aligns with Fowler's distinction of models as sketches, blueprints, and programs [17]. It turns out in terms of these categories, the sub-populations are in agreement. Furthermore, there is a clear priority among the three usage scenario types: the more formality is required, the less frequently the usage occurs. This is in line with the results reported previously by Baltes and Diehl [3].

Interestingly, when splitting the answers by sub-populations, we see differences between the usage profiles of senior professionals and other participants: observe how the relative ranks of the practices shuffle for senior practitioners (see Fig. 7, left) while the overall shape of the graphs stays more or less the same. When looking at the results from all other survey participants sorted by the same activities, differences become visible (see Fig. 7, right). However, these occur almost exclusively *within* the three major usage modes. That is to say, while the frequency of individual activities varies a bit by population, the overall usage modes do not: it seems, that there is surprisingly little difference between industrial and academic usages of models.

3.3 Benefits and beneficiaries of modeling

Next we asked who benefits from modeling (see Fig. 8). We presented seven kinds of stakeholders and roles that are involved in the process of software development and asked how much they benefit from modeling (Q 25). Unsurprisingly, participants believed that Software Architects have the greatest benefit: 91% of the respondents believed that Software Architects benefit of from modeling "a lot" or "crucially", and only one believed that their benefit is "little".

All other roles are ascribed less benefit from modeling, in particular, the percentage asserting "crucial" benefit more than halves from Software Architects to the next category Developer/Tester (18 down from 40). The following categories of Domain Expert and Requirements Analyst get very similar assessments with 17/24 and 16/26 respondents asserting "a lot of" and "crucial" benefit, respectively. The next two categories of Project Managers and Clients are again rated very similar. Interestingly, even the last category of End Users ere said to gain at least some benefit, according to 39% of the respondents.

In order to correct for question bias, we also asked respondents to add other stakeholder that we may not have asked for (Q26), and eight respondents mentioned substantially different roles. Partly, these were technical roles such as UX/UI designers, product managers, and technical writers. One respondent said that technical risk analysts have crucial benefit from modeling. The other answers referred to other, less obvious stakeholders, such as the media and regulatory bodies, or roles related to administrative roles bordering on software development such as enterprise modelers, database administrators, and contract managers. All of these roles are not directly inside the core of software development, highlighting the communicative function of models. Or, as one respondent put it "*I think modeling is beneficial in terms of knowledge transfer, and since software development is more and more a social spectacle, it's borderline crucial.*"

