Requirements and software engineering for automotive perception systems: an interview study

Downloaded from: https://research.chalmers.se, 2025-03-21 00:16 UTC

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

Habibullah, K., Heyn, H., Gay, G. et al (2024). Requirements and software engineering for automotive perception systems: an interview study. Requirements Engineering, 29(1): 25-48.
http://dx.doi.org/10.1007/s00766-023-00410-1

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

research.chalmers.se offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all kind of research output: articles, dissertations, conference papers, reports etc. since 2004. research.chalmers.se is administrated and maintained by Chalmers Library

(article starts on next page)

Requirements and Software Engineering for Automotive Perception Systems: an Interview Study

No Author Given

No Institute Given

Abstract. Background: Driving automation systems, including autonomous driving and advanced driver assistance, are an important safety-critical domain. Such systems often incorporate perceptions systems that use machine learning to analyze the vehicle environment.
Aims: We explore new or differing topics and challenges experienced by practitioners in this domain, which relate to requirements engineering (RE), quality, and systems and software engineering.
Method: We have conducted a semi-structured interview study with 19 participants across five companies and performed thematic analysis of the transcriptions.
Results: Practitioners have difficulty specifying upfront requirements, and often rely on scenarios and operational design domains (ODDs) as RE artifacts. RE challenges relate to ODD detection and ODD exit detection, realistic scenarios, edge case specification, breaking down requirements, traceability, creating specifications for data and annotations, and quantifying quality requirements. Practitioners consider performance, reliability, robustness, user comfort, and—most importantly—safety as important quality attributes. Quality is assessed using statistical analysis of key metrics, and quality assurance is complicated by the addition of ML, simulation realism, and evolving standards. Systems are developed using a mix of methods, but these methods may not be sufficient for the needs of ML. Data quality methods must be a part of development methods. ML also requires a data-intensive verification and validation process, introducing data, analysis, and simulation challenges.
Conclusions: Our findings contribute to understanding RE, safety engineering, and development methodologies for perception systems. This understanding and the collected challenges can drive future research for driving automation and other ML systems.

Keywords: requirements engineering · software quality · software development methodologies · driving automation systems · autonomous driving

1 Introduction

Driving automation systems, including both autonomous driving (AD) and advanced driver assistance systems (ADAS), are software systems designed to augment or automate aspects of vehicle control [59]. Driving automation systems have long been a domain of interest. However, the increased capabilities and usability of machine learning (ML) have subsequently improved the capabilities of—and interest in—such systems. Research advances have improved comfort and safety, and reduced fuel and energy consumption, emissions, and travel time [59].

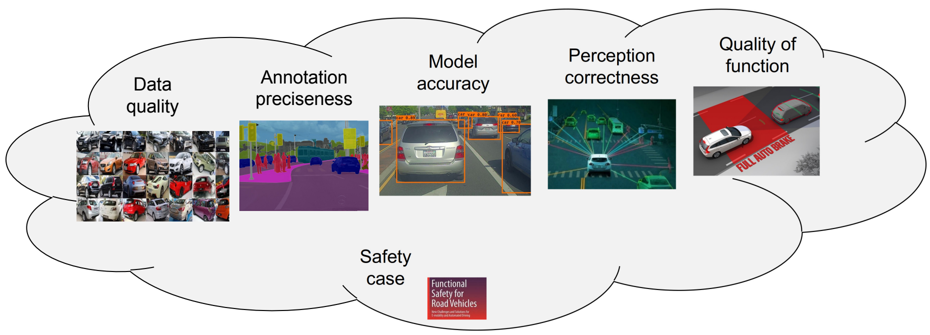

Fig. 1: Conceptual model of quality transitions from data collection to the quality of the automotive function.

Driving automation system functionality depends on the correctness and the integrity of perception systems that blend ML-based models and traditional signal processing[1]. The usage of ML for perception relies on a large quantity and high quality of data. Data quality, context, and attributes—as well as annotation quality—have a significant impact on the resulting system quality. However, it is difficult to make direct connections between data, annotation, ML model quality, and the resulting functional quality of a perception system (e.g., between the boxes in Figure 1). The inherent uncertainty of ML—coupled with high requirements on data quality and coverage—creates substantial requirements, systems, and software engineering challenges in perception system development [14].

Requirements engineering (RE) is an important foundational element of quality assurance and safety engineering. RE plays a critical role in perception system development by enabling explicit capture of safety and quality requirements, supporting communication, recording functional expectations, and ensuring that standards are followed. Additionally, systems and software engineering play a critical role in the successful development and deployment of perception systems by enhancing real-time decision-making [19], supporting adaptability and continuous learning, facilitating complex system integration [39], maximizing performance, ensuring dependability and safety [49], encouraging cross-disciplinary collaboration [70], and advancing ethical and responsible development methods [55].

[1] In this paper, we focus specifically on ML-based perception systems for driving automation systems, but often use the term *perception systems* as shorthand.

Recent research has explored RE challenges for ML systems, e.g., [12, 80], as well as systems and software engineering challenges [10, 11, 56]. However, such challenges have not been thoroughly explored in the context of perception systems for driving automation systems. Addressing this gap is necessary to advance practices in both this domain and in the broader context of RE for ML systems.

To explore important engineering topics and challenges for perception systems, we have conducted an interview study with 19 domain experts from five companies working in various driving automation systems roles. We analyzed interview data using thematic coding to produce eight major themes: perception, requirements engineering, systems and software engineering, AI and ML models, annotation, data, ecosystem and business, and quality.

This paper is an extension of previous work [30]. The initial article focused specifically on the RE themes from the thematic analysis, encapsulating RE topics and challenges discussed by the participants. In this paper, we extend the analysis and discussion of the RE theme to include findings from two additional themes—systems and software engineering and quality. For both themes, we also explore topics and challenges for driving automation systems development that were raised in the interviews[2]. These two themes, in particular, add relevant insights for practitioners and, additionally, enrich our understanding of RE practices and challenges in this domain (e.g., requirements and quality are tightly interconnected). In addition, we include a more extensive related work section, discussion an outline of future directions in research and practice for driving automation systems and other ML systems.

Related to RE, our findings indicate that practitioners have difficulty breaking down specifications for the ML components. In practice, individuals report that they use scenarios, operational design domains (ODDs), and simulations as part of RE. Practitioners experience RE challenges related to uncertainty, ODD detection, realistic scenarios, edge case specification, traceability, creating specifications for data and annotations, and quantifying quality requirements.

In terms of quality, practitioners consider performance, reliability, robustness, safety, and user comfort as important quality attributes. In the context of driving automation systems, safety is particularly critical. Practitioners establish safety goals, often in negotiation with component suppliers. To ensure safety, practitioners must comply with evolving safety and AI standards—which are challenging and costly to meet. They must also manage trade-offs between safety and other qualities. Safety cases are a critical element of ensuring that the safety goals are met. Quality assurance is performed by tracking critical Key Performance Indicators (KPIs) during the execution of catalogs of scenarios. Quality assurance is complicated by the non-determinism and data requirements of ML and the realism of simulation.

From a systems and software engineering perspective—though practitioners work with traditional and agile methods—ML complicates the overall development process. Current agile methods are insufficient for the needs of large-scale

[2] Another recent article has also used the same interview data, but focused on the annotation, data, and ecosystems and business themes [38].

ML because practitioners lack appropriate data quality methods as part of their overall development methodology [66]. Furthermore, the addition of ML leads to a data-intensive verification and validation (V&V) process with challenges related to data quality, statistical analysis, and simulation.

By exploring the views and challenges of practitioners on RE, quality, and software and systems engineering for ML-enabled perception systems, we provide valuable insights for practitioners working in this safety-critical domain. Additionally, our findings contribute to improving RE, and systems and software engineering knowledge more broadly, in other domains reliant on ML.

2 Related Work

In this section, we review related work in requirements engineering for machine learning systems and for automotive and driving automation systems. We give an overview of work on quality for machine learning, and software development methods for machine learning, as these are key themes of focus in our interview study.

2.1 RE for ML

Recent research has focused on how RE could or must change in the face of rising use of ML. Systematic mapping studies on RE for ML identified new contributions in this area, including approaches, checklists, guidelines, quality models, classifications and evaluations of quality models, taxonomies, and quality requirements [7, 28, 79]. Pei et al. reviewed literature on RE for ML, went through a collaborative requirements analysis process, and provided an overview of RE processes for ML applications in terms of cross-domain collaboration [63]. They provided an example case of an industrial data-driven intelligence application, discussed in relation to the provided requirements analysis process. Ahmad et al. performed a systematic mapping study to find articles on current RE for AI approaches, and identified available frameworks, methodologies, tools, and techniques used to model requirements, and finds existing challenges and limitations [4]. They identified 43 primary studies and found several challenges and limitations of existing RE for AI practices, for example that current RE processes are not adequately adaptable for building AI systems. The authors emphasised that new techniques and tools are needed to support RE for AI.

