Organisation and Communication Problems in Automotive Requirements Engineering

N.B. When citing this work, cite the original published paper.

(article starts on next page)

CrossMark

ORIGINAL ARTICLE

Organisation and communication problems in automotive requirements engineering

Grischa Liebel[1] · Matthias Tichy[2] · Eric Knauss[1] · Oscar Ljungkrantz[3] ·
Gerald Stieglbauer[4]

Abstract Project success in the automotive industry is highly influenced by requirements engineering (RE), for which communication and organisation structure play a major role, much due to the scale and distribution of these projects. However, empirical research is scarce on these aspects of automotive RE and warrants closer examination. Therefore, the purpose of this paper is to identify problems or challenges in automotive RE with respect to communication and organisation structure. Using a multiple-case study approach, we collected data via 14 semi-structured interviews at one car manufacturer and one supplier. We tested our findings from the case study with a questionnaire distributed to practitioners in the automotive industry. Our results indicate that it is difficult but increasingly important to establish communication channels outside the fixed organisation structure and that responsibilities are often unclear. Product knowledge during early requirements elicitation and context knowledge later on is lacking. Furthermore, abstraction gaps between requirements on different abstraction levels leads to inconsistencies. For academia, we formulate a concrete agenda for future research. Practitioners can use the findings to broaden their understanding of how the problems manifest and to improve their organisations.

1 Introduction

In order to successfully manage the rapid increase in software size and complexity in the embedded systems domain [20], requirements need to be communicated and coordinated [8]. This need arises in particular as an organisation is broken down into independent pieces [35], i.e., for large companies which are common in this domain.

Within the area of embedded systems, the automotive industry has specific characteristics that distinguish it from other areas [37]. First, vehicles are used under greatly varying conditions, e.g. imposed by different laws in different countries, different skill levels and behaviour of drivers or variations within different cars of the same model. Secondly, demands on compatibility of subsystems are high, as components are reused across vehicle models. In particular, this means that vehicle projects rarely start from scratch but rather evolve existing specifications. Thirdly, automotive projects follow a unique design flow with multi-tiered suppliers [16]. Finally, the high degree of safety critical functions and the large production volume greatly influence the costs of errors made during development [37], making automotive projects highly cost

✉ Grischa Liebel
grischa@chalmers.se

Matthias Tichy
matthias.tichy@uni-ulm.de

Eric Knauss
eric.knauss@cse.gu.se

Oscar Ljungkrantz
oscar.ljungkrantz@volvo.com

Gerald Stieglbauer
gerald.stieglbauer@avl.com

[1] Software Engineering Division, Chalmers | University of Gothenburg, Hörselgången 5, 41756 Gothenburg, Sweden

[2] Institute of Software Engineering, University of Ulm, Ulm, Germany

[3] Volvo Group Trucks Technology, Gothenburg, Sweden

[4] AVL List GmbH, Graz, Austria

🖄 Springer

sensitive [16]. All these characteristics need to be considered during RE, which distinguishes automotive RE from RE in other areas of software or systems engineering.

While there exist several qualitative studies investigating communication and coordination challenges in RE (e.g. [18, 26]), it is unclear whether and with which frequence these challenges surface in automotive systems engineering, and how relevant they are. Additionally, it is unclear how practitioners consider to address existing challenges and whether these approaches overlap with the existing literature. Therefore, our aim is to determine what issues and solution approaches in automotive RE exist with respect to communication and organisation structure, and to relate them to existing software engineering research. In this paper, we refer to the organisation structure as the logical relations or the "decision rule connections" between people in an organisation [5]. Communication refers to the exchange of information between individuals in an organisation or between organisations. Communication can be both formal, when it follows the organisation structure, and informal, when it does not follow it [2]. As such, communication is a part of organisation structure (when formal), but informal communication is not. In the course of this paper, we aim at answering the following two research questions:

–   **RQ1**: What are current problems or challenges in automotive RE with respect to organisation structure and communication?
–   **RQ2**: Which approaches are proposed by practitioners in order to address these problems in the future?

RQ1 aims at providing indications as to which problems in automotive RE are relevant and need to be addressed with respect to organisation structure and communication. RQ2 aims at providing a picture of what practitioners are considering as possible solution approaches and what directly related work suggests.

In order to answer these questions, we conducted a case study at one automotive car manufacturer (OEM-original equipment manufacturer) and one automotive supplier. The data were collected through 14 semi-structured interviews with 15 interviewees. Of these, 8 worked in embedded software engineering, 3 in systems engineering and 3 in application software engineering. Systems engineering refers to "an interdisciplinary approach and means to enable the realisation of successful systems" [29]. In automotive engineering, it comprises mechanical engineering, electrical engineering and software engineering. Embedded software engineering refers to software engineering within the context of automotive systems. That is, it can be seen as a part of systems engineering. With application software engineering, we refer to the area of software engineering that is targeted at developing

automotive software applications running on standard computer systems, outside the systems engineering area. That is, while all interviewees worked in the automotive domain, they had different constraints and could together offer a more complete picture.

We found seven problems/challenges related to organisation structure and communication:

–   P1: Lack of product knowledge: the lack of sufficient knowledge about the product in early stages;
–   P2: Lack of context knowledge: the lack of context information regarding requirements on low levels of abstraction;
–   P3: Unconnected abstraction levels: a mismatch between requirements on different abstraction levels;
–   P4: Insufficient communication and feedback channels: lacking communication with other people within or across the organisation;
–   P5: Lack of common interdisciplinary understanding: the lack of common understanding across multiple disciplines;
–   P6: Unclear responsibilities and borders: the lack of clear and communicated responsibilities between different parts of the organisation; and
–   P7: Insufficient resources for understanding and maintaining requirements: to lack enough resources in early phases to get an understanding of the needs and to maintain requirements later on.

We further tested these through a questionnaire with 31 practitioners from the automotive industry. The questionnaire confirms that these problems/challenges occur frequently in practice and need to be addressed in the future. The found problems are overlapping to some extent with other problems identified in the related literature, e.g. in [9, 31], which strengthens the external validity of the body of knowledge in the area. Furthermore, this means that existing work on solving problems in related work could be transferred to and reused in the automotive context.

The remainder of this paper is structured as follows. In Sect. 2, we discuss how this paper aligns with related publications in the areas of large-scale, market-driven and automotive RE. The methodologies of the case study and the validation questionnaire are discussed in terms of data collection and analysis in Sect. 3. In Sect. 4, we discuss the two case companies and how the studied units handle RE. The problems/challenges are presented and discussed in Sect. 5. For each problem/challenge, we provide a detailed description of what the characteristics of each problem are and how it surfaced in the case companies. Additionally, we discuss how each problem is covered in the related literature and whether or not our data or the literature provides suggestions on how to address it. The study's validity in terms of construct validity, internal validity,

external validity and reliability is discussed in Sect. 6. The paper is concluded in Sect. 7 with an account of the findings and an agenda for future research.

2 Related work

Several publications in the area of automotive software development state that RE is one of the largest problems in this domain. For example, Broy states that a suitable RE is "one of the biggest challenges in automotive software engineering" [15]. Braun et al. [13] report that inappropriate RE is a fundamental challenge in the automotive industry and that engineering staff involved in RE mostly proceeds in an ad hoc manner. Pretschner et al. [39] present a list of research challenges for automotive software engineering, in which the first two items refer to RE. These three publications give an overview of automotive RE, but they lack empirical support.

While the previously mentioned papers describe automotive software development in general, there is work addressing automotive RE in particular, e.g. [27, 28, 42, 45]. Based on their process improvement efforts at DaimlerChrysler, Houdek and Pohl report that RE activities are closely connected and intertwined [28]. However, the authors do not name specific problems/ challenges with respect to RE. Weber and Weisbrod summarise their process improvement efforts at DaimlerChrysler [45]. They report that textual requirements alone are insufficient for handling the complexity in automotive RE. Heumesser and Houdek list a number of factors which play an important role during RE, e.g. the inclusion of external suppliers, the high number of involved stakeholders and late changes in requirements [27]. Sikora et al. investigate RE in the embedded systems domain in a case study with 17 interviewees, of which 4 are from the automotive domain [42]. They report on the current state in use of natural language, support for high system complexity, quality assurance of requirements, transition between requirements and architecture and the relation between RE and safety engineering. These four papers show why automotive RE is complex and how it can differ from RE in pure software projects, but they do not present specific problems/challenges with respect to communication and organisation structure.

We are aware of two publications within the last 10 years which present specific problems/challenges in automotive RE based on empirical data. Almefelt et al. [1] report that requirements are often incomplete or conflicting. Furthermore, the authors report that it is challenging to overview specifications due to their large size and complexity, and that following up requirements fulfilment is often more complex than eliciting them. The authors give

recommendations on how to alleviate these problems/ challenges, e.g. by eliciting a cross-system specification which summarises the most important requirements or eliciting requirements early in the process, but taking into account future changes. The problems/challenges are reported as a small part of the overall paper which focuses on describing requirements management in practice. Investigating the interface between product development and manufacturing in the automotive industry, Pernstål et al. also report problems related to RE [38]. For example, they report specific problems such as unclear requirements in early phases of a project or the complexity of communicating requirements to suppliers.

Outside of the automotive domain, there are several publications which report RE challenges with respect to organisation and communication, e.g. in large-scale RE [8] and market-driven RE [31]. In market-driven RE, Karlsson et al. [31] report challenges such as the bridging of communication gaps between marketing and developers. The same problems could apply in the automotive domain, as the development there is mainly market-driven. Similarly, automotive RE fits into the category of large-scale RE. Therefore, the communication gaps in large-scale RE presented by Bjarnason et al. [8] are relevant in the context of our paper. The authors state that the causes for these gaps are the complexity of the product, the size of the organisation, the low understanding of RE-related roles and an unclear vision of the overall goals. In [7], gaps between requirements engineering and later software development are further investigated in a systematic mapping study and 13 distances between RE and development are presented. These distances can exist between people, e.g. geographical or sociocultural distribution, or between artefacts, e.g. between requirements and design specifications. As the authors' findings are not specific to any domain, it is unclear how they apply in automotive RE. Bjarnason et al. [9] report challenges in aligning requirements with verification activities, based on a case study at six companies. Among these companies, one is placed in the automotive domain. The authors present 16 alignment challenges, of which some are directly concerned with RE, e.g. defining complete requirements or coordinating requirements on different abstraction levels. Curtis et al. [18] report in their results from an early field study that three high-level problems in software development are "(1) the thin spread of application domain knowledge (2) fluctuating and conflicting requirements (3) communication and coordination breakdowns". In particular, the authors report that problems are typically related to people and not to the tools they are using. In the area of information systems, Hansen and Lyytinen [26] report interpersonal challenges in RE, some of which are related to the problems we found in the automotive domain. Additionally, the

authors define a framework for RE challenges which we will revisit in Sect. 5 for a categorisation of our problems.

In summary, the literature originating from the automotive domain lacks empirical data, does not identify specific problems/challenges for RE or does not focus on challenges covering the aspects of communication and organisation structure. While studies presenting such problems exist in the wide scope of RE, e.g. in large-scale and market-driven RE, it is unknown whether these problems fully apply in the automotive domain. Furthermore, additional problems could arise in the automotive context. To better understand the situation in this context, this paper identifies specific problems/challenges in automotive RE with respect to communication and organisation structure. This understanding is essential to cope with increases in complexity and size in future automotive projects.

3 Research methodology

We used a multiple-case study design, following a four-step process consisting of study design, data collection, data analysis and reporting. This process is described in detail in Sects. 3.1 through 3.3.