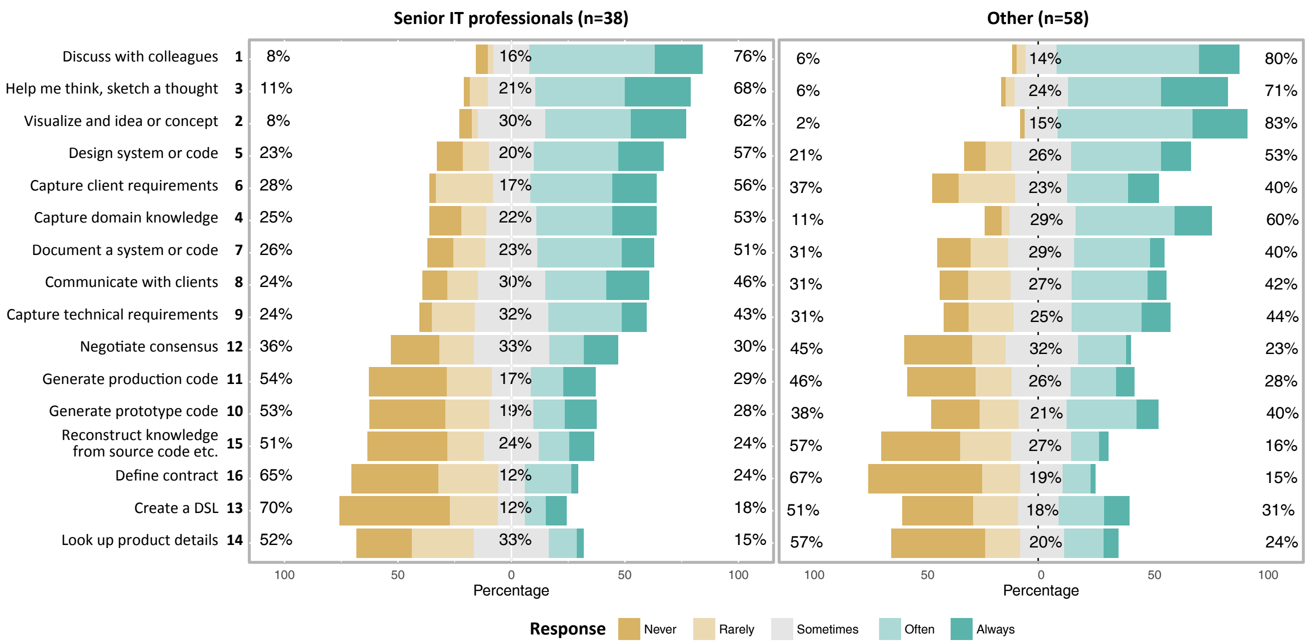

Figure 7: Frequency of model usage scenarios split by senior professionals (left) and all other participants (right).

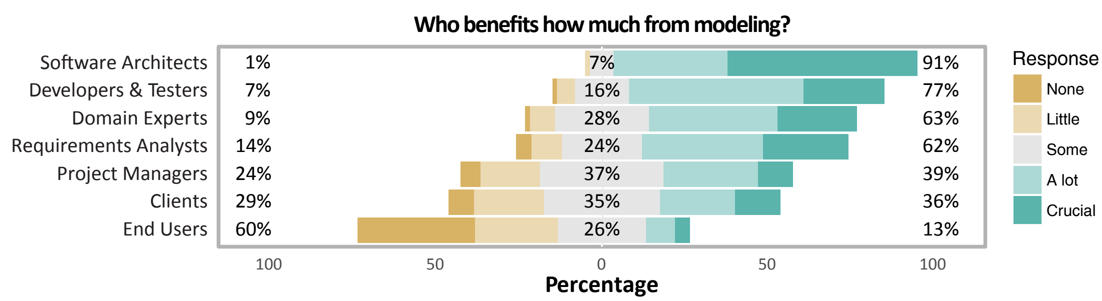

Figure 8: Beneficiaries of modeling according to respondents

3.4 Opinions about modeling

As the last part of our questionnaire, we asked respondents for their opinions on modeling in general. We first checked the degree of endorsement of seven statements, and asked for additional free-text comments. Regarding the predefined items (see Fig. 9), 97% of the respondents agree that modeling helps them deliver software with higher quality. Much lower, though still large numbers agree that modeling also helps with reducing effort, speeding up delivery, and increased market agility (72%, 66%, 61% agreement, respectively).

The opinion item with the most divided response was about the MDE vision of generating complete applications from models. Only half the respondents shared that vision, while 37% disagreed with it. This is a clear indication that modeling as such is doubtlessly useful, but MDE-style code generation is not, at least not today, or not to the same degree as modeling. This resonates with the usage modes reported above where almost all respondents use informal modeling frequently, but formal modeling seems to be used a lot less often. One respondent wrote that "*Modeling for code generation can (only) be useful in certain domains, e. g., smaller systems with safety critical applications*". Another respondent opined that "*Modeling for communication between humans and modeling for generating code can not be done using the same models [when considering] large, real world applications. Models for code generation are [...] too detailed, too technical for easy understanding. On the other hand, models for communication require "strong" abstractions which make them inappropriate for code generation.*"

The existence of different modes of modeling was highlighted independently by several other respondents with remarks like "*'modeling' [...] can range from information sharing at a whiteboard to construction of entire applications using modeling tools and code generation*" and "*there are no projects without modeling, but there are different kinds of modeling*". Another respondent wrote "*Not in every scenario, I will use a dedicated modeling tool to create something that is a model in the strict sense of the word. For instance, to communicate with my colleagues, I will often draw sketches that might incorporate some modeling notation.*"