Further papers have identified RE-related challenges for ML and AI. Ahmad et al. investigated current approaches for writing requirements for AI/ML systems, identified tools and techniques to model requirements for AI/ML, and pointed out existing challenges and limitations in this area [5]. Belani et al. identified and discussed RE challenges for ML and AI-based systems, and reported that identifying NFRs throughout the software lifecycle is one of the main challenges [12]. Heyn et al. used three use cases of distributed deep learning to describe AI system engineering challenges related to RE [36], including context, defining data quality attributes, human factors, testing, monitoring and reporting. In further study, Heyn et al. identified several challenges related to training data specification (e.g., unclear design domain, missing guidelines for data selection, and unsuitable safety standards) and run-time monitoring for ML models, with challenges relating to RE (e.g., lack of explainability for ML decisions, missing conditions for run-time checks, and overhead for monitoring solution) [35].

Other studies move towards proposed solutions. Farrell et al. identified key characteristics of ML-based software requirements such as confidence, accuracy, average value, robustness, data-driven learning, and quality aspects, providing a foundation for developing a taxonomy of requirements for such software [22]. Islam et al. presented a requirements process (RESAM) that integrates knowledge from different sources, such as discussion forums, domain experts, and formal product documentation, to discover and specify requirements and design definitions that contribute to the construction of effective deep learning anomaly detectors. They evaluated their process in a case study and demonstrate that it guides the construction of effective anomaly detection models that support explainability [6].

2.2 RE for Automotive and Driving Automation Systems

Significant research has been performed on RE for vehicles. Liebel et al. identified challenges in automotive RE with respect to communication and organization structure [53]. Pernstal et al. stated that RE is one of the areas most in need of improvement at automotive original equipment manufacturers (OEMs), and also identified the ability to communicate via requirements as important [64]. Allmann et al. also noted requirements communication as a major challenge for OEMs and their suppliers [9]. Mahally et al. identified that requirements are the main enablers and barriers of moving towards Agile for automotive OEMs [57].

Research has also looked specifically at RE for AD, e.g., providing an overview of AD RE techniques [74]. Riberio et al. identified AD RE challenges addressed by the literature, and identified the languages and description styles used to describe AD requirements, with special attention given to NFRs [68]. Heyn et al. investigated challenges with context and ODD definition in ML-enabled perception systems [37], including a lack of standardisation for context definitions, ambiguities in deriving ODDs, missing documentation, and lack of involvement of function developers while defining the context. Ågren et al. identified six aspects of RE that impact automotive development speed, moving toward AD [3].

In further driving automation systems work relating to RE, Zhang et al. conducted a systematic mapping study in the context of driving automation systems, and introduced a taxonomy for critical scenario identification methods including encompassing the problem definition of the solution, and the assessment of the established scenarios [86]. They also discussed challenges considering the perspectives of coverage, practicability, and scenario space explosion. Luo et al. proposed a hierarchical safety assessment approach to quantitatively analyze the quality trade-offs, violation severity of safety requirements, and distinguish safer autonomous driving systems configurations based on the requirements violations comparison in a hierarchical way, following requirements importance [54]. Zhang et al. presents a data-driven engineering process that includes hierarchical requirements engineering to link the operational design domain with the requirements and semi-automated generation of data sets for leveraging future application of ML in automated driving in industry [85].

2.3 Quality Assurance for ML

Although quality for ML can be interpreted in a narrow sense, i.e., basic model performance, work exists which has focused on ML quality in a broader sense. Felderer et al. discussed terminology for quality assurance for AI systems, defining concepts and characterizing AI systems into artifact type, process, and quality characteristics [23]. They also discussed challenges in quality assurance such as: lack of specifications and defined requirements; the need for validation data and test input generation; difficulty defining expected outcomes as test oracles, and baselines for AI-based systems. Furthermore, different challenges and opportunities related to quality requirements for machine learning systems are reported and discussed in [12, 29, 31, 41].

From the perspective of traditional quality-assurance, the Japanese industry has collectively proposed a set of recommendations for the quality assurance of AI systems (e.g., in the Consortium of Quality Assurance for AI-based Products and Services), and the second iteration of these standards, which includes a list of quality evaluation criteria, a list of cutting-edge methods, and explanations of each of the five representative domains that are proposed in [25]. In a research Project for the Establishment of Generally Accepted quality criteria, tools and methods as well as Scenarios and Situations for the release of highly automated driving functions (PEGASUS), 17 partners from research and industry worked together with the aim to develop a complete toolchain to include criteria and measures for the evaluation of functions and for driving automation systems quality levels, with test catalogues, central methods for driving automation systems development, and processes for establishing safety, and to release highly automated driving functions [82].

From the perspective of technical standards, the automotive industry is aware of adjustments in machine learning-based technology demands in terms of technical expertise, development paradigms, and cultural approaches. However, there is still a significant gap between the availability of technical standards and certification capacity. Currently, the automotive industry is governed by several standards. However, existing work has argued that these standards are not suitable for machine learning-based driving automation systems [21, 46]. Although further certifications of autonomous systems (e.g., SOTIF) are being developed and are advancing, these efforts only cover some of the existing challenges [24].

Other work has focused specifically on data quality in relation to machine learning. Jain et al. discussed the importance of data quality, and stated that the effort required to iteratively debug a machine learning pipeline in order to enhance model performance can be reduced by evaluating the quality of the data using intelligently defined metrics transformation operations [44]. The authors also survey the important data quality related approaches discussed in literature—highlighting their strengths and similarities and discussing their applicability to real-world challenges.

Further work has looked at AI in terms of risks. Poth et al. presented a systematic methodical approach (the evAIa method: evaluate AI approaches) that evaluate risks of the machine learning model using a questionnaire specifically for AI products and services [65].

2.4 Software and Systems Methods for Machine Learning

Current systems and software development methods often do not account well for machine learning-enabled systems. Giray points out a lack of techniques to support machine learning system development as part of a systematic literature review, reporting that the non-deterministic nature of machine learning systems complicates SE aspects of engineering machine learning systems that includes a lack of mature tools and techniques to support machine learning systems development and verification [26].

Given the rise of machine learning-enabled software, researchers have explored or introduced a number of methods and challenges for machine learning and AI system development. Hesenius et al. provided a structured engineering process framework named EDDA (engineering data driven applications) that bridges existing gaps, supports data-driven application development and ensures the required quality levels for critical components of machine learning systems [34]. Amershi et al. conducted a case study where the authors described how various Microsoft software teams developed software applications with customer-focused AI features—integrating existing Agile software engineering process with AI-specific workflows [10].

Further research looked into the challenges of engineering driving automation systems. Key collaboration challenges were identified in developing and deploying machine learning systems through interviews with 45 participants from 28 organizations [60]. The authors reported on common collaboration points and challenges from the perspective of requirements, data, integration, and team patterns and found the majority of the challenges center around communication, documentation, engineering, and process. In addition, safety criticality extends the decision-making, development, and related environmental perception [2]. This complexity does not harmonize with conventional safety engineering, hence, the application of concepts for intelligence is required to resolve the complexity.

3 Methodology

Our study is guided by the following research questions:

- **RQ1:** What requirements engineering topics of interest and challenges are encountered by the developers of perception systems for driving automation systems?

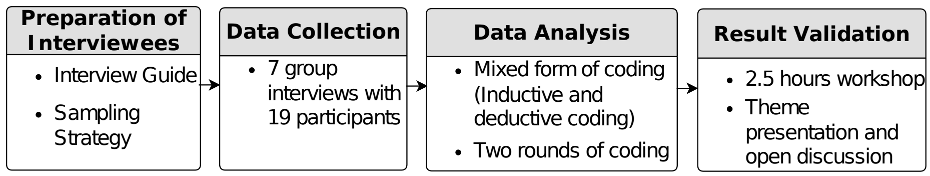

Fig. 2: Overview of the interview study.

– **RQ2:** What quality topics and challenges are encountered by the developers of perception systems for driving automation systems?
– **RQ3:** What software and systems engineering topics and challenges are encountered by the developers of perception systems for driving automation systems?

  We refer to a topic of interest as something that practitioners currently practice or are curious about or would like to learn more about. A challenge on the other hand refers to an obstacle or difficulty that practitioners encounter and must overcome in order to successfully develop perception systems for driving automation systems.

To address these questions, we conducted seven group interviews with 19 expert participants from five companies that are currently working with ML-based perception systems for driving automation systems. Figure 2 gives an overview of the interview study.

3.1 Data Collection

We used semi-structured group interviews with a set of predetermined open-ended questions. The use of semi-structured interviews ensured that all participants addressed the same questions, while still allowing the freedom to follow-up with additional questions on particular topics[3].

The interviews were conducted between December 2021 and April 2022 via Microsoft Teams, and each lasted between 1.5–2 hours. We recorded all interview sessions with the permission of all participants, then transcribed and anonymized the recordings for analysis. At least three researchers were present in each interview, with two particular researchers in all interviews to maintain consistency.