Using a questionnaire, we tested the results of the case study. Details on its design and analysis are presented in Sect. 3.4.

3.1 Study design

The cases under study are two automotive companies, which we selected from existing research collaborations. We opted for an OEM and a supplier, referred to as Company A (OEM) and Company B (supplier), in order to achieve maximum variation in these two cases. Additionally, we chose the two in order to be able to capture inter-organisation issues from both the OEM and supplier sides. These are highly relevant in the automotive domain due to the large amount of outsourcing [15]. The case companies are described in detail in Sect. 4.

Our case study is exploratory in nature [40]; hence, it uses an inductive approach without a specified theory at the beginning of the study. Instead, we use the related work presented in Sect. 2 as a theoretical basis for the study. This is not uncommon for software engineering, as theories are underdeveloped there [40]. The literature on automotive RE was used as a point of reference to derive the interview instrument. Additionally, we used our domain knowledge from existing collaborations with the case companies in order to find suitable questions.

A case study protocol was elicited, describing the overall methodology, the interview instrument itself and further aspects, such as confidentiality and anonymity of interviewees. The methodology was reviewed by one researcher not involved in the study and discussed repeatedly between the authors of this paper. Similarly, the interview instrument was iteratively discussed and improved in several rounds.

3.2 Data collection

We used a semi-structured interview guide for data collection, consisting of 5 demographic questions and 19 questions targeting the research questions. The interview guide is published at http://grischaliebel.de/data/research/LTKSL_RE_Prob15.zip. It is important to note that the study investigated both automotive RE and the use of modelling in automotive RE. The interview therefore also contains questions aimed at the use of models. In this paper, the scope is limited to automotive RE only, excluding the aspect of modelling in RE. The interview time ranged between 37 and 72 min.

At Company A, we conducted 8 interviews. Two people were interviewed together, as one of them had recently taken over the role of the other. At Company B, we conducted 6 interviews. All interviews were conducted by the first author of this paper.

We selected the interviewees through a contact person at each company, based on their own contacts within the companies. In Company A, we selected 4 requirements engineers, 3 software engineers and 2 verification engineers, all from the area of embedded software engineering. The different roles were chosen in order to include interviewees who write requirements and interviewees who receive requirements. In Company B, we interviewed 3 employees from the systems engineering area and 3 employees from application software engineering. While the different backgrounds limit the possibility to generalise our findings to a wider context, we achieve a higher variation and thus increase internal validity.

A summary of the areas and the interviewees' work experience is summarised in Table 1. Work experience denotes the experience in their respective area at the case company. Note that interview A5 was conducted with two interviewees with 3 and 10 years of experience.

All interviews were transcribed verbatim by the first author. In the transcriptions, names were anonymised and the resulting documents sent out to the interviewees for review.

3.3 Data analysis

The overall data analysis process is depicted in Fig. 1.

The interview data were coded by the first author using an open-source qualitative analysis tool [46]. We followed the 8-step coding procedure proposed by Creswell [17], but used the following list of hierarchical *a priori* codes.

Table 1 Interviewees with experience in years

Area	Interviewees
Embedded software engineering	A1(15), A2(4), A3(15), A4(3.5), A5(3, 10), A6(7), A7(10), A8 (<1)
Systems engineering	B1(3.5), B2(1.5), B6(4.5)
Application software engineering	B3(36), B4(27), B5(4)

1. Current State-of-Practice
2. FutureImprovement
3. Problem
4. Process

    (a) Clarification
    (b) Communication
    (c) Elicitation
    (d) Guidance
    (e) Inter-Organisational
    (f) Intra-Organisational
    (g) Measurement
    (h) Modelling
    (i) Refinement
    (j) Requirements Usage
    (k) Verification and Validation

5. Requirements

    (a) Ambiguity/Understandability
    (b) Notation
    (c) Level of Abstraction
    (d) Quality
    (e) Reuse
    (f) Tooling
    (g) Tracing
    (h) Variability

These codes were elicited as keywords from the different aspects addressed in the interview guide and related work, discussed among the authors, and improved based on the discussion. Furthermore, one interview was coded in a pilot attempt, after which the code set was revised.

During the coding procedure (Step 1 in Fig. 1), we applied instances of these 24 codes to the transcriptions, where each statement was assigned one or many code instances. Essentially, using this procedure, each statement made by an interviewee is enhanced with keywords (the codes). These keywords can then be used to extract only the statements relevant for data analysis. For example, if an interviewee talked about how requirements are clarified within the organisation using use case diagrams, we would assign the codes *Current State-of-Practice, Clarification, Intra-Organisational and Notation*.

After the assignment, we filtered the coded data according to the following code combinations following the study's aim (Step 2 in Fig. 1):

(Problem AND Communication)

OR (Problem AND Inter-Organisational)

OR (Problem AND Intra-Organisational)

That is, all statements which address problems in the area of communication and organisation (further refined into inter-organisational and intra-organisational problems) are extracted from the raw interview data.

This yielded 88 statements from Company A and 66 statements from Company B. The overall set of 154 statements were then abstracted (Step 3 in Fig. 1), removing concrete examples or company-related explanations. This enabled us to compare statements to each other and group them into categories (Step 4 in Fig. 1). This categorisation resulted in 7 major categories, which were supported by more than one interviewee. The category descriptions were sent to the interviewees for feedback and
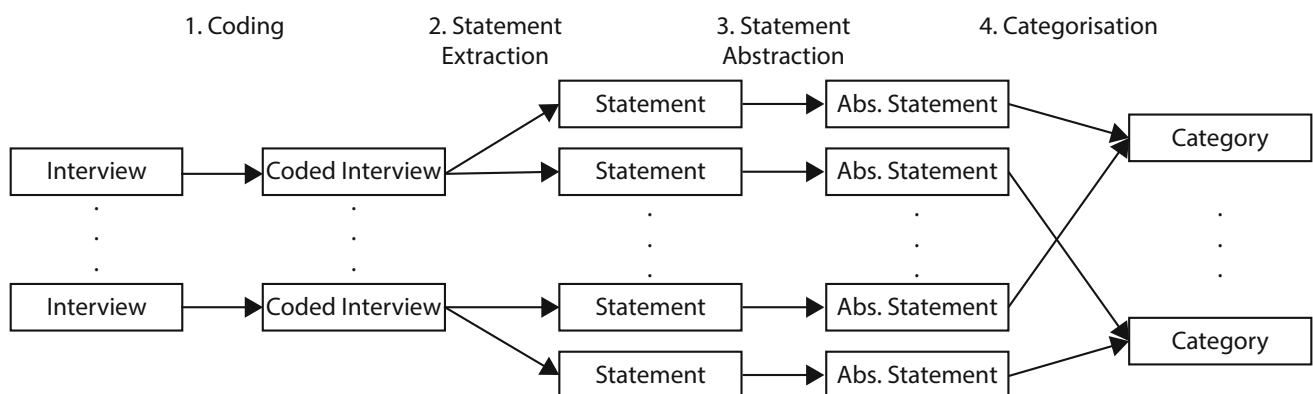
Fig. 1 Data analysis procedure

discussed with the contact persons at the case companies. This process resulted only in smaller changes to the category descriptions, but all interviewees and contact persons agreed that the categories described relevant problems in practice.

Finally, we compared the categories to similar descriptions brought up in the related work. From both interview data and matching issues in related work, we extracted potential solutions. These are not intended as an universal means to address each problem, but to outline different approaches towards a specific solution in practice.

3.4 Validation survey

After extraction of the seven problems/challenges, we constructed a questionnaire in order to validate our findings. The questionnaire contained 10 demographic questions in order to be able to assess potential differences between survey participants. The questions addressed the country in which the participants work, the company they work for, as well as its size and position in the value chain, their experience in the automotive industry, their current role, the subdomain in which they work and how they get into contact with requirements. In order to be able to distinguish interviewees and other survey participants, we also asked whether or not participants have been part of the interviews.

After the demographic part, we asked which key problems the participants currently face in requirements engineering using an open question. In order to not influence the participants, we did not mention our focus on communication and the organisation structure until the page after. Also, this focus was neither mentioned in the introduction text of the survey nor in the invitation email. Afterwards, we asked participants to rate their agreement to the following two statements on a five-point Likert scale:

1. Key problems in automotive RE lie in communication.
2. Key problems in automotive RE lie in the organisation structure.

We provided the definitions for communication and organisation structure together with this question (as defined in Sect. 1).

Finally, we provided participants with the definition of all seven problems/challenges we extracted from the interview data. For each problem/challenge, we then asked them to state (a) how often they experience it and (b) to rate their agreement to the statement that it is indeed a key problem which needs to be addressed. For (a), participants had to choose on an ordinal scale with the values *Never, Yearly, Monthly, Weekly, Daily*. The agreement in (b) was rated on a five-point Likert scale.

An initial version of the survey was discussed among the researchers involved in this paper. After corrections, we piloted the survey with two additional researchers not involved in the study and with five practitioners from the automotive industry. The final version of the questionnaire is available for download together with the interview guide (see Sect. 3.2).

The survey was initially made available for a period of 2 months. As it had an overlap with the summer holidays in several European countries, it was later extended for three weeks. We distributed the survey to personal contacts at automotive companies in Europe, North America and Asia and encouraged them to distribute it further within their network.

In total, we received 42 answers to the survey. We filtered all participants who did only answer demographic questions. This left us with 31 surveys for data analysis. We explain the low amount of answers mainly with the holiday period and the long time (approximately 20 min) it took to answer the questionnaire. Several contact persons stated their engineers did not all speak English well enough to understand and answer the survey.

4 Case companies

We conducted the presented case study at two case companies. We extracted an overview of the current RE processes and practices at both companies from the interview data. In the following, they are described in more detail.

4.1 Company A

Company A is a global automotive OEM based in Sweden. In this company, we performed all interviews in a single department, where a large part of the embedded software development for vehicles takes place.

Within the studied department, high-level requirements for each project usually come from product planning, which is outside of the department. These requirements are usually user oriented and often vague, which is partially intentional in order to leave room for creativity during implementation.

In the department, the high-level requirements are broken down into smaller parts and assigned to employees whose functions are affected. The employees break down the requirements into logical components. These are on a very detailed level, often close to pseudo-code. The resulting component specifications are then handed over to in-house development or used as a contracting document for external suppliers.

In the department, requirements are generally specified in natural language text, while pictures of state machines

are sometimes used for clarification. There is no general opinion among the interviewees whether natural language specifications should be replaced by formal or semi-formal descriptions. Some interviewees stated that producing formal specifications can lead to accidental designs, while others were clearly in favour of executable specifications.

As soon as the requirements are broken down, the department's test organisation is starting to prepare the verification activities in parallel with development. This includes, e.g. to write test cases or prepare models of the environment for testing purposes. The models and tests are developed independently of the source code for the actual software.

The overall vehicle specification is usually kept and evolved throughout projects. That is, specifications are modified over multiple projects and not written entirely from scratch. For storing the specifications, logical and physical designs and test cases, the department uses a systems engineering tool. Traceability is good between everything that is kept within the tool. Everything which is outside is currently not traced, e.g. the high-level requirements from product planning. The tool is currently only used within the department, while Word and PowerPoint are common formats for documenting requirements in other departments as well as for documenting high-level requirements.

The development currently works in an agile way in 8-week iterations on system level, with shorter sprints on subsystem and component levels.

Even though there are challenges as in every organisation, interviewees highlighted that they are, after all, successfully producing vehicles. Also, several interviewees stated that a lot of improvements have taken place over the last years and that the company is constantly moving forward. For example, there are several pilot and R&D projects which regularly try out new things, such as introducing executable requirements specifications, or quality gates and handshakes throughout the development process.