Another ubiquitous aspect is that of tooling. It is worth stating here that the modeling tools used most commonly in industrial software development are not, as one might think, UML tools like

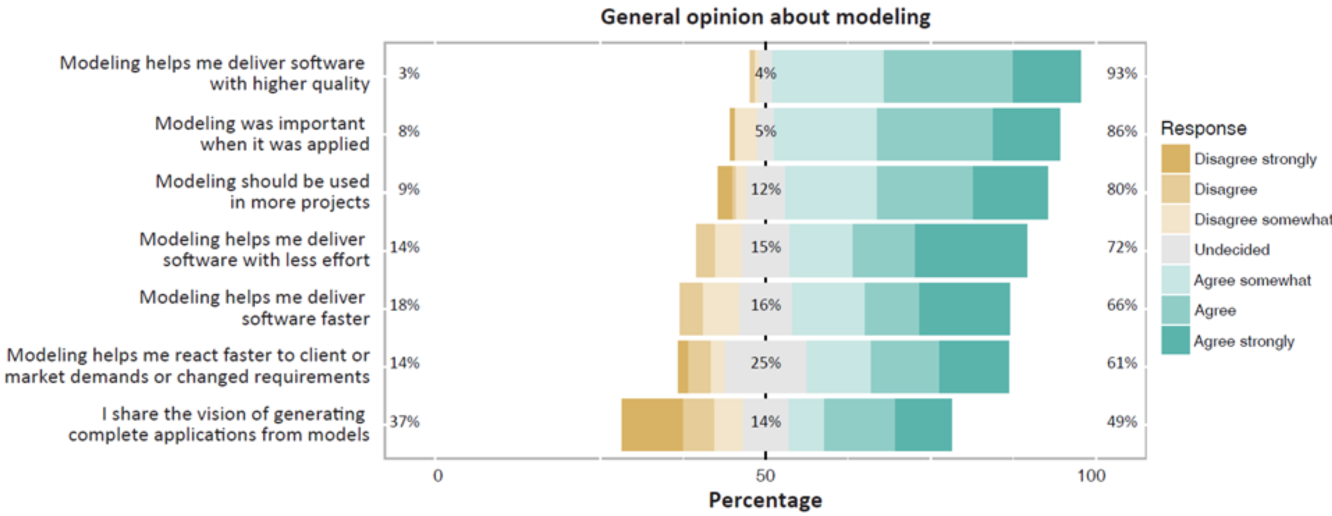

Figure 9: General opinions about modeling: degree of endorsement of a number of opinions

MagicDraw or EnterpriseArchitect, but generic drawing tools like PowerPoint and Visio. Respondents expressed dissatisfaction with existing tools, e. g. "*the available tools, such as UMLet and Microsoft Visio, are often frustrating since they either don't have the capability to create every model or are annoying to use (erroneous resize, move, ...). [...] They also do not perform semantic checks, such as duplicate detection, inefficient pattern usage etc.*". Another respondent wrote "*Many "great modeling tools/methods" are only applicable [to] small systems, e.g. student projects or "hello world examples"*." Finally, "*Most modeling tools are either awkward to use, lack key features (such as good code generation) or are too expensive*" and "*there is no really good tooling available when it comes to modeling plus requirements engineering plus development plus testing plus bug tracking plus traceability*".

Lastly, it appears that there are diverging ideas about the nature of modeling. Three participants indicated that they have a fundamentally different understanding of the term "model": "*In my domain, modeling implies mathematical modeling.*" With a similarly global understanding of the term "model", another participant wrote "*programming ought to be considered just a special case of modeling*". One respondent offers a point of view often heard in the modeling community: "*modeling will allow software development to evolve from craftsmanship to a engineering discipline in the traditional way.*"

4 RELATED WORK

The related work is concerned with three closely related topics: (1) the adoption and usage of UML in industry, (2) modeling as a practice in industry, and (3) the adoption and benefit of MDE in industry. Historically, many people would consider these three topics to be a single one:

- If UML is the "lingua franca" of modeling, then (mostly) all modeling is UML modeling.
- If MDE is the prime purpose of using models, then (mostly) all modeling is MDE.