A summary of the interviews and the participants who took part is shown in Table 1. We chose participants who posses experience with ML, perception systems for driving automation systems, software and systems engineering, RE, or data science, or who were working in the driving automation systems industry. The sampling method was a mix of purposive, convenience, and snowball sampling. We sent open calls to the Swedish automotive industry, and our known

[3] The interview guide can be found at: `https://doi.org/10.7910/DVN/HCMVL1`.

Table 1: Overview of the conducted interviews, with the focus of the work conducted by the participants and the roles of the participants (same interviews reported by Heyn et al. [38]).

Interview	Field of Work	Participants
A	Object detection	Product owner
B	Autonomous Driving	Product owner, test engineer, ML engineer, software developer
C	Vision systems	System architect, product owner, requirement engineer, deep learning engineer
D	AD and ADAS	System engineer, manager AD
E	Testing and validation AD	System architect, two product owners, compliance officer, data scientist
F	Data annotations	AI engineer, data scientist
G	Autonomous Driving	System safety engineer

contacts, then we asked the interviewees for further contacts. Our participants work with different aspects of driving automation systems.

We started by asking for demographic information about the participants. We then showed them Figure 1, asking for their feedback and using the figure to ground further discussions about how functional requirements relate to requirements on data and data annotation. We asked further questions about their requirements documentation, safety issues, and quality. Although we carefully chose interview participants, the opinions of the individual interviewees do not necessarily reflect the overall opinion of their companies. Due to the sensitive nature of information provided by interview participants and their respective companies, we are unable to disclose the raw interview data or specific details about ways of working. Finally, in a 2.5-hour workshop with roughly 20 participants, many of whom were interviewees, we presented and discussed our findings with illustrative quotes.

3.2 Data Analysis

We applied thematic analysis, following the guidelines by Saladana [71]. We used a mixed form of coding, where we started with a number of high-level deductive codes based on the interview questions, then we started inductive coding, adding new codes while going through the transcripts. At least three of the researchers worked together to code each of the transcribed interviews.

We observed saturation after five interviews, as not many new inductive codes emerged. In a second round of coding, a new group of at least two researchers per interview reviewed the interview transcripts and verified the codes. Finally, we used pattern coding to identify emerging themes and sub-categories. The final codes of each interview and the assignment of the statements of the interviewees to the sub-categories were reviewed by an additional independent researcher.

To illustrate our points, we use a number of interview quotes. For increased anonymity, participants are assigned a random identifier, such that P1 does not necessarily match to interview A.

As noted in Section 1, the results of the theme Requirements Engineering have been previously published [30]. This study enriches the findings on the RE theme, by reporting on the quality and systems and software engineering themes. The ecosystem and business, data, and annotation themes have been reported as well by Heyn et al. [38]. Although the article focuses on different themes, the qualitative topics covered in that article and our study here have some overlap, particularly in topics related to data and annotation. However, here, the topics of data and annotation are approached from an RE perspective, while the other article takes an ecosystems and process view on topics and challenges related to perception systems in driving automation systems.

4 Results: Requirements Engineering (RQ1)

Based on the thematic analysis, we divide the RE theme into sub-themes—"Operational Design Domain (ODD), "Scenarios and Edge Cases", "Requirements Breakdown", "Traceability" and "Requirements Specification"—and important topics within each sub-theme. The sub-themes and topics are summarized in Figure 3. We also note how many interviewees discussed each sub-theme. Our findings reflect both RE topics and challenges, addressing RQ1.

4.1 Operational Design Domain (ODD)

An ODD is a description of a domain that a driving automation system will operate in—e.g., the road or weather conditions. As part of RE, one needs to define not only requirements, but assumptions about the domain, context, and scope of operation. Operational context and scope for perception systems is particularly important as the intensity of hazards depends upon the current ODD. ODD-related topics came up in all interviews and were discussed by 12 of the 19 participants.

ODD Definition: ODDs should be captured as part of the requirements specification. Several interviewees mentioned ODD detection—where the system detects that a certain ODD is currently applicable for a driving automation system function—and ODD exit detection—when the ODD is no longer applicable. ODD detection requires information on what to detect and detection accuracy. For example, on highways, a driving automation system needs to detect different dynamic objects than in urban areas.

ODD and Standards: Interviewees state that ODDs are critical, and therefore, it is desirable to follow a standard or process for specifying and defining ODDs. This need has been recognized and new initiatives for the definition of ODD exist, e.g., the interviewees mention the PAS-1883 standard, and we are aware of other standards (e.g., ISO 21448/SOTIF) that include ODDs.

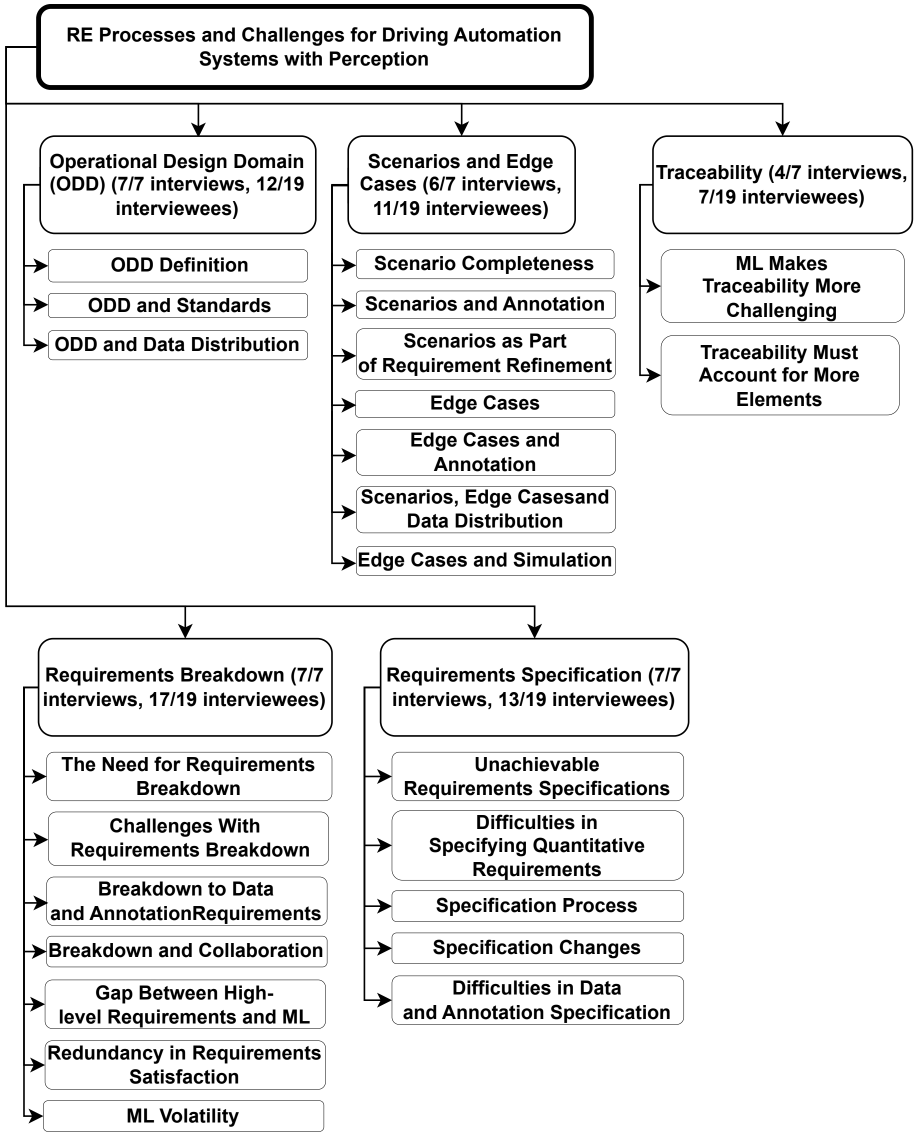

Fig. 3: Mind map illustrating identified RE topics and challenges (**RQ1**) for driving automation systems with perception.

ODD and Data Distribution: One interviewee stated that data distribution requirements are highly influenced by ODDs. For example, camera data can be classified according to descriptions in the ODD, and this mapping can reveal missing data, driving further data collection. As it is not feasible to collect data in all possible contexts, it is necessary to have an efficient sampling process covering the most common ODDs.

> "*If the performance of the model is not good enough in some part of the ODD, for instance during the night or snow weather and so on, then we can select more samples from those areas.*" - P16

Another interviewee pointed out that although ODDs drive data collection, collecting certain types of data required by the ODD can still be very difficult.

> " *... mining for specific use cases. For instance, it is not easy to collect data that contains animals in it. You need some way to mine and find those specific frames which will be sent for annotations and then be used during training.*" - P16

4.2 Scenarios and Edge Cases

Several interviewees described how scenarios are crucial as part of the requirements specification process. In this context, scenarios describe specific operational paths and conditions for a vehicle, and one ODD may include a number of scenarios. As such, although there are links to scenario-based requirements methods [77], there are also clear differences. Scenarios and edge cases came up in six of the seven interviews and were discussed by 11 of the 19 participants.