4.2 Company B

Company B is an Austria-based automotive supplier developing powertrain systems, as well as simulation and test bed systems. The company has both market-driven projects and customer projects, where the requirements come directly from an OEM. The process is different for the two different parts of the company that were studied, *systems engineering* and *application software engineering*.

In *systems engineering*, the development follows a process related to the waterfall model, with smaller differences between the departments and units. Generally, projects start with a user requirements specification (URS) coming in from different sources. The URS is analysed in the company by the customer relations unit, and a System Requirements Specification (SRS) is elicited to meet the URS requirements. The level of detail and the quality of the received URS vary depending on the customer. However, the customers generally already have experience with the products offered by the company and know their own use cases.

In some parts of *systems engineering*, SysML is used to specify requirements and design later on, while other parts plan to introduce this in the future. There are currently several piloting efforts in this direction, and the feedback from parts that introduced SysML is positive. However, interviewees also stated that natural language for requirements remains important as a way to express uncertainty.

In *application software engineering*, a waterfall process was followed for a long time. It has recently been replaced by an agile process following the SAFe framework [41]. The interviewees were positive towards this newly introduced process, as far as they could already judge it. In particular, the fact that communication had improved significantly was mentioned several times. However, it also means that experiences from this new process are preliminary. Therefore, certain challenges might disappear over time as people get used to the new process.

An incoming URS is first handled by the customer relations unit and then handed over to product management, if the current product range does not already fulfil the request. Product management has the task to understand the problem of the customer, with the help of domain experts and fit it to the company's current product range. In particular, an incoming URS is often covering multiple products, as the customers only describe their problems and needs, which do not necessarily adhere to single products. From the received URS, an SRS is derived, usually together with the stakeholders. For customer projects, the SRS is then used as a contracting document.

Requirements on all levels are written in natural language. This is intended to keep the specification effort low. Initially, the so-called business stories are specified and then refined or broken down. In the first step, requirements are broken down into epics. Epics have a customer value and should be implementable within one release cycle of 6 months. The next level of granularity is the feature, which should fit into one iteration of 10 weeks. Features are picked by the development teams, consisting of developers and the development owner. The teams then take ownership of the features and break them down into user stories, which can be implemented in a single 2-week sprint. Every item, including the acceptance criteria, are stored in a JIRA repository [3] and traced to the other levels. Acceptance criteria are written by the receiving party for all levels of granularity. The receiving team only accepts the item if they understood it.

Testing is done on multiple levels, but a large part of the functionality of software modules should already be tested by the development teams before it is integrated with other modules. On a system level, there exists a team responsible for testing the integrated parts. Additionally, outside the unit, there is a unit focused on testing the entire product range as a whole.

5 Results and discussion

In the following, we describe each of the seven identified problems/challenges and discuss them in relation to our cases. These arose as distinct problems/challenges in the statement categorisation, as described in Sect. 3.3. We critically discuss them in relation to similar statements in related work and list potential solutions that interviewees and directly related studies brought up. The solutions are not backed up by empirical data on their effectiveness, but help to understand better the interviewees' reasoning about the problems.

5.1 Identified problems/challenges

The seven identified problems/challenges with respect to the organisation structure and communication are summarised in Tables 2, 3, 4, 5, 6, 7 and 8. The *Area* row depicts the percentage of interviewees who brought up the problem in the areas of embedded software engineering (EmbSE), systems engineering (SysEng) and application software engineering (AppSE). We do not classify the problems into communication problems and organisation problems, as the categories overlap and each problem could potentially be in either category depending on the context.

5.1.1 P1: Lack of product knowledge

Several interviewees in both the embedded software and the systems engineering area mentioned that it was challenging to specify requirements in the early phases of a project. For instance, multiple interviewees in embedded

software engineering stated that people eliciting these requirements often lack product knowledge. This lack of experience is often a result of short-term contracts with consultancies, or of organisation restructuring. Another reason for a lack of product knowledge is that requirements come from many different stakeholders, e.g. regulatory bodies, and differ depending on the country or region in which the automobile should operate. Furthermore, technology is getting outdated quickly, which makes it difficult to know which features the product should have once it is completed, after several years of development. This lack of knowledge can lead to vague or unclear requirements.

At Company A, there is little communication or clarification between product planners who elicit the high-level requirements and the studied department. Hence, unclear requirements cannot be discussed or refined, and both product planners and the receivers of high-level requirements have to rely on their own domain knowledge. Consequently, many high-level requirements are either obvious or obsolete, but they are not changed later on. This leads to a large overhead for verification as the process requires that all high-level requirements need to be verified.

In the systems engineering area, P1 is mostly related to the requirements received by customers. The customers often do not understand their problem and the products offered by Company B. Hence, they simply state the solution they would like to receive in the end. Additionally, there might be a general lack of understanding due to the novelty of emerging technologies, such as hybrid engines. Therefore, it is difficult to specify requirements for these novel products. Lastly, short contracting times can cause pressure to elicit requirements in a rushed fashion, without sufficient clarification of the customer's problems.

Potential solution In the embedded software area, one interviewee suggested to have executable *models* to use as a feedback mechanism and to reach a shared knowledge with higher levels in the company. An executable model could be used as a feedback mechanism for sharing knowledge of the product.

Similarly, interviewees in the systems engineering and in the application software area mentioned the creation of
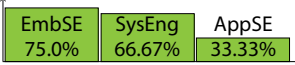
Table 2 P1: Lack of product knowledge

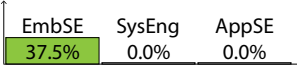| Name	Lack of product knowledge
Short description	People eliciting requirements need a large amount of domain knowledge. Additionally, in early phases of a project, a high level of uncertainty regarding the final product is common. This can lead to vague and unclear, or even obsolete, requirements with greatly varying quality
Areas	EmbSE 75.0% / SysEng 66.67% / AppSE 33.33% [38, 31, 8, 1]
Related work	[1, 9, 31, 38]
Addressed by	Handshaking, increased communication, agile processes, customer involvement, cross-functional teams, models

Table 3 P2: Lack of context knowledge

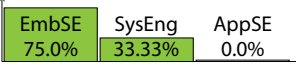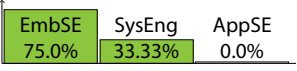| Name	Lack of context knowledge
Short description	Requirements specifications can easily be several hundred pages long. While they can be overlooked to some extent on a higher level, it is difficult to provide developers or testers on lower levels with enough context information, especially when development is outsourced. Therefore, when receiving requirements, it is problematic to acquire enough knowledge regarding the requirements' context, in order to understand them and the rationale behind them
Areas	EmbSE 37.5% SysEng 0.0% AppSE 0.0%
Related work	[1]
Addressed by	Agile processes, increased communication, SRS summaries, linking documentation and requirements, increased traceability

Table 4 P3: Unconnected abstraction levels

Name	Unconnected abstraction levels
Short description	Early in the project, there is a need to document requirements on an abstract level. On the other end, especially because of the tight coupling to hardware and the highly distributed systems with possibly multiple external suppliers involved, there is a need to describe requirements on a low level of abstraction, close to pseudo-code. However, it is often unclear how these low-level requirements have been derived from the requirements on a high abstraction level. This can result in incomplete tracing, obsolete high-level requirements, or a mismatch between high-level testing and the implementation
Areas	EmbSE 75.0% SysEng 33.33% AppSE 0.0%
Related work	[9, 31]
Addressed by	Documented decisions, higher abstraction, cross-functional teams

Table 5 P4: Insufficient communication and feedback channels

Name	Insufficient communication and feedback channels
Short description	With the high complexity of today's automotive systems, it is not feasible to document every single detail in terms of requirements. This results in a need to communicate, ideally between different organisation levels and roles. It is however not an easy task to establish and maintain the right communication and feedback channels within and across organisations
Areas	EmbSE 75.0% SysEng 33.33% AppSE 0.0%
Related work	[31]
Addressed by	Agile processes, models, formal reviews, defined language

graphical models as a "thinking tool", to better understand the system. In this context, one interviewee stated that the use of models should primarily aim to increase system and problem understanding, communication and serve as documentation. Furthermore, the interviewee stated that these models should not be on a level of detail where they could be used for purposes such as code generation or simulation, due to the effort required to create and maintain them. This statement is highly interesting given that a large amount of the efforts in the modelling community are aimed at enabling reuse of models for exactly these purposes. For example, the vision model-driven engineering aims at transforming abstract models systematically into a concrete implementation [22].

In the systems engineering area, P1 is mainly addressed by increasing the amount of *communication* with the customer and internally along the whole development chain.

Similarly, the application software area uses an agile process, which favours a large amount of communication within and across development teams.

The existing literature offers several related problems or challenges, together with suggestions on how to address them. Pernstå* et al. state that "ambiguous product requirements from pre-studies [..] cause difficulties in interpreting and implementing correct functionalities [..]" [38]. In their specific context, namely the interface between development and manufacturing, the authors mention that interviewees expressed difficulties "in constraining manufacturing requirements to be understandable". That is, to express requirements in a way that is easily understandable by the people receiving the requirement and, at the same time, in a correct way. As a possible countermeasure, they refer to Fricker et al.'s [23] proposal to use *handshaking*, where product management acts as a

Table 6 P5: Lack of common interdisciplinary understanding

Name	Lack of common interdisciplinary understanding
Short description	In the automotive industry, several disciplines are involved in the development. While there are many connection points and while there is a clear need for communication and exchange, most disciplines have their own understanding and their own language. If these languages do not match between disciplines, this can lead to misunderstandings
Areas	EmbSE 12.5% SysEng 66.67% AppSE 33.33%
Related work	[38]
Addressed by	Increased training, models, knowledge codification, job rotation

Table 7 P6: Unclear responsibilities and borders

Name	Unclear responsibilities and borders
Short description	Automotive systems are nowadays developed by numerous globally distributed teams with a high staff turnover. These teams need to communicate and need to be coordinated. Additionally, they are all responsible for a part of the overall system under development. It is difficult to draw organisation borders between these teams, units, or roles and assign clear responsibilities to individuals
Areas	EmbSE 62.5% SysEng 100.0% AppSE 66.67%
Related work	[9, 38]
Addressed by	Formalised roles, job rotation, cross-role requirements reviews

Table 8 P7: Insufficient resources for understanding and maintaining requirements

Name	Insufficient resources for understanding and maintaining requirements
Short description	Time and money are always pressing factors. Additionally, systems are often built incrementally and built to live for a long time. However, early project phases, where many of the decisions are taken, are often short. Therefore, it is problematic to get a sufficient understanding in early project phases and to maintain or even improve the quality of the specifications later on
Areas	EmbSE 25.0% SysEng 33.33% AppSE 33.33%
Related work	[9, 31]
Addressed by	None

customer towards development and has to accept solution proposals offered by development. This approach would however impose additional effort on projects, especially when the process is not aligned, i.e. when receivers of requirements are still working on other projects while the requirements are written. Therefore, introducing handshaking could require large process changes in an organisation.

Similarly, one interviewee in the embedded software area in our data proposed to change the process from a push process to a *pull process*. That is, instead of delivering requirements, which then need to be processed by the requirement receivers, they could be requested from the receivers. This would avoid that time is spent on requirements that are not needed later on. However, requirement receivers might in some cases lack the overall picture and therefore miss important requirements.