We believe this is an unhelpful confusion of separate aspects, and we aspire to help tease them apart with the present survey.

Since its advent in the early 2000's, practices like Model Driven Engineering (MDE) or Model Driven Architecture (MDA) have attracted a great deal of academic and industry attention, with several dedicated conference series and journals. There was a stream of industrial success stories like [4; 7; 34] and informal experience reports [2; 5] of individual cases. However, it took some time until researchers asked whether MDE really lived up to its promise with Mohagheghi and Dehlen's landmark paper [29].

Since then, there has been substantial research into the success factors and application conditions of MDE. With the breadth of the field, most studies focused on specific types of systems and/or specific industries, for instance [9; 27] study embedded systems in the automotive industry. Others focus on quasi-experiments to compare MDE with model-based and code-based development [28] or report industrial case studies [1]. Yet others explore industrial applications of particular MDE technologies, such as the multi-case study by Brambilla and Fraternali [8], that report repeated industrial applications in different companies over several years. Such studies have been useful in elucidating factors at play, but, of course, lack ecologic validity due to the very narrow application conditions necessarily applied. More generic studies in industry have revealed more factors, and yielded more generalizable results [20–22; 41; 42].

A second stream of related work is concerned with UML as a modeling language and how it is used in industry. This stream goes back to the studies of Dobing & Parsons who demonstrated that different parts of the UML are used for different purposes and by different people [14; 15]. These findings have been confirmed and refined by later analyzes of available models and documents, cf. [26; 35]. Probably the only piece of secondary research regarding the UML and its usage in practice is [11], which dates back to 2010.

A third stream of related work looks at how modeling actually takes place in industry. Davies at al. studied the usage of various modeling languages (including UML) in Australian Companies and already found a wide spectrum of purposes for modeling of which communication between stakeholders was the most important one [12]. Forward and Lethbridge surveyed software professionals for problems and opportunities for MDE as opposed to Code-based software development [16]. Chaudron and colleagues surveyed software architects in industry [25; 30] and asked them how they used UML. They found evidence for different degrees of formality, with most using it in a rather loose sense. Later, Baltes and Diehl investigated the usage of informal models in practice [3]. They also concluded that most purposes were related to communication and understanding.

Recently, Petre reported that *"the majority of those interviewed simply do not use UML, and those who do use it tend to do so selectively and often informally"* [31, p. 731], see also [32]. This contrasts our results to some degree: While we also find that the more informal usage modes of models are much more frequent than the more formal usage modes, we do find that UML is widely used. This usage may not be "*universal*", but there is a sizable sub-population that uses UML in industry in fully formal usage modes, such as MDE. As one respondent put it: "*there are no projects without modeling. But there are different kinds of modeling.*"

The differences between Petre's findings and ours are likely due to three factors. First, we asked for models and modeling where Petre asked for UML. Second, we considered a broad spectrum of users and usages of models where Petre seems to have focused on the very formal usages of UML, in particular, code generation, which our research confirms is the least common usage scenario. Third, there is a different regional distribution of the participants of Petre's study and ours: Petre seems to have focused on North America and the UK, two regions that are not well represented in our study. On the other hand, we have significant numbers of participants from Germany, Scandinavia, and India. It is possible, that this is the reason for the different findings. In other words, it is possible that there are regional, and indeed cultural differences in software development practices.

5 CONCLUSIONS

Based on an online survey with 96 participants from all over the world, we answer our research questions as follows.

RQ 1: Are conceptual modeling languages like UML or BPMN (widely) used in the (software) industry at all?