Scenario Completeness: It is important that perception systems perform correctly and that the vehicle handles failures in as many scenarios as possible. As such, scenarios can help in requirements derivation.

> "*If we refer to the classic system engineering process, I think nowadays it's quite hard ... we are trying to use the scenario to derive the requirements. If we ... see the features or the distribution of the scenarios based on the data from the real world. Then we can derive the high-level requirements based on that data, the scenario database.*" - P4

One interviewee stressed the difficulty of defining and assessing coverage.

> "*How do you define coverage? ... What is the scenario space for pedestrian children? Is it based on how the area you have annotated looks inside of your bounding box? Do you parameterize it on the size of the bounding box, parameterized on conditions around you? How would you divide that space and define it in a way that allows even measures? Have I covered not just enough children, but also enough variety of children?* " - P18

Scenarios and Annotation: Even if all important scenarios are reflected in training data, annotation errors may result in unsafe behavior—e.g., a perception system may recognize a human as a tree during a snowy or rainy day.

> "*We'll pick out some scenarios that we feel (are) likely not correct, for instance, if it's a rainy night, then maybe the annotator is not annotating (people) as accurately as in the day.*" - P8

Scenarios as Part of Requirement Refinement: Our results show that testing through scenarios enables iterative requirements refinement. Engineers iteratively refine their expectations of correct behavior by examining scenarios and capturing observations from simulation or in the field.

> "*... we have to learn through testing, so probably it will start with some rough set of requirements, some obvious set of requirements. Then we will, through real-world testing, discover and learn exactly how we want to behave.* " - P2

> "*It seems like a test-driven development process ... we have the scenarios to drive the development and give more input and also we get the benefit of testing.*" - P4

Edge Cases: Interviewees stated that, in addition to normal scenarios, it is crucial and challenging to deal with edge cases. The interviewees used subtly different terms, such as edge cases, rare cases, and cases that occurred very infrequently. We use the term "edge cases" for simplicity. These cases may be missed by studying data distributions, but are very critical to ensure safety.

> "*The cars ... will end up in situations that no one could predict, that we've never seen before, and somehow we need, even in this situation, one individual car needs to perform better than a human driver, and human drivers are real good at handling edge cases. The neural networks will not do that.*" - P13

Edge Cases and Annotation: Edge cases cause issues by creating confusion among annotators. Data from edge cases is often annotated inconsistently. The topic of annotation is explored in more detail by Heyn et al. [38].

> "*We label whether a vehicle is in our lane or not. But how should you? You can think of so many corner cases when you are out driving. When you are doing a lane change. Which lane are you in then, and how would you then place all the other vehicles or lane lines? Maybe there are double lane lines and which is valid and which is not? This leads to a lot of confusion among annotators.*" - P17

Scenarios, Edge Cases and Data Distribution: One interviewee pointed out that scenarios, and especially rarer edge cases, are important for driving data collection efforts as part of having an effective data distribution. How well edge cases are covered can be an important development metric.

Edge Cases and Simulation: Interviewees stated that collecting data points for particular scenarios from the real world is necessary, but is particularly difficult for edge cases. This makes simulation challenging, as for safety-critical edge cases, practitioners have difficulty safely gathering enough data to run realistic simulations. This makes the process of iterative requirements refinement, as described previously, difficult for requirements associated with edge cases.

4.3 Requirements Breakdown

Requirements breakdown can involve both refining or decomposing requirements. Requirements breakdown was brought up as a topic in all interviews and was discussed by 17 of the 19 participants participants.

The Need for Requirements Breakdown: We see evidence that a traditional requirements breakdown is followed for perception systems. At least one participant spoke of splitting the problem to reduce complexity.

> "*We need to split the problem. We can't do all work at the same time on the complete problem.*" - P12

Another participant described an architectural-oriented breakdown.

> "*Let us say you don't want to collide with an object more than once in a billion hours. This is your top requirement and then you need some kind of architecture or idea of what your system looks like. That should realize this safety goal. This is where we typically come up with a functional architecture, and we start to break down the requirements of the parts of that functional architecture. Then we work. We refine it. The functional architecture becomes a system or logical architecture and we break it down into smaller and smaller pieces.*" - P7

Others describe the importance of separation of high-level requirements from technical requirements to have an upper layer that is resilient to change.

> "*To me, at least the function level will be the same in 100 years because there's no need that you change it. If your function doesn't change, because today you satisfy that function by combustion engine, in the next 50 years by electric, and in the next, I don't know, 100 years by something more intelligent ... By changing your technical system level specifications, you still can satisfy your function.*" - P19

Challenges with Requirements Breakdown: Participants commented on the challenges of connecting high-level requirements to low-level requirements and general challenges with requirements breakdown in this context.

> "*I would say we're working with that challenge and, not that it's an easy one, but we do believe that it's necessary to connect the top-level requirements or the quality of the function, and to map that to quantitative or performance requirements on, for example, perception, precision, and control.*" - P13

> "*What you can do is interact the most closely with … some component, maybe in perception, and these are the ones who would place direct requirements on the previous component, so it is to me a bit of a hierarchical model to approach the difficulties in breaking down the final safety goal to the early stages in our processing chain. I think one tricky thing is, that it's a hierarchical way in some ways, but you also have to go in both directions in that hierarchical model.* " - P6

Several interviewees report that traditional requirements breakdowns cannot be easily applied.

> "*For sure, we will not start with the classical software approach, where you start with some requirements and then keep breaking those down and through the V-Model because it will be impossible to capture the behavior of autonomous vehicle with requirements.*" - P2

Breakdown to Data and Annotation Requirements: Interviewees explained that, although linking functional requirements to system accuracy is often possible, breaking functional requirements into data and annotation requirements is more difficult.

> " *Working with system level requirements, I can look at function requirements and figure out roughly what kind of accuracy we need … That does not necessarily mean that I can tell how precisely annotation has to be, because I need to know how the software works to figure that out. Another translation needs to happen where I gave my requirements to the developers and they have to figure out what kind of accuracy they need from the data to meet the system requirements and with so many translations on the way, it is easy for things to get lost somewhere.*" - P6

> "*…it is difficult to write good requirements on data quality and annotation preciseness and have those links all the way up to feature requirements (Figure 1). Which I think is because of the dimensionality of the problem. The input space is so enormous that it's really tricky to get a single set of requirements there.*" - P15

Breakdown and Collaboration: Challenges arise when teams collaborate to specify quality requirements.

> "*Creating one function would involve multi-team collaboration usually. I guess it's not as easy as evaluating your own system when other people are kind of involved, so you have to come up with scenarios and things to test your algorithms with and could try to come up with a plan.* " - P4

Frequent and direct interaction with the stakeholders can reduce this difficulty and help engineers to identify the requirements. In this case, stakeholders have internal roles in the perception system development.

> "*I think it is a lot of interaction with direct stakeholders in the end ... because the direct consumers of whatever you are producing know exactly what they need to fulfill their own requirements from their own stakeholders. So the negotiation across these interfaces is where the most interaction happens.* " - P9

Gap Between High-level Requirements and ML: When breaking down high-level requirements to very specific requirements on the ML-based perception system, results show that traditional RE practices are able to be applied up to a certain point - even though challenging. However, the breakdown for the ML based components is particularly challenging. As such, there are boundaries within the system where requirements methods change.

> "*If we talked about some other requirements or specifications not for the AD stack. ... those things still can follow the traditional way for critical system. ... if we distinguish those two parts, ... for the black box or part or AD business part, it's hard to follow, but for the rest we still can leverage the classic knowledge.*" - P4

We see that it is difficult to specify requirements for the whole perception system. However, there are often still requirements—in terms of various performance metrics—at a high-level.

> "*If we say the requirements were specified for the entire AD stack, I think it's quite hard to have very precise or detailed specifications for all functions, but actually, we have some high-level metrics like safety, performance, functionality, or traffic comfort metrics ... We have something, but they are very different from the traditional understanding of the specification.*" - P4

Redundancy in Requirements Satisfaction: One interviewee described how requirements are allocated to ensure redundancy in the solution.

> "*We typically try to break down the problem to come up with redundant solutions. You would have one algorithm using one sensor, which has some capacity to detect the pedestrian, and then use another algorithm and another algorithm in parallel. And you use another sensor and ... decompose the problem such that ... it's very unlikely that all of them would miss this pedestrian. That's a way to try and get reasonable requirements on every perception component.*" - P6

ML Volatility: One interview pointed out, due to dependencies between components and the volatile nature of ML, changes in the ML model can cause drastic changes in other parts of the system.

> " *People sometimes start setting requirements on sensors, and then start setting requirements on data, and calibration accuracy, and then also on annotation, preciseness, and that somehow should influence the model accuracy. Maybe one problem we have with ML is that, if there are things slightly off, it cannot just lead to a slight degradation, but to complete degradation of the entire system.*" - P17

4.4 Requirements Traceability

Seven interviewees, across four interviews, brought up points related to traceability in perception systems.