Karlsson et al. [31] mention that writing natural language requirements in an understandable way is generally

challenging. The authors state that this problem is not solvable, and therefore, a discussion will always be needed in addition to written requirements. That is, the authors encourage explicit communication regarding individual requirements. It is unclear whether this is attributable to natural language only or if it is in fact caused by the lack of knowledge that needs to be written down. If the former was the case, semi-formal or formal specification techniques could be used. However, we believe that the challenge lies in the uncertainty in early phases. Therefore, discussion and volatile requirements will always be present in automotive RE.

Defining clear requirements and writing complete requirements are two challenges mentioned by Bjarnason et al.'s [9] study on alignment of requirements and verification. The authors list a number of practices that can address these problems, namely *customer involvement*,

having *cross-functional teams* for requirements specification and cross-role requirements reviews.

While the solution proposals from our interviewees and from the cited related work could cover some aspects of the challenge, writing requirements for emerging technologies is inherently more complex. Here, the uncertainty is not caused by a single person who might lack domain knowledge. Instead, it is a lack of knowledge of how the market or technology will evolve in the future. These types of requirements cannot be addressed by involving development and/or customers. Therefore, Almefelt et al. [1] recommend that one should be open to changes in the requirements even if these are elicited early on. Similarly, Karlsson et al. [31] name this "requirements volatility." The authors suggest that the iterative nature of *agile processes* could address this problem/challenge, as only few requirements are written down at an early stage. However, it is important to note that agile on a large scale [19] and especially in regulated environments [21] remains challenging. As automotive engineering is both large-scale and regulated, and as it combines software development with traditional engineering disciplines, agile processes should not be easy to introduce.

5.1.2 P2: Lack of context knowledge

It is difficult to provide recipients of a requirement enough background information in order to understand the meaning of and the rationale behind it. Often, this is caused by the size of the requirements specification and the low level of detail of single requirements. The level of detail is however needed to ensure compatibility between subsystems developed by different organisations. When background information is lacking, it is difficult for engineers to reason about a requirement, e.g. to choose the most appropriate way for implementation or to ask for further clarification.

This problem/challenge is even stronger when requirements are exchanged across company borders and parts of the overall information is not available to subcontractors. This is often the case when OEMs try to protect their intellectual property and therefore hold back details.

Several interviewees in the embedded software area reported this challenge. For instance, some of them only receive the specification for the functionality or the component they are responsible for, but do not receive any information on how it should function within its environment.

Iterative development can be one cause of P2, as one of our interviewees observed. If information is not shared appropriately, developers might only receive part of the overall requirements in each iteration and consequently lack the overall picture.

In the application software and the systems engineering areas, the problem was not reported by any of the interviewees. The increased communication introduced together with the *agile process* could be a reason why the problem does not arise in the application software area. Furthermore, the lack of subcontractors in this area facilitates communication, which could be another reason for the absence of P2.

In contrast to P1, P2 surfaces when the specification already exists. P1 occurs during requirements elicitation or specification instead. However, they are closely related and could possibly be combined into a single problem/challenge.

Potential solution In-house at Company A (EmbSE), the problem/challenge is addressed by *increasing the amount of communication* with higher levels within the same department. However, it is often challenging in itself to find the right person to talk to. That is, increasing communication could be unsuccessful due to P4 (insufficient communication and feedback channels).

As another way to address the challenge, interviewees in the embedded software area stated *increased traceability* between requirements, especially to higher levels of abstraction. If a clear trace link is provided to a higher level of abstraction, it is possible for developers to understand the context better.

Additionally, one interviewee stated that high-level function documentation should be provided to developers. Similarly, one interviewee in the application software area mentioned that *linking the documentation to the requirements* could be a potential way to improve understanding on lower abstraction levels. A potential risk we see with this solution approach is that the documentation could be equally outdated as the high-level requirements, hence leading to problems similar to P3 (unconnected abstraction levels).

Almefelt et al. [1] report a problem close to P2, namely that it is difficult to give a complete overview of a typical SRS, due to the large complexity in automotive development. The authors therefore recommend to provide cross-disciplinary *specification summaries*, containing the most important requirements.

5.1.3 P3: Unconnected abstraction levels

As mentioned in the description of P1, high-level requirements are often vague and abstract as not every detail should be specified early on. On lower levels of the system hierarchy, requirements are typically very technical and detailed in order to facilitate system integration. This level of detail is intentional, as hardware and software are tightly coupled and as multiple subcontractors are usually

involved in the development of hardware and software components. In between the two different abstraction levels, there is in many cases a gap.

This gap is not problematic as long as the connection between the levels is clear. However, as high-level requirements are vague, it can happen that engineers add details to the requirements or correct them, without communicating it. In this case, there are no clear connections between the requirements on different abstraction levels any longer and they might be contradicting in the worst case.

Interviewees in the application software area did not report this problem/challenge. This might be attributed to the fact that the agile process they introduced forces teams to define acceptance criteria for each requirement they receive from higher levels. If they cannot define these, the owners of the higher-level requirements are forced to clarify them and make changes, if needed. This process forces engineers to break down requirements in a defined way with no gaps in between, as each level has to be testable and clarified.

Potential solution The interviewees in the embedded software area disagreed on how a solution to this problem should look like. Some stated that the low level of abstraction for component-level requirements should be removed and requirements should instead be formulated on a *higher abstraction* level. Others, especially people working with software development on that low level, stated that the low abstraction level is needed. In particular, they stated that previous projects, which started on a higher abstraction level, encountered issues during the integration of several components. Instead of a higher abstraction, they proposed that developers should work with verification and requirements engineers in *cross-functional teams*. This cooperation could then increase communication, and low-level textual requirements could be replaced with executable models on a higher level.

In the literature, Bjarnason et al. [9] mention that it is problematic to coordinate requirements on different abstraction levels. To address this challenge, the authors recommend to *document decision rationales*, i.e. why and how a requirement was broken down to a more detailed level. In the embedded case, this would address only part of the issue, i.e. it would become clear which details are added along the way by the engineers. However, it would not address corrections made by engineers on lower levels, as high-level requirements would still be faulty and the mismatch would remain. Also, it is interesting to note that our interviewees suggested measures which increase communication, whereas Bjarnason et al. propose an increased amount of specification.

We are not aware of any further related work explicitly addressing this particular issue. However, how to break down or refine requirements is a common topic in software processes and in RE in particular.

5.1.4 P4: Insufficient communication and feedback channels

In all three studied areas, the need for communication and feedback was mentioned as an important aspect of RE and while using requirements later on in development and verification. This need arises as the complexity of products and organisations in the automotive industry is typically too high in order to document every aspect in written form. The evolving nature of requirements in the automotive industry complicates this further, as the original authors of the requirements might have moved to other projects, or even left the company.

In the embedded software engineering area, interviewees stated that there are not always specific contact persons, or they are unknown, for parts of the system. This is strongly related to staff turnover, as responsibilities can change quickly.

The interviewees also brought up that there is currently no interaction with higher levels outside the department. That is, requirements coming from these levels cannot be clarified or changed.

Finally, interviewees stated that feedback loops with suppliers are difficult to establish and maintain. For instance, due to contracts between the OEM and the suppliers, changes or clarifications in the requirements have to be handled officially through change requests. This hinders more informal communication between the two parties to take place.

Most interviewees in both areas at Company B did not report P4 as an issue, but instead mentioned that communication is one of the main important aspects to tackle complexity and ambiguous requirements.

Potential solution In application software engineering, the problem/challenge occured before the change to *agile development*. This change leads to increased communication among developers and with customers, instead of formal documents, and thus solved the problem according to the interviewees. In contrast, the systems engineering area uses a strict processes with *formal requirements reviews* in order to make feedback and communication explicit. These two contrasting approaches reflect well the bandwidth in the automotive domain and that it is not always possible to find a common solution. While application software engineering is facing short development cycles and often no strict requirements on safety criticality, the studied parts of systems engineering have long-running projects with high standards on safety criticality for control software.

In the embedded software area, several interviewees stated that the situation improved considerably over the last years, as *communication channels* were increasingly established between different parts of the department. That is, they chose to establish fixed communication channels crossing organisation borders, through which requirements changes are discussed as a means of formal communication.

Karlsson et al. [31] mention the communication between the marketing and the development departments. In their case, the departments did not communicate sufficiently and had different views on how requirements should be written. This lack of communication resulted in a large overhead for specification. The authors list as a possible solution to establish a *common defined language for exchange and discussion* [31], similar to the suggestion to introduce models in order to address P1. However, it is questionable how a common language would address the conflicting opinions on what a good requirement constitutes or how they should be written.

A solution which was neither mentioned by the interviewees nor related work on requirements challenges are collaborative workshops or tools. For example, Gottesdiener proposes to organise requirements workshops to get requirements right from the beginning [24]. Similarly, Kylmäkoski proposes authoring workshops for documentation [32]. However, in the case of the automotive industry, the evolutionary nature of requirements would have to be taken into account. This would ask for repeated workshops for requirements clarification instead of just for authoring or elicitation. Leaving the idea of pure workshops, integrated collaboration mechanisms in tools [43], or the creation of cross-team communities of practice [30] also target improved communication.

5.1.5 P5: Lack of common interdisciplinary understanding

Several interviewees reported communication between the multiple disciplines involved in the development as challenging, dominantly in the systems engineering area at Company B. While the disciplines often have common grounds, such as common terms, all speak their own language and the same term can have entirely different meanings in two disciplines. Furthermore, the sheer amount of different stakeholders, such as regulatory bodies in different countries, or multi-tiered suppliers complicates this situation. Avoiding misunderstandings is therefore difficult.

Within the application software area, this challenge was reported in context of the task to understand a customer's requirements. In many cases, customers come from a discipline different than software engineering, which complicates communication and mutual understanding.

As an additional aspect of this problem, one interviewee mentioned the challenge of fitting RE into an organisation, i.e. perform RE in a cross-organisational, interdisciplinary function. The different disciplines are often separated by organisation borders and have different views of the system, e.g. the production and functional view. However, system requirements often span several disciplines and views. While this can be a rather straightforward task for a company that develops software only, multiple disciplines often have their own units within the company, and therefore, RE can be heavily distributed as well.

Potential solution The interviewees in the systems engineering area mentioned that they plan to introduce model-based systems engineering using SysML as a modelling language. That is, introducing *models* to describe functionality, including for requirements. Using UML profiles, the language could be adapted to the company's vocabulary and define fixed language concepts which are then used across disciplines. As stated for P1, the requirements models are mainly intended as a "thinking tool", not a detailed model which can be used for other purposes, such as simulation. Therefore, it is important that all stakeholders can understand the model in the same way.

Additionally, interviewees mentioned that offering *sufficient training* and building up employees who are able to communicate across disciplines is currently in their focus. This solution proposal is highly interesting for research, as it raises the issue of cross-disciplinary curricula. Not only are future engineers required to have detailed knowledge of one discipline, they also need to communicate effectively across disciplines.

Multiple disciplines are often used to motivate the difficulty of automotive engineering in general, e.g. in [37]. However, P5 is only mentioned specifically as a challenge in Pernståls et al.'s study [38], as an example for the more general issue of knowledge development throughout the organisation. The authors state that there is a lack of understanding for hardware and software development outside the developing units. As potential solutions, the authors name *knowledge codification* and *job rotation*. In knowledge codification, gathered knowledge is explicitly documented, e.g. through experience factories [6]. This clearly requires substantial effort and might not be suitable for communication across disciplines or organisation boundaries [4]. In job rotation, employees change role in order to increase their understanding of the organisation. Similarly to knowledge codification, it is unclear whether this would work across disciplines, as the knowledge required in different disciplines will prevent most employees from rotating.