We can assert that models are used in industry, at least in some regions of the world. UML is, by far, the most used modeling language in the population of our participants. Even if this study allows only limited generalizations, there is no hint that any other language enjoys a similar degree of adoption.

RQ 2: When, for what purposes, and by whom are models used in industrial practice?

Models are used for a great variety of purposes by diverse stakeholders in the context software development. Interestingly, every single group of stakeholders benefits from modeling, at least to some degree, even End Users are said to benefit from modeling. At the other end of the spectrum, 91% of the respondents believed that Software Architects benefit of from modeling "a lot" or "crucially". Various roles on the fringes of software are mentioned as beneficiaries, too.

RQ 3: Are there distinct usage modes for models, and if so, how many can be distinguished?

There are three distinct modes of modeling: informal modeling for cognition and communication, semi-formal modeling for planning and documentation, and formal modeling for generation and contracts. This finding aligns with previous accounts, such as the one by Martin Fowler.

RQ 4: What is the relative frequency of the usage modes of models?

Informal modeling clearly dominates in terms of frequency (used by 70 to 79% of the respondents), followed by semi-formal modeling (43 to 57% of the respondents). However, even formal modeling does occur in relevant quantities—between 18 and 35% of the respondents use models in formal ways. So while it is correct to say that modeling in industry is mostly informal, it is wrong to say that formal modeling does not take place in industry. There is no indication that the usage profile in academia is different than that in industry.

Our findings align with much of the previous research, but contrast with Petre's findings [31; 32]. This raises the question of the validity of the previous studies (as well as the present one), and their conclusions. Obviously, any study like ours suffers from selection bias: only those people are likely to answer a survey who are interested in the topic, and have something to say about it. So, people who do not model (and possibly do not like modeling) will likely not have taken part in this survey to begin with. If this questionnaire would have been answered by the participants of Petre's study, say, our findings would probably have been different. However, the only conclusion that we derive from our observations is an *existential* one, while Petre's study seeks to support a *universal* statement, which clearly requires much stronger support.

It is remarkable, though, that the populations sampled by Petre and us have different regional distribution (Petre is unfortunately a little bit vague on this and other methodological details of her study). This could indicate regional differences in software development practices, which are likely the result of more general cultural differences, and will possibly also impact other aspects of software development practice. This is a factor that is largely ignored in the software engineering research community today—we believe, however, that this is a very interesting result of our study and merits further investigation.

It is important to notice that the biases we discuss above do not compromise our main conclusions, namely, (1) that modeling is indeed used in industry, (2) that UML is the dominating modeling language, (3) that there are several distinct modes of modeling that serve different purposes, and (4) the less formal the usage mode the more frequently it is found in practice.

Observe, however, that our study does not determine the absolute degree of adoption of either UML or modeling as a practice in industry. For such a claim, a *representative* study would have to be conducted. While some readers may our findings trivial and self-evident, we believe that the differences between modeling, UML, and MDE are not sufficiently appreciated, which is highlighted by recent publications such as [31; 32].

REFERENCES

[1] Bente Anda, Kai Hansen, Ingolf Gullesen, and Hanne Kristin Thorsen. Experiences from introducing UML-based development in a large safety-critical project. *Empir. Software Eng.*, 11:555–581, 2006.

[2] H. Andersson, E. Herzog, G. Johansson, and O. Johansson. Experience from introducing Unified Modeling Language/Systems Modeling Language at Saab Aerosystems. *Systems Engineering*, 13:369–380, 2010.

[3] Sebastian Baltes and Stephan Diehl. Sketches and Diagrams in Practice. In *Proc. 22nd ACM SIGSOFT Intl. Symp. Foundations of Software Engineering (FSE)*, pages 530–541. ACM, 2014.

[4] Ian Barnard. Increase MDD Efficiency by Customizing Tau Automatic Error Checking. Technical report, Telelogic, May 2007. White Paper.

[5] M. Belaunde, G. Dupe, and B. Nicolas. France Telecom ROI, Assessment, and Feedback. Technical report, MODELWARE Project, 2006.