ML Makes Traceability More Challenging: Known requirements traceability challenges are exacerbated by the use of ML and associated data. Interviewees described that when systems or modules fail to meet particular key performance indicators (KPIs), tracing the source of the issue is difficult due to the combination of ML models and traditional code. Traceability was discussed in four out of our seven interviews and by seven out of 19 participants.

> "*I think what is important at the end is the KPIs on the rightmost features of the figure (Figure 1). Then if you want to track down why it is not working, it's not very easy to find which module is not working as supposed to, or maybe it works, but in a combination of something else, it creates some kind of strange behavior.* " - P14

Traceability Must Account for More Elements: It is important that traceability be maintained not just between code and requirements, but also with ML elements—e.g., models and datasets—that determine the overall functionality.

> "*I think it is important to keep track of exactly which data was used to train the model, and be able to also show that to the general public if needed, right? ... having traceability all the way through development is something we aim for.*" - P8

Typically, trace links would link to typical elements like requirements and safety goals, but now they should also link to scenarios.

> "*I don't want to say something that is wrong, you need this traceability, and then when you trace back you see that, OK, I had a safety goal that was talking about this specific scenario.* " - P19

4.5 Requirements Specification

Aspects of documentation and requirements specification were discussed in all interviews, and by 13 of 19 participants.

Unachievable Requirements Specifications: Two interviewees mentioned that sometimes clients provide unachievable requirements, even though requirements specifications are clear and precise.

> "*Sometimes clients come to us with a very well written set of requirements, like we want this annotator and want this precision or accuracy ... Then they send us data. But when we start looking at the data, it turns out that, given this data, these requirements are basically impossible to meet.*" - P18

Difficulties in Specifying Quantitative Requirements: Due to confidentiality, interviewees were not able to elaborate on specific target levels for quantitative requirements. However, they did reflect generally about the difficulty in determining quantitative quality targets.

> "*... for model accuracy, what does success look like in functional safety? If you can recognize 99% rebounding boxes of possessions, is it good enough? If you have a recall of 100%, but your precision is only 50%, would that be good enough?*" - P17

Specification Process: One interviewee emphasized that documentation of the rationale and goals of the project can serve as a form of requirement specification.

> "*I think it's valuable to actually document after what principles you're working, document the problem you're trying to solve and that is basically a set of requirements, even if they're not necessarily traceable upwards all the way.*" - P15

Specification Changes: The uncertain and highly iterative nature of perception systems and their development environment means that specifications are particularly prone to change.

> "*Requirements at any level are not something that is static. They should reflect your current best interpretation. These things can change because your understanding or your development process changes or the environment changes because there are suddenly new demands on how something is supposed to perform or you learn something new about the system or its environment.* " - P15

Difficulties in Data and Annotation Specification: One interviewee said that specifying data requirements is difficult and different from functional specification, as it is hard to identify features and ensure data quality upfront.

> "*It's very different how you write a data specification ... it's hard to know what the future expects and what type of classes we want and how we want to combine certain objects ... we future proof our datasets quite well by specifying. We do specify a lot of classes.*" - P5

Another interviewee reported that it is difficult to specify quality (non-functional) requirements on data and annotation, and to understand how qualities affect model performance.

> " *I work a lot with image quality before any ML is involved. Even that is very difficult to quantify. We can have very much right objectively measurable requirements on image quality, sharpness. Then how those translate to the actual performance of a ML algorithm is not at all linear.*" - P16

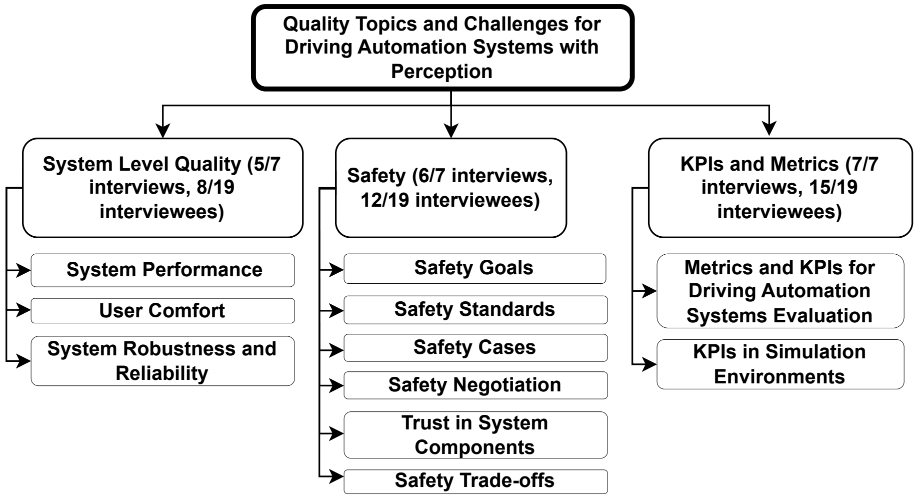

Fig. 4: Mind map illustrating relevant quality topics and challenges (**RQ2**) for driving automation systems with perception.

Another participant described challenges in specifying requirements for data annotation when dealing with external partners. It is difficult to have an upfront, detailed specification of data classes and accuracy levels. Instead, data specification needs to be developed iteratively and experimentally with suppliers.

5 Results: Quality (RQ2)

Based on the thematic analysis, we divide the Quality theme into the following sub-themes—"System-level Quality", "Safety", and "KPI and Metrics"—and important topics within each sub-theme. The sub-themes and topics are summarized in Figure 4. We also note how many interviewees discussed the sub-theme. These topics and challenges are used to address RQ2.

5.1 System-level Quality

This first sub-theme focuses on quality at the system level. This sub-theme came up in five interviews, and was discussed by eight of the 19 participants.

System Performance: As in other safety-critical domains, practitioners are required to satisfy performance and accuracy requirements for the entire system.

> "*We have to come up with the precise numbers for how accurate our model needs to be after training on that data. This also comes down to different transitions from the end requirement or the safety goal that we set up for the product.*" - P9

These requirements are typically quantitative in nature, and often are prescribed as a bounded range of values rather than a single specific value to account for non-determinism. As an example, one interviewee discussed a parking situation where the exact performance depends on associated sensors and other factors, but the resulting behavior is deemed correct as long as it falls within the specified range.

> "*The vehicle should park no farther than X [numbers removed] centimeters from the vehicle behind and no farther away than the Y centimeters from the wall... That's typical performance. And [that depends] on the ultrasonic sensor unit, and that's up to the supplier.* " - P11

However, the interviewees emphasized that performance measurement baselines are not always easy to define and that the current standards do not provide specific information on how to establish statistical expectations. Instead, the standards expect deterministic, specific behavior. Even without the use of ML, many factors result in non-determinism in embedded systems. The inclusion of ML leads to even further potential for non-determinism.

> "*That way of thinking doesn't work with that specific standard (ISO26262), because that standard doesn't have these kind of numbers. You can't even write and say that things should be correct or the product should be correct with this kind of statistical numbers, but rather should be correct always. And I think that's a bad way of thinking, let's say I design a classical piece of electronics, even that one doesn't work binary: yes or no always.*" - P1

Our interviewees stated that redundancy (e.g., redundant algorithms) can help improve performance. However, effective redundancy should be carefully planned—ideally not just with different approaches, but with different methods and data sets.

> "*... have at least [multiple] parallel algorithms just working on the camera which directly or indirectly would be able to detect the pedestrian.*" - P6

> " *I have this algorithm detecting pedestrians and this other algorithm detecting pedestrians, which is slightly different. If they're both trained on the same data sets and they're both using deep learning methods, OK, how redundant are they then, really? But you could also think, I have these two algorithms and they are not completely redundant, but together they are still better than one, one by one, so we could still deploy both to make the system better. ... And then you might be able to show that with this extra algorithm, and I have better performance.*" - P7

If there are multiple ways to gather data (e.g., radar, cameras, infrared), the possibility exists of sensor fusion. Sensor fusion enhances redundancy and, subsequently, performance in various scenarios.

> "*Radars are very good at detecting orientation and speed of an object, though not to classify it as the cameras are doing. I would love to see camera and radar, due to most of the functionalities from an ADAS perspective, because you need both of them for sensor fusion from a redundancy point of view, from a technology point of view. For object classification, for instance, camera is good, but it's not so good for detecting speed and orientation.*" - P10

Another interviewee pointed out, however, that redundancy comes with a high cost and may lower usability:

> "*One pretty high level trade off would be cost or usability to the user. You could pack dozens of compute units and redundant sensors, which would drain of course the money of the user, or the customer, and the battery as well. You might just be able to drive around for a few minutes. But, you would shift the trade off towards more safety or availability just by piling up more redundancy. Of course, at some point, that it just is not feasible to use in an actual product that you could provide to customers.*" - P9

User Comfort: The topic of user comfort in a vehicle also came up in our interviewes. For example, in addition to redundancy, multiple sensors may improve a user's comfort.