5.1.6 P6: Unclear responsibilities and borders

Due to their size and dynamicism, it can become difficult to draw clear borders and define responsibilities in large organisations. This can have several effects, such as rivalry

between teams with overlapping functionality or neglection of tasks if no one is feeling responsible for them. Also, the large amount of multi-tiered suppliers and dependencies among them makes interaction between organisations challenging. In contrast to P4, this problem/challenge addresses the defined organisation structure, not the emerging communication between different subjects.

In all three areas, this issue was brought up by multiple interviewees. In the embedded software area, e.g. interviewees stated that it is not always clear which requirements are tested by which part of the test organisation. Several interviewees in the application software area mentioned multiple systems with similar functionality that exist in the company. Here, instead of benefiting from each other's knowledge, rivalry can ensue between the different groups.

Potential solution In the literature, Pernstål et al. [38] state P6, but in a different context. They report that there are "unclear definitions and allocations of roles and responsibilities and difficulties in understanding who is accountable for what" between product development and manufacturing. The authors suggest a *formalisation of roles*, following Vandevelde and Van Dierdonck [44]. In the context of P6, this could address unclarities. However, ensuing rivalries between organisation units would have to be dealt with separately.

Bjarnason et al. [9] name the challenge of aligning goals and perspectives within an organisation. The authors state that this can complicate coordination and communication among different units in the organisation. This challenge can be related to P6, as responsibilities and organisation borders should be aligned with the (sub-)goals in an organisation. The authors suggest several ways to address the challenge, e.g. *job rotation* or *cross-role requirements reviews*.

Additionally, one of the contact persons suggested to increase the understanding of the process and the roles within an organisation. This could lead to employees who are more willing to solve existing problems, even if they are not formally responsible. However, increasing the understanding would also increase cost. Therefore, it has to be compared against the potential benefit of this step.

5.1.7 P7: Insufficient resources for understanding and maintaining requirements

Insufficient resources are a fundamental challenge in most businesses. In our study, the problem/challenge was brought up by two interviewees in the embedded software area with respect to the existing requirements base. In their case, the requirements specification is not rewritten for every project but instead constantly evolving. However, there is no budget specifically allocated to review existing requirements. Therefore, there is a lack of resources to improve the quality of existing requirements. While several interviewees expressed that older requirements did not have sufficient quality, they could not change this either.

Interestingly, this issue is directly related to the matrix structure in many organisations. As requirements span across products or projects, they would have to be maintained and improved outside of specific car projects. However, there is no budget allocated for such projects.

Potential solution Among our interviewees, no one offered a potential solution to P7. This is potentially related to the fact that the problem/challenge is related to the matrix structure of the organisation. Therefore, it should be solved on a managerial level and not within the actual product development.

In the related work, Karlsson et al. [31] state that a lack of resources for RE is problematic. While the authors report that RE is underestimated in terms of costs and therefore often lacking resources, they do not mention the maintenance of requirements. However, this task could take up an even larger part of the resources than the actual requirements elicitation. The authors do not offer concrete solution proposals for this lack of resources.

Bjarnason et al. [9] report the challenge of keeping requirements updated. This challenge is addressing the evolution of the requirements, but not general quality improvements in requirements. However, their proposed solutions, e.g. involving testing in change management, do not address the challenge in terms of a lack of resources.

5.2 Problem/challenge context

Each problem/challenge was brought up in a certain operational context within our data. Problems/challenges P1, P2, P3 and P5 can be related to specific phases in the project context. Lack of product knowledge (P1) and common interdisciplinary understanding (P5) were only brought up in the context of the early project phases. Similarly, a lack of context knowledge (P2) was stated to occur in the low-level details and in combination with outsourced development. Unconnected abstraction levels (P3) occur from the high-level requirements on product-level down to logical components on system level. While these four problems/challenges could be relevant in other project phases, our interviewees only reported them as stated in the descriptions.

The other three problems/challenges were brought up with respect to different situations throughout the whole project life cycle. This makes them cross-cutting, which means they should be addressed in a general way irrespective of the project phases.

Also, the problems/challenges are not isolated, but can be caused by one another or cause other issues when

occurring together. For example, interviewees in the systems engineering area stated that when misunderstandings are common because of P5, communication channels are difficult to maintain (P4). Additionally, as high costs are associated with intensive communication across organisation borders, insufficient resources (P7) can cause insufficient communication (P4) and a lack of product knowledge (P1). Furthermore, when high-level requirements are not appropriate, due to a lack of product knowledge (P1), and when they are not discussed or improved due to a lack of communication (P4), abstraction levels can become unconnected (P3).

While we do not have support for further relationships between the problems/challenges, some of them seem logical. For instance, having insufficient resources (P7) can be seen as a root cause which can potentially lead to any of the other problems/challenges. That this problem is inherently complex might also be depicted by the fact that there was no potential solution offered by neither the data nor the related literature.

While many of the interviewees recalled situations from projects, the problems are not necessary related to a single project. For instance, having insufficient resources (P7) is possibly related to the matrix structure in many organisations, as discussed in the problem description. These problems are not related to the development process and therefore also relevant if the process changes, e.g. when transitioning from a traditional waterfall-like process to agile methods or continuous delivery.

Additionally to the company or project context, we tried to classify the problems retroactively according to three different schemes/models proposed by Bjarnason et al. [10], Hansen and Lyytinen [26], and Curtis et al. [18]. As the interviews or the validation survey was not designed with these models in mind, this classification can only be discussed on a high, subjective level with the intention to assess the suitability of the models in our context. Bjarnason et al. [10] propose a model of eight distances, namely geographical, organisational, psychological, cognitive, adherence, semantic, navigational, and temporal distance. We consider this model here as it is inspired by [7], which is part of the work directly related to our contribution. Several of these distances are clearly relevant for our findings. For example, based on our interview data it seems likely that P1, lack of product knowledge, is caused by an organisational gap (no contact between product planning and studied department in Company A), a cognitive gap (competence levels not sufficient for eliciting product requirements), and a temporal gap (people who elicit high-level requirements have already moved to the next project once problems arise). While we do not see any occurence of psychological, adherence, and navigational gaps in our data, it is likely that we would encounter them

if we would investigate the problems in more detail or with further companies/interviewees. Hence, using the Theory of Distance by Bjarnason et al. [10] to systematically address and reduce the identified gaps could indeed reduce the effect of our found problems.

Hansen and Lyytinen [26] group requirements challenges observed in information systems projects into *individual cognitive challenges*, *interpersonal challenges*, and *complexity-based challenges*. Additionally, they argue that these three levels are hierarchical, i.e. that interpersonal challenges are "largely premised" by individual cognitive challenges and complexity-based challenges are similarly premised by both individual and interpersonal challenges. We observe that only few of our problems can clearly be sorted into only one of these categories. For example, P6, unclear responsibilities and borders, seems to fit mainly into the category of complexity-based challenges, as it is tightly coupled with the formal organisation structure and existing role descriptions. However, a lack of informal communication across organisation borders can contribute a large part to this problem in practice, which would make it more of a interpersonal or even individual challenge. P1, lack of product knowledge, is an extreme example, as it fits all three categories depending on the context. If a single requirements engineer is lacking the necessary domain knowledge, P1 becomes an individual challenge. However, the problem could also be related to a lack of communication skills in the team eliciting requirements or, as in the case of Company A, a lack of communication between product planning and developers. Apart from this lack of fit of our problems to the challenge categories proposed by Hansen and Lyytinen [26], the interrelation between these categories is not always evident in our problems either. For example, it is difficult to see how P7, insufficient resources for understanding and maintaining requirements, is related to interpersonal or even individual challenges. Primarily, it has to do with how the organisation is structured, how projects are organised, and how the budget is allocated to these. We attribute this mismatch to the two aspects of communication and organisation structure included in our problems. While communication aspects seem to fit well with Hansen and Lyytinen's model, the organisation aspect does not fit their hierarchy in the same way.

In contrast to Hansen and Lyytinen's model, the layered behavioural model used in [18] allows us to discuss each problem on multiple layers, without having to sort it into a fixed and exclusive category. For example, P1, lack of product knowledge, has important aspects on individual layer (individual lack of product knowledge), on project layer (lack of discussion and clarification between different teams or development phases) and on company layer (knowledge sharing across projects).

In summary, the problems we found are multi-layered and cannot be sorted into single categories. For example,

P2: Lack of context knowledge could be caused by a lack of defined structure and roles in the organisation, which would make it an organisation problem. The same problem could however also be caused by a lack of informal communication between departments (or companies), which would make it a pure communication problem. Therefore, the problems have to be discussed on several layers which both address the communication and the organisation structure. Using the eight gaps described in [10] seems suitable and should be further investigated when addressing the problems in practice.

5.3 Validation survey

In the following, we discuss the outcomes of the survey performed to evaluate the outcome of our case study, i.e. to test whether the found problems occur and are relevant in other companies in the automotive domain. The demographic data of the survey participants are discussed in Sect. 5.3.1. In Sect. 5.3.2, we present the answers to the topic questions and discuss them in relation to the case study outcomes.

5.3.1 Participant demography

Of the 31 survey participants, 26 work for large companies (at least 250 employees) and 2 for small- or medium-sized companies. Three participants chose to not answer this question.

The survey participants are mainly based in Sweden (23 answers). The remaining 8 participants stated that they work in Germany (3 answers), Austria (2 answers), China (1 answer), or chose not to answer the question (2 participants). Within the Swedish sample, the participants named 5 different employing companies. However, 13 of the 23 Swedish participants left the employer blank.

The majority of participants (17) work for OEMs, with 10 participants working for first-tier suppliers, 2 for second-tier suppliers and 2 for consultancy companies. The participants work in diverse subdomains of the automotive industry, with a large part (16) working within infotainment. The overall distribution is depicted in Fig. 2. Participants who answered "other" mentioned subdomains such as electric propulsion, energy management, or human–machine interfaces.

The participants in our sample perform a wide range of tasks in their daily work. Requirements specification is leading with 15 answers, with Design Definition, Architecture Specification and Testing following with 9 answers each. The entire range is depicted in Fig. 3. Note that multiple answers were allowed in this question. The mentioned tasks for the "other" category are methodology development, interaction design and functional owner.
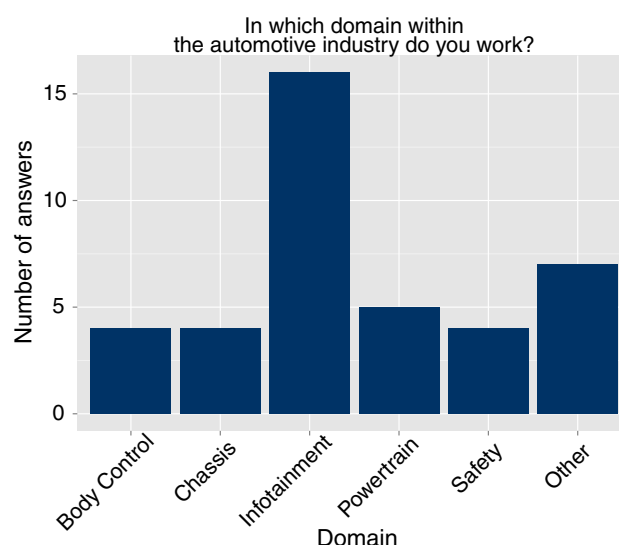

Fig. 2 Subdomains of survey participants

In our sample, we cover both people who write requirements and who receive requirements, with the exchange of requirements both happening within and across organisations. Additionally, one participant stated that he/she is defining how to write good requirements and another participant stated that he/she is working with requirements management. The result for this question is depicted in Fig. 4, with multiple answers possible per participant.