[6] Grady Booch, Alan Brown, Sridhar Iyengar, James Rumbaugh, and Bran Selic. An MDA Manifesto. *MDA Journal*, 5:2–9, May 2004.

[7] M. Born, J. Hössler, O. Kath, M. Soden, and S. Saito. Significant Productivity Enhancement through Model Driven Techniques: A Success Story. In *Proc. 10th IEEE Intl. Conf. Enterprise Distributed Object Computing Conference (EDOC)*, pages 367–373. IEEE, 2006.

[8] M. Brambilla and P. P. Fraternali. Large-scale model-driven engineering of web user interaction: The WebML and WebRatio experience. *Science of Computer Programming, Part B: Special issue on Success Stories in Model Driven Engineering*, 11(89):71–84, 2014.

[9] M. Broy, S. Kirstan, H. Krcmar, B. Schätz, and J. Zimmermann. *What is the Benefit of a Model-Based Design of Embedded Software Systems in the Car Industry?*, pages 410–443. IGI Global, 2011.

[10] Jason Bryer and Kimberly Speerschneider. *likert: Analysis and Visualization Likert Items*, 2016. R package version 1.3.5.

[11] D. Budgen, A. J. Burn, O.P. Brereton, B.A. Kitchenham, and R. Pretorius. Empirical Evidence about the UML: a systematic literature review. *Journal of Software: Practice and Experience*, 41:363–392, April 2011.

[12] Islay Davies, Peter Green, Michael Rosemann, Marta Indulska, and Stan Gallo. How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, 58:358–380, 2006.

[13] Jürgen et al. Dingel. 17th Intl. Conf. Model Driven Engineering Languages and Systems (MoDELS'14). In Jürgen et al. Dingel, editor, *17th Intl. Conf. Model Driven Engineering Languages and Systems (MoDELS'14)*, number 8767 in LNCS. Springer, 2014.

[14] Brian Dobing and Jeffrey Parsons. How UML is used. *Comm. ACM*, 49(5):109–113, 2006.

[15] Brian Dobing and Jeffrey Parsons. Dimensions of UML Diagram Use: Practitioner Survey and Research Agenda. *Principle advancements in database management technologies: new applications and frameworks*, pages 271–290, 2010.

[16] Andrew Forward and Timothy Lethbridge. Problems and Opportunities for Model-Centric Versus Code-Centric Software Development: A Survey of Software Professionals. In *Proc. Intl. Ws. Models in Software Engineering (MiSE)*, pages 27–32, 2008.

[17] Martin Fowler. *UML Distilled*. Addison-Wesley, 3rd edition, 2003.

[18] Robert France and Bernhard Rumpe, editors. *Proc. 2nd Intl. Conf. Unified Modeling Language (UML)*. Number 1723 in LNCS. Springer, 1999.

[19] Andy Hunt and David Thomas. Software Archaeology. *IEEE Software*, 19(2):20–22, 2002.

[20] John Hutchinson, Mark Rouncefield, and Jon Whittle. Model-Driven Engineering Practices in Industry. In *Proc. Intl. Conf. Software Engineering (ICSE)*, pages 633–642. IEEE Press, 2011.

[21] John Hutchinson, Jon Whittle, and Mark Rouncefield. Empirical Assessment of MDE in Industry. In *Proc. Intl. Conf. Software Engineering (ICSE)*, pages 471–480. IEEE Press, 2011.

[22] John Hutchinson, Jon Whittle, and Mark Rouncefield. Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure. *Science of Computer Programming*, 89:144–161, 2014.

[23] Barbara A. Kitchenham and Shari L. Pfleeger. Personal Opinion Surveys. In Forrest Shull, Singer Janice, and I. K. Sjøberg, Dag, editors, *Guide to Advanced Empirical Software Engineering*, chapter 3, pages 63–92. Springer, 2008.