> "*An AD car with only cameras and the radar on top of them adds redundancy, so if something fails you have a second set of sensors. The radar could be useful, detecting speed so it makes a more comfortable ride, because you can detect more easily the speed and adapt to it quicker than if you only have cameras.* " - P11

System Robustness and Reliability: Several interviewees brought up topics surrounding robustness and reliability. Evaluating robustness and reliability requires consideration of a complex system made up of many sub-components—many of which come from suppliers. Driving automation systems consists of several components, and their robustness and reliability is taken into consideration for the allocated ASIL (Automotive Safety Integrity Level) levels on the system.

> "*We have a total goal of robustness, reliability that includes the perception point of view, the components from a hardware point of view. We are discussing with the system suppliers which levels we are on right now from an ASIL point of view, and also from a reliability point of view, and confidence point of view.* " - P10

One interviewee pointed out the importance of data quality for robustness. If data quality is low, estimates of robustness may be inaccurate. In such situations, safety measures must be put in place to account for this uncertainty.

> "*We need to build robust systems. At least we're able to detect that now incoming data is really poor, so we should take some safety measures,* " - P15

5.2 Safety

As might be expected, given the criticality of driving automation systems, safety was one of the most popular discussion points in interviews. This sub-theme came up in six of the seven interviews, and was discussed by 12 of the 19 participants. Interviewees brought up "Safety Goals", "Safety Standards", "Safety Cases", "Safety Negotiation", "Trust in System Components", and "Safety Trade-offs" as important topics in this sub-theme.

Safety Goals: Multiple interviewees mentioned the importance of establishing safety goals. Such goals are the starting point of safety-related requirements specification. They are established at a high level, then connected in a hierarchy to lower-level individual system functions, where measures are put in place to ensure that the safety goal is realized throughout the system. Note that existing work has pointed out that safety goals for AD are entirely different from those defined for an ADAS system [81].

> "*Safety goals will basically be the starting point of the safety requirement specification... Then this will be the first parent requirement in the safety hierarchy and, then, in the next level—by some analysis, like fault analysis—you will try to understand what in the function level can violate that safety goal and introduce safety mechanisms.* " - P19

Safety Standards: Safety is one of the most important quality attributes of driving automation systems. To ensure safety one needs to make sure that the nominal function is safe as well as in presence of faults according to the faults assumptions. However, the guidance on applying AI is limited and best practices are not established for safety argumentation.

Safety aspects of driving automation systems are normally structured according to ISO26262. However, it is commonly discussed if ISO26262 can actually address driving automation systems in an efficient way when they include machine learning, e.g., [33].

> "*I think also one well-known problem here is that the sort of traditional V-model safety is [prescribed by] ISO26262, and ... it's not useful for systems that use machine learning, or at least it's not sufficient.*" - P6

However, beyond ISO 26262 and SOTIF, there are emerging standards and ISO documents focusing on AD (e.g., TS5083 [4], AMLAS [32]) and AI/ML (e.g., ISO PAS8800 [5], TR5469 [6]).

[4] `https://www.iso.org/standard/81920.html`
[5] `https://www.iso.org/standard/83303.html`
[6] `https://www.iso.org/standard/81283.html`

> "*The only thing that I'm sure I can say is that it's a tough topic and this is why there is these different standards ...* " - P19

Several interviewees mentioned SOTIF (Safety of the Intended Functionality [7]) as a further standard for functional safety of driving automation systems—used in conjunction with ISO26262. However, neither standard is fully adequate for ensuring both system and function safety, especially given that ML blurs the boundaries between functional and system safety.

> " *We have a standard which has been widely used within the automotive industry for more than 10 years which addresses part of the overall safety of the system or function (ISO 26262), but leaves out other parts. Now there is another standard called SOTIF. The way to go for automotive seems to be to handle these two aspects of safety separately. But I believe that the border between the two is not very strict, it's not black and white. It's kind of floating, and especially when it comes to machine learning ... we do not really know yet how to address with the existing standards.*" - P13

One interviewee also mentioned multiple emerging safety standards for AI, including the TR5469 standard for functional safety in AI [8], PAS8800 for road vehicle safety for AI [9], and TR24029 for assessment of the robustness of neural networks (AI) [10] . Another interviewee referenced UL4600, which offers guidance on, among other topics, data safety [50].

Two interviewees also stated that they follow PAS-1883, an emerging standard for defining ODDs [11].

> "*You need to define ODDs and specify them... so far, there has not been a standard for how you define and how you specify ODDs. But we are very well aware about ... BSI standard PAS-1883. PAS is becoming a standard or instruction to follow. It's an initiative to spell out how you should define an ODD, and how you cascade down, and how to specify an ODD.*" - P10

For these new standards, some challenges can be expected. One interviewee noted the challenge of applying the standards.

> "*I think it is also a little bit difficult for many of the engineers who are not experts in functional safety according to the standard to really understand what it means to them in their daily life...*" - P6

[7] `https://www.iso.org/standard/77490.html`
[8] `https://www.iso.org/standard/81283.html`
[9] `https://www.iso.org/standard/83303.html`
[10] `https://www.iso.org/standard/77609.html`
[11] `https://www.en-standard.eu/pas-1883-2020-operational-design-domain-odd-taxonomy-for-an-automated-driving-system-ads-specification/`

Furthermore, one interviewee pointed out that although standards can help developers avoid costly software failures, conforming to safety standards can also be quite costly.

One interviewee added that the top-down approach of safety standards does not match does not match the current working procedures (e.g., agile way of working).

> "*I think the standards traditionally are very much a top-down approach, but in reality a development project does not work like that of course. Requirements at any level are not something that is static.* " - P15

Safety Cases: Safety cases are structured arguments used to show documented evidence that the system is sufficiently safe. Safety cases are an important aspect of driving automation systems development. However, it is important to ensure that the safety case matches the reality experiences by the developers:

> "*The safety case is one of the strongest ones that really matters... in order to reach that one, it's good for us to interact with development, and ensure that ours is as good as possible, so we can actually reach 0 cases of an unsafe break...* " - P1

Data requirements are part of the evidence used in a safety case. It is important to have data management and data quality as part of the safety case argumentation.

> "*The data requirements then will be part of the safety case. We are talking about both the training data used to actually train the models and what we use in verification and validation in the end.*" - P10

When asked about the impact of ML on safety cases, interviewees noted that safety cases can be defined in a modular manner over components of the system. ML-based components, naturally, must be part of such safety argumentation. However, one participant pointed out that safety cases are not yet well defined for driving automation systems, as the new standards are still upcoming.

> " *...the safety case, what does safety case mean? Here we say, well, we have a safety case. The methodology behind it is really tricky and not solved fully within the community.*" - P16

Another interviewee describes safety case argumentation as a joint process conducted with OEMs and suppliers:

> "*... We will definitely be interested in how our supplier has solved that problem and what safety argumentation they give us because we need to integrate it in in our safety case for the vehicle. We always have joint reviews, and we go to the detail on that.*" - P11

Safety Negotiation: Driving automation systems are often built, at least in part, from existing components offered by external suppliers. Therefore, just as safety case argumentation is built jointly with suppliers, safety requirements must also be developed in conversation with suppliers.

Interviewees described a process where suppliers are assumed to have already assessed the basic safety of their components outside of the context of a particular driving automation system before being contacted. This can be considered a safety element out of context, as discussed in [47]. Then, the driving automation system developers present safety requirements for a particular driving automation system as part of contract negotiation.

> "*When we meet them [suppliers] for the first time and we start talking we expect them to have done their homework over safety elements out of context—that they can present their assumptions and the safety holes that they have for their systems based on those assumptions. That means that they know what they're talking about, and they're good as a supplier. ... and then what we do is take care of the responsibility for the complete vehicle. Once we know our functions, we do a HARA (Hazard Analysis and Risk Assessment, from ISO26262), and from those we derive functional safety requirements that we give to them. You will need to fulfill this [requirement] to get this project.* " - P11

> " *Most of the time, the things already exist ... For example, you have a vision system that can detect the objects, and then you go to suppliers. Most of the time, except some examples recently, it's this agile way, that every supplier works together. Normally, it's like that safety element out of context—that you developed the component, and then you go to the OEM, and then you introduce your product or your safety element and then they check in ISO part 10.* " - P19

Trust in System Components: Because a driving automation system is constructed using components developed externally, interviewees noted that trust is initially a potential issue. Developing driving automation systems requires trusting that externally-developed components are safe and reliable. GPS, in particular, was brought up by multiple interviewees due to the potential safety hazards of inaccurate GPS readings and map data. Obviously, they still will have a safety case at the end.