Five participants stated that they were interviewed in the course of the case study presented in this paper. While this has to be kept in mind when analysing the questionnaire data, these participants constitute only a small percentage of the overall participants.

Overall, the large amount of participants from Sweden and the absence of participants from North America limit the generalisability of the survey. We were however able to cover a wide range of different roles and subdomains within the automotive domain. Additionally, our sample contains both participants working for OEMs and for suppliers. Therefore, we capture a wide variety of tasks and roles in the automotive domain, which improves the validity of the study. This is also shown by the different ways participants get into contact with requirements (see Fig. 4). Finally, the survey data are only used to evaluate the findings of our case study with data from a larger sample and not to make general conclusions or elicit entirely new problems/challenges in automotive requirements engineering.

5.3.2 Problem/challenge evaluation

After the demographic questions, we asked the participants to state key problems they experience in requirements
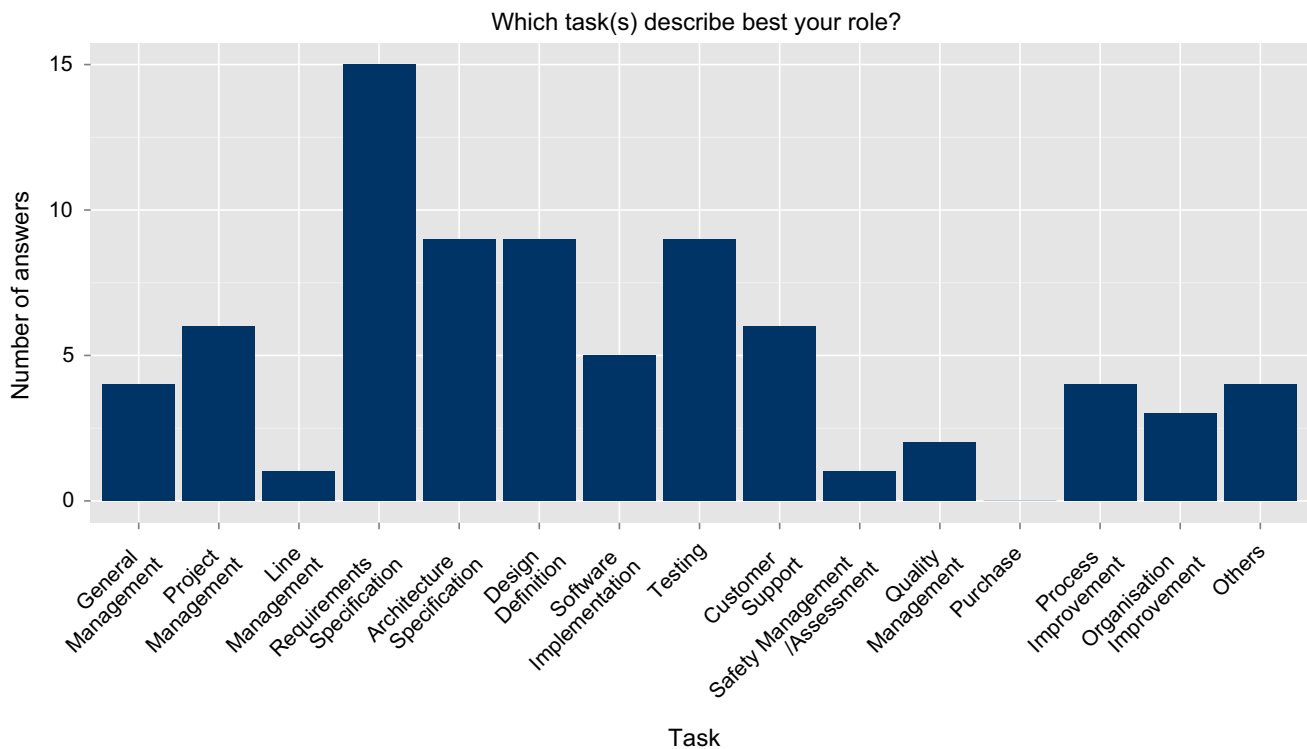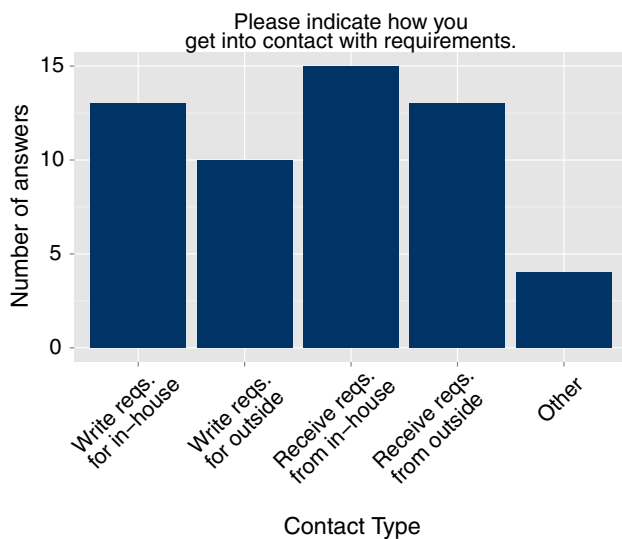
Fig. 3 Tasks of survey participants

Fig. 4 Requirements contact of participants

engineering. We explicitly also encouraged the participants to include the use of requirements later on in the development process, e.g. during testing, as not all participants worked directly in requirements engineering. Apart from the five participants who were interviewed for the case study and were aware of the outcomes of it, the survey participants did not know our results at this point in time. This was done intentionally to not bias them towards our extracted problems/challenges or the area of organisation

or communication challenges in general. The named problems are diverse, without any visible patterns among the different participants. The participants name classical RE challenges, such as formulating requirements on the appropriate level of detail, writing verifiable requirements, or understanding the rationales behind requirements specified by the customer(s). Additionally, problems which are more related to the interface between OEMs, and suppliers are mentioned, e.g. a lack of specification of interfaces between ECUs supplied by different suppliers, or the overhead of handling change requests only through official channels between supplier and OEM.

On the next questionnaire page, we started focusing on the specific topic of communication and organisation structure. We asked participants to rate their agreement to the statements that key problems in automotive RE lie (a) in communication and (b) in the organisation structure. Additionally, the participants were supplied with our definitions of communication and organisation structure, as defined in Sect. 1 (organisation structure as the "decision rule connections" between people in an organisation and communication as the exchange of information between individuals in an organisation). We asked this question in order to see whether the focus area of this paper is indeed of importance in the automotive domain.

The results are depicted in Fig. 5 for communication and in Fig. 6 for organisation structure, grouped by all

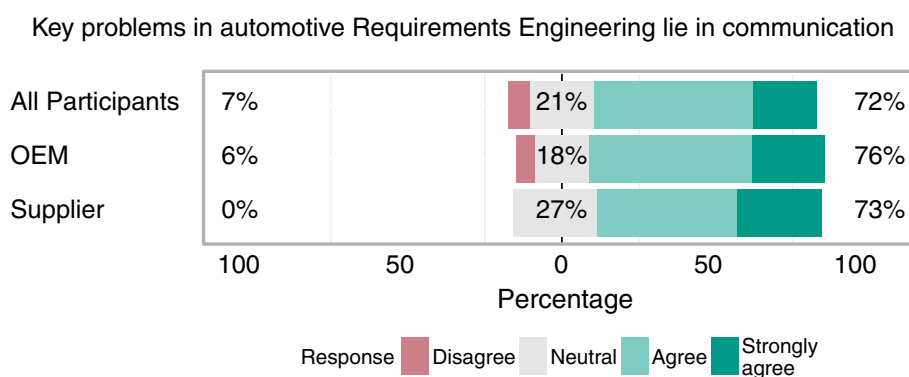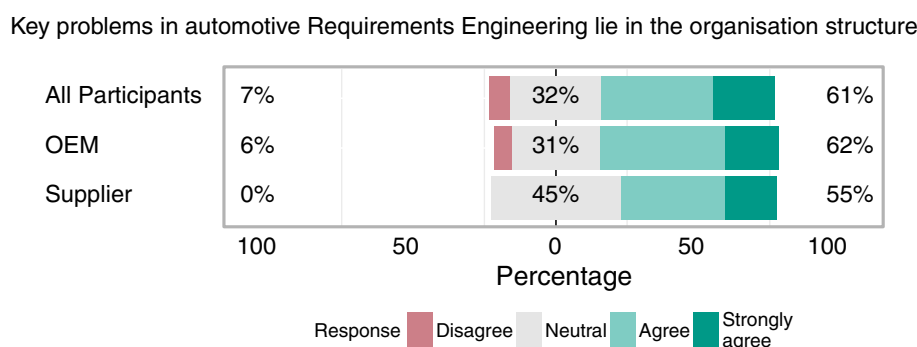**Fig. 5** Communication key problems in automotive requirements engineering

Key problems in automotive Requirements Engineering lie in communication

All Participants	7%	21%	72%
OEM	6%	18%	76%
Supplier	0%	27%	73%

Response ■ Disagree ■ Neutral ■ Agree ■ Strongly agree

Fig. 6 Organisation structure key problems in automotive requirements engineering

Key problems in automotive Requirements Engineering lie in the organisation structure

All Participants	7%	32%	61%
OEM	6%	31%	62%
Supplier	0%	45%	55%

Response ■ Disagree ■ Neutral ■ Agree ■ Strongly agree

participants, participants working for OEMs and participants working for suppliers. We do not visualise the results for the two participants who work for consultancies, due to the small sample size. The green bars to the right depict the percentage of participants who agreed (light green) and strongly agreed (dark green) to the question. The percentage number on the right shows the sum of the two agreement percentages. The light red bar on the left depicts the percentage of participants who disagreed. No participant strongly disagreed with any of the statements. The grey bar in the middle depicts the amount of participants who chose the neutral option. Two participants answered that they do not know the answer for the communication aspect and three participants answered "I don't know" for the organisation structure aspect. The majority agrees with both statements. This indicates that our initial impression, that these two aspects play a crucial role in automotive requirements engineering, is supported by the participants. For this question, there are no significant differences between answers of participants working for OEMs and those working for suppliers (Mann–Whitney U test, $p \approx 0.96$ for the communication aspect, $p \approx 0.87$ for the organisation structure aspect).

Finally, for each of the seven problems/challenges, we asked how often the participants experience each and whether they think that the named problem is a key problem that needs to be addressed.

The answers to the first question (experience frequency) are depicted in Fig. 7. For all seven problems/challenges,

10 or more participants answered that they experience the respective problem/challenge weekly or daily. For P1, lack of product knowledge, it is even the majority (16) of all participants who experience this problem/challenge weekly or daily.

The answers to the second question (problems need to be addressed) are depicted in Fig. 8, sorted by the percentage of participants who agree or strongly agree with the statements. Generally, most participants agree that the named problems need to be addressed. Only for P5, lack of interdisciplinary understanding, less than the majority agrees or strongly agrees to the statement.

In order to find possible solutions, we looked at the free-text answers of participants who disagreed with the statements. For P7, one participant argued that instead of spending more resources, fewer people and better processes are needed. This indicates that instead of adding resources, as the problem title suggests, P7 could be addressed by making the processes slimmer. However, it is unclear what a "better" process is in the participant's opinion.

For P4, two participants reported that fixed processes were established in order to improve the communication between OEM and supplier, e.g. by mirroring the organisation on the supplier side. While this could address the problem, one of the participants stated that it causes a large overhead.