[24] Robert Kraut, Judith Olson, Mahzarin R. Banaji, Amy Bruckman, Jeffrey Cohen, and Mick Couper. Psychological research online: Report of Board of Scientific AffairsŠ Advisory Group on the conduct of research on the Internet. *American Psychologist*, 59:105–117, 2004.

[25] Christian F.J. Lange, Michel R.V. Chaudron, and Johan Musken. In Practice: UML Software Architecture and Design Description. *IEEE Software*, pages 40–47, March/April 2006.

[26] P. Langer, T. Mayerhofer, M. Wimmer, and G. Kappel. On the Usage of UML: Initial Results of Analyzing Open UML Models. In H.-G. Fill, D. Karagiannis, and U. Reimer, editors, *Proc. Modellierung*, pages 289–304. Gesellschaft für Informatik, 2014.

[27] Grischa Liebel, Nadja Marko, Matthias Tichy, Andrea Leitner, and Jörgen Hansson. Assessing the State-of-Practice of Model-Based Engineering in the Embedded Systems Domain. In Dingel [13], pages 166–182.

[28] Yulkeidi Martínez, Cristina Cachero, and Santiago Meliá. MDD vs. traditional software development: A practitionerŠs subjective perspective. *Information and Software Technology*, 55:189–200, FEB 2012.

[29] Parastoo Mohagheghi and Vegard Dehlen. Where is the proof?-A review of experiences from applying MDE in industry. In *Proc. Eur. Conf. Model Driven Architecture–Foundations and Applications*, pages 432–443. Springer, 2008.

[30] A. Nugroho and M.R.V. Chaudron. A Survey into the Rigor of UML Use and its Perceived Impact on Quality and Productivity. In Dieter Rombach, Sebastian Elbaum, and Jürgen Münch, editors, *Proc. 2nd ACM/IEEE Intl. Symp. Empirical SE and Measurement (ESEM)*, 2008.

[31] Marian Petre. UML in Practice. In *Proc. Intl. Conf. Software Engineering (ICSE)*, pages 722–731. IEEE Press, 2013.

[32] Marian Petre. "No shit" or "Oh, shit!": responses to observations on the use of UML in professional practice. *Software & Systems Modeling*, 13(4):1225–1235, 2014.

[33] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011.

[34] G. Bhaskar Rao and Keerthi Timmaraju. Model-Driven Development of HA Network elements: Experience Summary with UML 2.0. Technical report, Telelogic, October 2005. White Paper.

[35] G. Reggio, M. Leotta, and F. Ricca. Who Knows/Uses What of the UML: A Personal Opinion Survey. In Dingel [13], pages 149–165.

[36] Per Runeson, Martin Höst, Austen Rainer, and Björn Regnell. *Case Study Research in Software Engineering*. Wiley, 2012.

[37] Bran Selic. The pragmatics of Model-Driven Development. *IEEE Software*, 20(5):19–25, September/October 2003.

[38] Bran Selic. What will it take? A view on adoption of model-based methods in practice. *Software & Systems Modeling*, 11:513–526, 2012.

[39] Shaughnessy, J. and Zechmeister, E. and Zechmeister, J. *Research methods in psychology*. McGraw Hill, 9th ed. edition, 2011.

[40] Harald Störrle. Survey questionnaire and resulting raw data for the DMP4-study, 2017. available at https://figshare.com/s/86d797cc3b816e972f1b, DOI:10.6084/m9.figshare.4810600.

[41] J. Whittle, J. Hutchinson, and M. Rouncefield. The State of Practice in Model-Driven Engineering. *IEEE Softw.*, 31(3):79–85, May 2014.

[42] J. Whittle, J. Hutchinson, M. Rouncefield, H. Burden, and R. Heldal. Industrial Adoption of Model-Driven Engineering: Are the Tools Really the Problem? In A. Moreira and others, editor, *16th Intl. Conf. Model Driven Engineering Languages and Systems (MoDELS'13)*, number 8107 in LNCS, pages 1–17. Springer, 2013.