> "*I have always faced a challenge, which is how to trust the GPS, and even the map. Safety wise, what we have always been told is, that a GPS should we treat it as a [ASIL level] QM (quality management, i.e., all assessed risks are tolerable), and the same with maps. But whenever you pinpoint to some specific target, or a supplier, or a sensor provider, the story is very different. They claim that they started talking about the accuracy and things like that, but functional safety-wise we will be fine. ... but you never can overcome that act of faith. I would say, show me some study that you have done to say that what you provide is up to an integrity level; that has never happened in my opinion. For me, it is still a challenge that we need to make a leap of faith when we select a specific supplier.*" - P11

Safety Trade-offs: Attainment of certain quality attributes, such as safety or performance, tends to require trade-offs with other quality attributes. Safety was a high priority for interviewees, even if it came at the cost of other quality attributes or driving automation system functionality. For example, one participant discussed limiting function availability to improve safety, choosing to only consider automation under certain situations rather than widely allowing the car to be driven autonomously:

> "*You then come into the compromise between availability and safety of the function. Standing still is safe, but it's also very bad functionally, as you want to go somewhere. That's why you have a car. It is possible to build a self driving car that does the job, but it will not be safe. Then, if you don't build it safely, then it will not take you anywhere. ... we will only have a very limited ODD of our first AD function.* " - P7

Another interviewee discussed limiting vehicle functionality to improve safety, in this case, compromising the speed of the vehicle for improved safety:

> "*Speed is an obvious trade-off, so we also trade high speeds for safety. We go at lower speeds. Speed is such a dimensioning parameter in all the safety work since all the risks are high when you increase speed.*" - P7

5.3 KPIs and Metrics

Assessments of driving automation systems quality are made by tracking certain performance metrics, often called "Key Performance Indicators" (KPIs), and comparing the attained value to selected thresholds. This sub-theme was discussed in all seven interviews, by 15 of the 19 participants. Interviewees brought up "KPIs in Simulation Environments" and "KPIs and Metrics for Driving Automation Systems Evaluation" as topics in this sub-theme.

Metrics and KPIs for Driving Automation Systems Evaluation: For evaluation of the driving automation system, interviewees explained that—rather than specifying deterministic properties—they track a set of high-level metrics related to safety, performance, functionality, comfort, and other factors. These metrics can be tracked over the execution of many different scenarios, either in simulation or in a real vehicle, then statistical analysis of the collected observations can be used to make an assessment of the driving automation system.

> " *If we say the requirements were a specification for the entire driving automation system stack, I think definitely it's quite hard to have very precise or detailed specification for all the functions, but actually we have some high level metrics like safety metrics or performance metrics, functionality or traffic comfort metrics, those metrics are on a very high level, which means we can use those metrics*

> *combining with the scenario database and then we run millions of the scenarios and get the statistical analysis report. We can't say we don't have anything for testing or validation. We have something but they are very different from the traditional understanding of the specification.*" - P4

For evaluation of the low-level ML components—such as specific classification models—interviewees stated that they employ standard metrics used in other ML domains. They note, however, that they pay close attention to the specific data in the dataset to ensure that careful evaluation is performed.

> "*For the model accuracy, we use very standard metrics. IoU (Intersection over Union), mean average, precision, or mean intersection over union and things like that. We also try to make sure that we, for example, work on a per-class basis, where we just don't average out a person in the pedestrian as the same values.*" - P5

When interacting with ML, assessments of safety—and the design of the functionality under assessment—must take into account uncertainty in the ML output. Uncertainty estimation is employed at all levels, from algorithm design, to data selection, to evaluation.

> "*To ensure some form of safety measures from the model, we do produce uncertainty estimations from the outputs. So those are also used in the data selection of course to look for what type of data is uncertain. What do we need to learn more from? As well as when it's used in the perception chain, we can give an uncertainty estimation of a certain object so that we can make good enough decisions when it comes to the driving.*" - P3

The KPIs of systems dependent on ML will ultimately be determined by the data used for training and validation of the ML components. Therefore, to ensure that KPIs are informative and realistic, high-quality training and validation data must be used.

> "*... of course [we] want to have as good data as possible for training, but that is going to be expensive. Because your KPI are not going to be any better than the validation data, right? So, if your validation data is broken, then your KPIs are going to be broken, regardless of how good your quality data is.* " - P1

> "*It's very hard to get an objective quality measure then or, like, you will need a lot of background knowledge of the average size of cars in order to do that consistently.*" - P10

Often, data is reused, especially for rare cases. However, data reuse is one of many factors that can affect the realism of the attained KPIs.

> "*I think one interesting thing is also [with regard to KPIs], how do you measure positive performance and how do you verify positive performance, that is really tricky. I think it's very important that we be able to reuse the data and also some of the rare cases, it's not easy to collect again. It's very important to be able to use them again... if you have done so, then how you can guarantee that the KPIs that you get after training or during validation are representative of the real world?* "
> \- P15

One interviewee also stressed the importance of communicating KPIs and metrics for data quality and variance to users who use annotated data.

> "*You should produce such KPIs (on data variance) and communicate this to the users. In this case, the users say alright, or if we need to adjust anything from ... the data collection perspective. And the same goes with the annotation as well, if they are of good quality or not, essentially it's KPI numbers to the users as input.*"
> \- P8

KPIs in Simulation Environments: Because simulations may not accurately reflect the real world, KPI observations gathered from simulations may also not match the observations that would be made in reality. One interviewee was skeptical of using values from simulations as evidence of safety.

> "*During the verification and validation, how you can verify that the KPIs that you get during the validation actually show the reality? Especially if you do some sort of synthetic simulation, ... then it's not very clear to me how one can argue the safety aspect, that we can reach the same KPI in real world.*" - P16

Another interviewee noted that KPIs should be compared between simulations and real-world testing to help show that the simulation is realistic.

> "*This is something you can do by looking at KPIs for single frames situation on both of them [reality and simulation], and if they are comparable, then I think you can have good argumentation for why your closed loop simulation is actually reflective... But if these two numbers are not the same then that's rough.* " - P1

One interviewee described using KPIs for selecting simulator vendors.

> "*We have some KPIs we are using. And the that means we are going to select 3, 5 of them mostly. Those KPIs, we have sent over to them [vendors]. I asked them about that in the RFI [request for information], and then depending on that, we will pinpoint one [vendor]. And then send an RFQ [request for quote] to them, but the KPI, there are many KPIs in there.*" - P10

6 Results: Systems and Software Engineering (RQ3)

As with the previous results, we divide the Systems and Software Engineering theme into sub-themes—"SE Methodology", "Verification and Validation

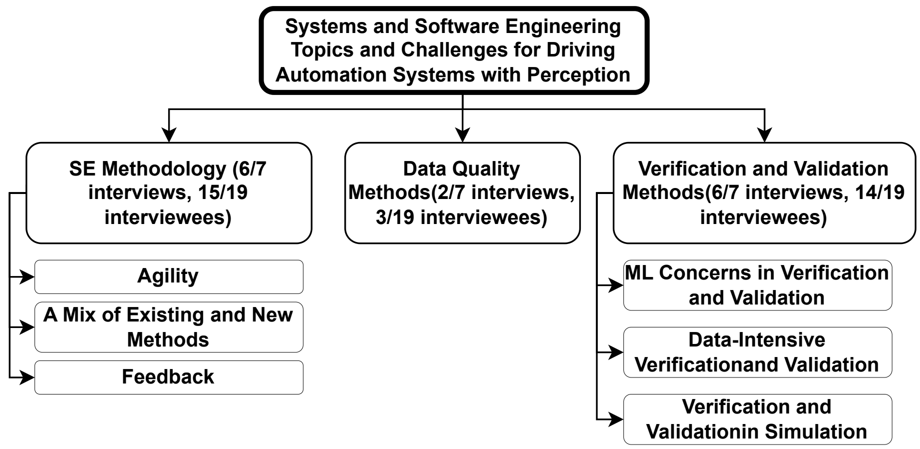

Fig. 5: Mind map illustrating relevant systems and software engineering topics and challenges (**RQ3**) for driving automation systems with perception.

Methods", and "Data Quality Methods"—and important topics within each sub-themes. The sub-themes and underlying topics are summarized in Figure 5. We also note how many interviewees discussed the sub-theme. These sub-themes and topics address RQ3.

6.1 SE Methodology

Software engineering methodologies refer to the frameworks or approaches that guide the processes, activities, and tasks involved in software development [73]. Such methodologies provide a structured and systematic way to plan, design, develop, test, evaluate, and deliver software products to customers. Different methodologies (e.g., waterfall, agile) offer a distinct set of principles, practices, and techniques to manage the software development lifecycle. The interviewees described several aspects of their software development methodologies and how they have changed in the face of ML use. This sub-theme came up in six of the seven interviews, and was discussed by 15 of the 19 participants. Interviewees brought up agility, feedback and a mix of existing and new methods as topics in this sub-theme.

Agility: As context, many of our interviewees have transitioned or are transitioning to a more agile way of systems development for driving automation system development. For example, participants are moving from a process based on the traditional V-model to a specific agile framework, e.g., SAFe.