For none of the seven problems, there were any significant differences between the answers of participants
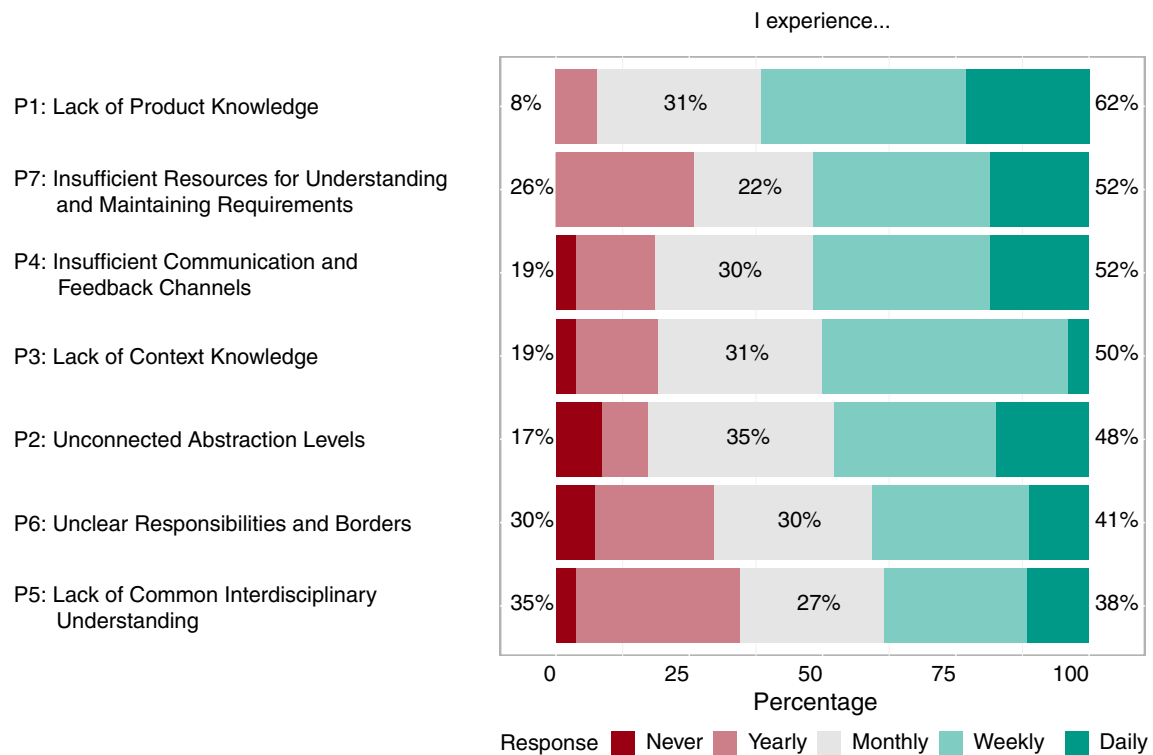
I experience...

Fig. 7 Problems: experience frequency

The named problem is a key problem which needs to be addressed.
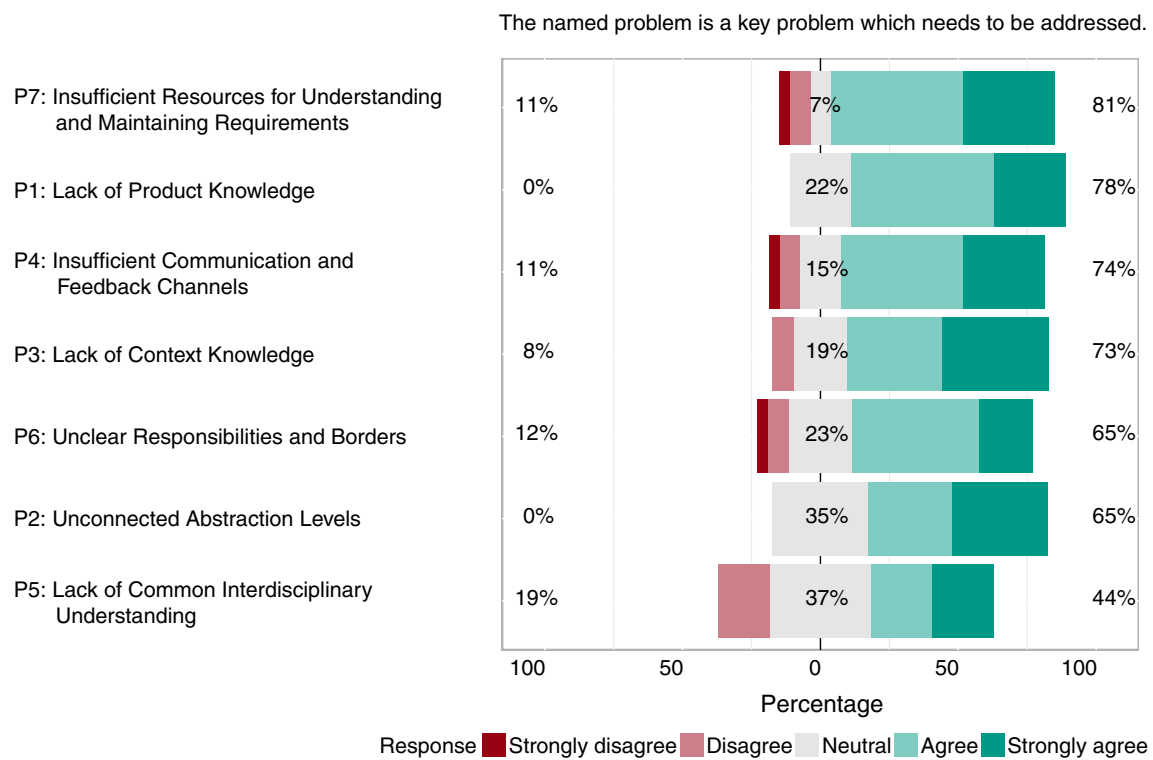

Fig. 8 Problems: necessity to be addressed

working for OEMs and for suppliers. That is, they are recognised regardless of the position in the value chain.

Overall, the survey confirms that all problems or challenges we extracted from our interview data are relevant and occur regularly in practice. Furthermore, there are no significant differences of answers given by participants working for OEMs and those working for suppliers, indicating that the problems are relevant at all positions in the value chain. Only five of the 31 survey participants were interviewed as a part of the case study, and at least seven additional companies are represented in the survey data. The large amount of participants working in infotainment could be explained by a potential selection bias. However, it could also indicate that infotainment is facing more challenges/problems in RE than other areas, thus triggering more interest to participate in a survey studying these problems.

6 Validity threats

We applied several measures in order to reduce the threats to validate in this paper as much as possible, for instance *Triangulation, Member Checking and Prolonged Involvement* [17]. In the following, we describe the threats and our countermeasures following the categorisation by Runeson et al. into *Construct Validity, Internal Validity, External Validity and Reliability* [40].

6.1 Construct validity

Construct validity describes whether the constructs used in the study, e.g. terms and definitions, are interpreted in the same way by the researchers, the interviewees and other people involved in the study. Additionally, it describes whether interviewees were biased by the presence of one or multiple researchers or whether the questions were asked in a way that suggested a certain answer.

The interview guide was designed by the first author and discussed with the second author, and one additional researcher not involved in the planning. We tried to reduce ambiguities with respect to terms and definitions as much as possible, e.g. by adding definitions to the interview guide. Furthermore, the interview questions were reviewed and refined to ensure clarity and, additionally, in order to eliminate suggestive questions. Similarly, the validation questionnaire was reviewed by four researchers and four practitioners, and definitions of the two focus areas of communication and organisation structure were given to survey participants.

Both the first and the second authors have a *prolonged involvement* of at least one year with both case companies. Therefore, there were already non-disclosure agreements in place, forming a basis for mutual trust between the involved parties.

While the interviewees were selected by one contact person at each company, they participated on a voluntary basis. They were granted anonymity and the option to review their transcribed interview before it was shared or discussed with the contact persons (*Member checking*).

6.2 Internal validity

When causal relations are analysed, internal validity reflects whether all of them have been examined or whether there are further, unknown, factors which might affect the outcome. As with every qualitative study, especially when the context is a large organisation, it is impossible to study all contextual factors. However, we used a number of measures in order to ensure internal validity.

We used data *Triangulation* throughout the data collection and data analysis, using only statements expressed by multiple interviewees. At Company A, we interviewed multiple roles and multiple people in each role in order to get a cross-cutting picture of the problems. While we did not have multiple interviewees for each role in Company B, we still used data triangulation in order to extract common problems.

The recently changed process in the application software engineering area of Company B is a threat to validity as well, as observed challenges/problems might be related to this process change. By using triangulation across the different areas, we address this threat. However, we might also have missed challenges/problems that could appear once the process has matured. This should be investigated in a follow-up study.

In order to ensure continuity in the data collection process, all interviews were conducted by the first author using the same interview guide. However, due to the semi-structured nature of the interviews, different follow-up questions were asked depending on the context.

Selection threats cannot be ruled out as one contact person at each company contacted potential interviewees. To address this, participation was voluntary and invitations were sent to a larger sample of potential interviewees. Similarly, the survey suffers from potential selection threats as it was sent out to our own contacts in the automotive domain. To limit this threat, we encouraged our contacts to further distribute the survey to their contacts and additionally contacted researchers with established networks outside our automotive contacts.

6.3 External validity

External validity concerns to which degree the results of the study can be generalised to a broader context. The

external validity of case studies is generally low and by studying only two cases, and we could hardly claim that the presented problems are general to the automotive domain or even a wider context. As a way to increase this low external validity, we conducted the validation survey. While the survey participants mostly work in Europe, we reached a much larger variety in terms of roles and companies than in the case study alone. The fact that the majority of the participants experiences the stated problems regularly and recognises them as relevant problems indicates that our findings are to some extent generalisable. Furthermore, we believe that the results should be generalisable to areas that resemble the automotive domain in terms of constraints and scope. That is, heavily regulated systems engineering domains, such as aerospace or railway, or domains that have a strong division of value creation between OEMs and a chain of suppliers should face similar challenges in their software engineering efforts.

6.4 Reliability

Reliability describes to which extent the outcomes of a study are dependent on the researchers who conducted it, i.e. if someone else would arrive at the same conclusions when replicating the study.

With respect to the case study design, we aimed at reducing reliability threats as much as possible by reviewing the design and the interview instrument. Similarly, we discussed the code set used for analysis of the interview data among the three first authors.

The interviews were transcribed word by word by the first author in order to avoid subjective judgment. When it comes to the abstraction and categorisation of the coded statements, a certain degree of subjectivity is however unavoidable. We tried to address this by discussing the resulting analysis with our contact persons at both companies and by comparing the resulting problems to existing problems in the literature. Furthermore, we conducted the validation survey to evaluate whether our categorisation leads to problems that can be recognised by practitioners. However, we can not completely rule out the subjective element in our analysis.

7 Conclusions and future work

In this paper, we presented the results of a case study at two companies, one original equipment manufacturer (OEM) and one supplier, within the automotive domain. We studied their current challenges in requirements engineering (RE) with respect to organisation structure and communication by collecting data through 14 interviews with people from the areas of *embedded software engineering*, *systems engineering* and *application software engineering*. We extracted seven key challenges, thus answering our first research question, *What are current problems or challenges in automotive RE with respect to organisation structure and communication?* We tested them using an online questionnaire with 31 participants, mainly from European OEMs and suppliers.

Related problems/challenges have been reported in previous publications, e.g. [8, 31] with regard to large-scale and market-driven RE. However, we are the first to report specific RE problems/challenges with respect to communication and organisation structure in the automotive domain, where specific characteristics such as a large supplier network, or varying laws in different countries need to be considered. Additionally, the conclusions of our interviewees for how the problems should be addressed differ from the literature in other domains.

In order to answer the second research question, *Which approaches are proposed by practitioners in order to address these problems in the future?*, we extracted potential solutions to the problems from our data and the studied related work. Within these solutions, we see two seemingly contradicting trends. These are (a) the trend to shift towards *agile processes*, including aspects like *customer involvement and handshakes* and (b) the trend towards stricter processes using *formal reviews, models and increased tracing*. To some extent, these different trends could arise due to the different areas which we studied. That is, introducing agile processes could be easier within areas that are purely or mainly focusing on software engineering than in areas such as systems engineering, where stricter processes might be advantageous or even imposed by standards. However, we see that even in those latter areas, several practitioners are advocating a change towards agile processes. An alternative explanation could be that the increase in agile processes necessitates stricter processes on other organisation levels in order to cope with the increased levels of informal communication and interaction. That is, the two trends could in fact complement each other. This explanation bears similarities to the discussion in the area of codified and tacit knowledge in knowledge management. Here, it is typically argued that both codified and tacit knowledge are important in organisations [36] and that cultural and social aspects are of high relevance [12, 14, 25, 34]. Generally, the area of knowledge management should be considered for a more systematic investigation of solution approaches to our found problems. For this purpose, several systematic literature reviews exist both within software engineering, e.g. [11], and outside of software engineering, e.g. [36].

For academia, our findings open up several opportunities for future research. First, our case study could be extended by further cases in the automotive domain in order to get a

deeper understanding of the found issues. In particular, the data gathered at Company B should be followed up, as parts of the company had introduced an agile process shortly before the interviews. This change seems to have solved some problems with requirements or requirements-related knowledge, e.g. misalignment of requirements and tests. However, interviewees also voiced concern that the lack of documentation in agile development might raise new challenges in the future. Secondly, future studies could try to extend the list of possible solutions to each problem and investigate in which concrete situations they are indeed solutions. Based on the found problems and the proposed solutions, we see the following needs which should guide academic research on automotive RE in the future. First, we see the *need for a process that allows for sufficient levels of uncertainty during early RE*. While ideally requirements engineers could have enough product knowledge of a company's product range, emerging technologies make uncertainty in automotive RE unavoidable. Uncertainty is not a new concept in RE and in project management, but it is becoming more and more important due to the increasing speed of technological change. Automotive RE needs to support this and manage uncertainty in a way that it becomes shared knowledge throughout the entire automotive value chain. Secondly, we see the *need for an organisation structure that effectively supports interdisciplinary RE, taking into account the central role of software*. As software starts to play an ever more important role and grows in terms of scale and complexity, structuring an organisation based on manufacturing has to be questioned. Furthermore, as requirements can easily span multiple disciplines in automotive RE, it is not enough to specify them separately for each discipline. It is not only essential that multiple disciplines are involved, but also that the organisation structure supports this and does not artificially separate the disciplines into their own units or departments. This also includes an active support of formal and informal communication between the involved disciplines. Thirdly, we see a *need for concepts and an organisation structure that allow for and support managing 'requirements debt'*. In automotive systems engineering, projects build on parts which are developed in previous projects, e.g. an existing requirements based or the electrical vehicle architecture. Additionally, with paradigms such as continuous delivery playing an increasingly important role, classical projects could be replaced by a continuous development effort. As long as there is no budget for maintaining and improving existing requirements, technical debt [33] for requirements, i.e. "requirements debt", will be accumulated and never paid off. While structured ways to write requirements could help avoiding this kind of debt, e.g. by using requirements patterns or models, requirements are in many cases already existing and often reused. Therefore, it is important to investigate how to manage and to improve legacy requirements that are not written in such a way. Future research should therefore investigate how this debt can be managed and paid off, so that requirements quality does not deteriorate slowly. All three needs are not focused on software development alone. Therefore, they have to be addressed in an interdisciplinary context, considering existing research from areas such as project management or organisation theory.

For the automotive industry, the descriptions of the found issues can serve as an aid to understand under which circumstances they can occur. Additionally, the outlined solutions could support organisations to evaluate multiple viable options in process and organisation improvement attempts. The formulated needs can guide their improvement efforts in specific directions.

References

1. Almefelt L, Berglund F, Nilsson P, Malmqvist J (2006) Requirements management in practice: findings from an empirical study in the automotive industry. Res Eng Design 17(3):113–134
2. Altman S, Valenzi E, Hodgetts RM (eds) (1985) Organizational behavior: theory and practice, 1st edn. Academic Press, Cambridge
3. Atlassian: JIRA-Issue and Project Tracking Software (2015) https://www.atlassian.com/software/jira
4. Averbakh A, Knauss E, Kiesling S, Schneider K (2014) Dedicated support for experience sharing in distributed software projects. In: Proceedings of 26th international conference on software engineering and knowledge engineering (SEKE'14) (2014)
5. Baligh HH (2006) Organization structures: theory and design, analysis and prescription, 1st edn. Springer, Berlin
6. Basili VR, Caldiera G, Rombach HD (1994) Experience factory. Encycl Softw Eng 1:469–476
7. Bjarnason E (2013) Distances between requirements engineering and later software development activities: a systematic map. In: Doerr J, Opdahl AL (eds) Requirements engineering: foundation for software quality, lecture notes in computer science, vol 7830, pp 292–307

8. Bjarnason E, Wnuk K, Regnell B (2011) Requirements are slipping through the gaps—a case study on causes and effects of communication gaps in large-scale software development. In: Proceedings of 19th IEEE international requirements engineering conference (RE '11), pp 37–46

9. Bjarnason E, Runeson P, Borg M, Unterkalmsteiner M, Engström E, Regnell B, Sabaliauskaite G, Loconsole A, Gorschek T, Feldt R (2014) Challenges and practices in aligning requirements with verification and validation: a case study of six companies. Empir Softw Eng 19(6):1809–1855

10. Bjarnason E, Smolander K, Engström E, Runeson P (2014) Alignment practices affect distances in software development: a theory and a model. In: Proceedings of 3rd SEMAT workshop on general theories of software engineering (GTSE '14), pp 21–31

11. Bjørnson FO, Dingsøyr T (2008) Knowledge management in software engineering: a systematic review of studied concepts, findings and research methods used. Inf Softw Technol 50(11):1055–1068

12. Boer NI, Berends H, van Baalen P (2011) Relational models for knowledge sharing behavior. Eur Manag J 29(2):85–97

13. Braun P, Broy M, Houdek F, Kirchmayr M, Müller M, Penzenstadler B, Pohl K, Weyer T (2014) Guiding requirements engineering for software-intensive embedded systems in the automotive industry. Comput Sci Res Dev 29(1):21–43

14. Bresnen M, Edelman L, Newell S, Scarbrough H, Swan J (2003) Social practices and the management of knowledge in project environments. Int J Project Manag 21(3):157–166

15. Broy M (2006) Challenges in automotive software engineering. In: Proceedings of 28th international conference on software engineering (ICSE '06), pp 33–42

16. Chakraborty S, Ramesh S (2015) Guest editorial special section on automotive embedded systems and software. IEEE Trans Comput Aided Des Integr Circuits Syst 34(11):1701–1703

17. Creswell JW (2003) Research design: qualitative, quantitative, and mixed methods approaches, chap. 10., 2nd edn. SAGE Publications, Thousand Oaks

18. Curtis B, Krasner H, Iscoe N (1988) A field study of the software design process for large systems. Commun ACM 31(11):1268–1287

19. Dingsøyr T, Moe NB (2013) Research challenges in large-scale agile software development. SIGSOFT Softw Eng Notes 38(5):38–39

20. Ebert C, Jones C (2009) Embedded software: facts, figures, and future. Computer 42(4):42–52

21. Fitzgerald B, Stol KJ, O'Sullivan R, O'Brien D (2013) Scaling agile methods to regulated environments: an industry case study. In: Proceedings of the 2013 international conference on software engineering (ICSE '13), pp 863–872

22. France R, Rumpe B (2007) Model-driven development of complex software: a research roadmap. In: Future of software engineering (FOSE '07), pp 37–54

23. Fricker S, Gorschek T, Byman C, Schmidle A (2010) Handshaking with implementation proposals: negotiating requirements understanding. IEEE Softw 27(2):72–80

24. Gottesdiener E (2003) Requirements by collaboration: getting it right the first time. IEEE Softw 20(2):52–55

25. Hanisch B, Lindner F, Mueller A, Wald A (2009) Knowledge management in project environments. J Knowl Manag 13(4):148–160

26. Hansen S, Lyytinen K (2010) Challenges in contemporary requirements practice. In: 43rd Hawaii international conference on system sciences (HICSS), pp 1–11

27. Heumesser N, Houdek F (2004) Experiences in managing an automotive requirements engineering process. In: Proceedings of 12th IEEE international requirements engineering conference (RE '04), pp 322–327

28. Houdek F, Pohl K (2000) Analyzing requirements engineering processes: a case study. In: Proceedings of 11th international workshop on database and expert systems applications, pp 983–987

29. INCOSE (2015) INCOSE-What is systems engineering?. http://www.incose.org/practice/whatissystemseng.aspx

30. Kahkonen T (2004) Agile methods for large organizations-building communities of practice. Agile development conference 2004, pp 2–10

31. Karlsson L, Dahlstedt ÅG, Regnell B, och Dag Natt J, Persson A (2007) Requirements engineering challenges in market-driven software development—an interview study with practitioners. Inf Softw Technol 49(6):588–604

32. Kylmäkoski R (2003) Efficient authoring of software documentation using rapid7. In: 25th international conference on software engineering, pp 255–261

33. Li Z, Avgeriou P, Liang P (2015) A systematic mapping study on technical debt and its management. J Syst Softw 101:193–220

34. Lindner F, Wald A (2011) Success factors of knowledge management in temporary organizations. Int J Project Manag 29(7):877–888

35. March JG, Simon HA (1993) Organizations, 2nd edn. Wiley, London

36. Maryam Alavi DEL (2001) Review: knowledge management and knowledge management systems conceptual foundations and research issues. MIS Q 25(1):107–136

37. Maurer M, Winner H (2013) Automotive systems engineering. Springer, Berlin

38. Pernstål J, Magazinius A, Gorschek T (2012) A study investigating challenges in the interface between product development and manufacturing in the development of software-intensive automotive systems. Int J Softw Eng Knowl Eng 22(7):965–1004

39. Pretschner A, Broy M, Kruger IH, Stauner T (2007) Software engineering for automotive systems: a roadmap. In: Future of software engineering (FOSE '07), pp 55–71

40. Runeson P, Höst M, Rainer A, Regnell B (2012) Case study research in software engineering, 1st edn. Wiley, London

41. Scaled Agile Inc.: Scaled agile framework (2015) http://www.scaledagileframework.com/

42. Sikora E, Tenbergen B, Pohl K (2012) Industry needs and research directions in requirements engineering for embedded systems. Requir Eng 17(1):57–78

43. Sinha V, Sengupta B, Chandra S (2006) Enabling collaboration in distributed requirements management. IEEE Softw 23(5):52–61

44. Vandevelde A, Van Dierdonck R (2003) Managing the design-manufacturing interface. Int J Oper Prod Manag 23(11–12):1326–1348

45. Weber M, Weisbrod J (2002) Requirements engineering in automotive development-experiences and challenges. In: Proceedings of IEEE joint international conference on requirements engineering (RE '02), pp 331–340

46. Weinstein M (2015) TAMS analyzer (2015). http://tamsys.sourceforge.net/