> "*We are using SAFe as a formal agile framework in our software development process for autonomous vehicles.*" - P2

Such transitions and the use of agile methods at a large scale bring their own challenges unrelated to ML, e.g., [48]. These issues form a background to our exploration of ML use in driving automation systems.

Previous work focusing on methodological transitions has pointed out culture clashes between more agile and traditional ways of working [16, 61]. One interviewee points out similar clashes between control engineers — who develop and test software and hardware – and software developers, who are focused on data and simulations.

> "*I think the biggest clash is the way of working. Control engineers want to approach things in their way. They want to synthesize things and test things for real in the vehicle. Whereas the software companies have a more data-driven background and they start off with the data and they just work in different ways.*" - P2

A Mix of Existing and New Methods: Many traditional methods and processes where, e.g., requirements are defined and broken down to the different parts of the system will still apply. This includes large-scale agile methods, such as SAFe. However, several interviewees reported that it is difficult to implement existing software engineering methods when the system includes machine learning because of non-deterministic behavior. In addition, since the use of machine learning is rather opportunistic or technology-driven, the existing processes can be infeasible because of non-deterministic behavior of such systems. The participants focus on integrating and adapting methods while using machine learning.

> "*If we distinguish those two parts, we just say for the black box or part or autonomous driving business part, it's hard to follow [existing methods], but for the rest, we still can leverage the classic (development) knowledge.*" - P3

Some interviewees described this mix of methods by describing both a top-down and a bottom-up start. Traditionally, development would start with up-front requirements elicitation, with a break-down facilitating system development (similar to Sec. 4). However, in the case of driving automation systems, the development process is less top-down but rather composed of many different components that must fulfill the product definition on top level.

> "*It's like bottom-to-top and then back to the bottom. We have just like a rough starting point, like we want to drive this route, and that's basically the scenario or the high-level requirements, and then we build some algorithms and try it. We start in that sense in the bottom and then, when we have something in place, we can start testing in a structured way with replay of logs and with the structured scenario database and all that and then we learn.*" - P2

Several participants describe using a highly iterative development method for driving automation systems. Although agile methods are difficult to apply out-of-the-box, as they still require some upfront requirements knowledge (e.g., user stories, backlogs), iterations appear to be key for machine learning development.

6.2 Data Quality Methods

The criticality of data quality in machine learning systems has an effect on development methods and ways of working. This sub-theme came up in two of the seven interviews, and was discussed by three of the 19 participants. Interviewees emphasized the importance of placing a high priority on data quality.

> "*First, you focus on data itself. You can try to identify the purpose of using this data in your system and identify other possible issues. There are some guidelines like data safety guideline from the safety Critical System Club, and then they have a very structured guideline about how to identify them. I think they are hundreds of identified possible issues about the data in safety critical system. So, first you can probably try to explore the data itself.*" - P4

To ensure the quality of the data, tools, and requirements, one interviewee indicated that traditional quality assurance methods are still valid, even with new technologies.

> " *For example, if we use tools, we do tool qualification. If we use data, we do data qualification. Then we can follow the classic way, like we identify the possible issues and we do a fault analysis and then we identify the possible preventing or handling measures and implement those measures in the system and then we prevent the possible problems.*" - P4

One participant discussed the importance of feedback as part of the data annotation process, which involves creating documents to clarify annotation uncertainties.

> "*They just gave feedback to people (annotators) and just like from then on that's the way that they perform (in annotating).* " - P3

6.3 Verification and Validation Methods

Verification and validation methods refer to the systematic techniques and activities employed to confirm that the system meets the specified requirements and intended purposes, and the system has been designed, developed, and implemented correctly. This sub-theme came up in six of the seven interviews, and was discussed by 14 of the 19 participants.

Interviewees brought up machine learning concerns in verification and validation, data-intensive verification and validation, and verification and validation in simulation as topics in this sub-theme. Note that, although there is a difference between validation and verification, interviewees are using both terms (verification and validation) as a high level definition of general verification and validation activities.

Machine Learning Concerns in Verification and Validation: Practitioners must select appropriate verification methods for the specific machine learning algorithms being deployed. Verification of machine learning is a relatively new field of research, and practitioners may need to monitor ongoing research regarding particular types of machine learning.

> "*There is some research about how to verify the neural network like abstract interpretation and then you can use the safety engineering method like to design some measures or set a scope for the neural network and try to have like redundant pipeline and to monitor this algorithm.*" - P3

A common challenge when performing verification is that the model acts like a black box—it is difficult to infer how a model makes its decisions. From the engineering perspective, to ensure that the whole system works well, it can be important to focus on the explainability of the model. Practitioners start with something that they feel might work, and then check the results, then adjust accordingly. Once the system achieves basic functionality, if any limitations (e.g., poor performance) of the artificial intelligence algorithms or the neural network is observed, then the engineers attempt to understand the model to identify the reason.

> "*If we see the limitations of the artificial intelligence algorithms or the neural network, then probably we want to dig into this black box and try to figure out what's the reason behind that.*" - P4

Although, explainability affects the way in which systems are developed, our interviewees reported that it is typically not their first concern. They often focus on getting the system working first, then consider the qualities such as explainability.

> "*I think we're in the stage where we're just trying to get the whole thing working first and then we would go into more understanding why it works the way it is.*" - P2

Data-Intensive Verification and Validation: With the incorporation of machine learning and artificial intelligence, the verification and validation of the perception systems become more data-intensive. For example, an interviewee pointed out that effective verification and validation requires representative data selection.

> "*Also from the validation point of view, we also need to get the right data for the data driven validation.*" - P4

One interviewee stated that practitioners require correct data, and that they perform statistical analysis on the collected data as part of verification. The participant also emphasized the need for an efficient sampling method to cover the most common ODDs, as extensive data collection is not practical.

> "*Also from the validation point of view, we also need to get the right data for the data-driven validation. For example, we also need to qualify the camera data ... for this kind of data we need to have statistical analysis, which means we count how many scenarios are located in this distribution. Because we don't want to collect all the data. As much as possible is not efficient, if we just think we have a good or a clever sampling process. Then we can see the distribution of the scenarios and the scenes from the reality. Then we can say we have covered the most common ODDs and then we can use this data set to do validation.* " - P4

Driving automation system practitioners often meet the need for verification data with synthetic data, emulating data collected from sensors or cameras. Due to the complexity of collecting real data for scenarios that appear very rarely (e.g., edge cases), it would be beneficial to use such synthetic data, but the effort of validating synthetic data could higher than actually collecting this kind of data.

> "*I don't believe that synthetic camera data or LIDAR data has sufficient fidelity to be used for validation yet. It's lacking in many aspects. And to prove that it is actually useful for validation would probably require more data than not even using it to begin. [Note: validating the usefulness of synthetic data would require so much data that synthetic data is no longer needed]*" - P7

Traditional systems are often verified by comparing observations to specific expected output. Given the quantity of data needed and the difficulty of specifying a deterministic outcome, driving automation systems with machine learning may need to be verified based on statistical analyses.

> "*I think it will come down much more to statistics in the end. We know we have accumulated a lot of driving and from all this driving we managed to create a set of different datasets that altogether captured the whole ODD and statistically then we, you know perform well.*" - P2

Verification and Validation in Simulation: Interviewees reported a shift toward verification and validation in simulated environments from using actual hardware because of the safety-critical nature of driving automation systems, which require different steps in the validation and verification process to rule out as many issues as possible before going on the roads. Practitioners collect data and conduct simulations to test the driving automation system in a virtual environment. One interviewee stated that they are focusing more on scenario-based approaches for verification and validation.

> "*From the simulation team, we are trying just to shift the direction from the classic embedded system world ... that's the reason why we have the scenario-based approach for validation. So if we see the data from some other companies like Waymo or Uber, they spend 99.95% of the test cases in their virtual environment,*

> *just 0.05% on the real vehicle. Because some of the scenarios are really dangerous, we can't just to ask a driver to drive on the road and do some to test some edge cases.* " - P4

On the other hand, it can be challenging to perform verification and validation of driving automation system in a simulation. An interviewee points out that there might be limitations when using synthetic simulations for safety argumentation.

> "*When you do the verification and validation of your software, how you can verify that the KPIs that you get there during the validation actually shows the reality. For instance, especially if you do some sort of synthetic simulation, you do validation and verification, then it's not very clear to me how one can argue the safety aspect that we can reach the same KPI in real world.*" - P14

Furthermore, when machine learning is incorporated into the system, it can be difficult or very expensive to create realistic data for the verification and validation process.

Interviewees consider the balance between the percentage of verification that should take place in the simulation and in the real environment. The challenge of realism indicates that some verification should take place on real hardware. However, it is more time consuming to test in the real world than using simulations and therefore, simulations is an efficient method to be used before doing the final tests on the roads. Regardless, interviewees stressed that it is important to ensure that the test cases are representative of the driving scenarios in both the real and simulated environments.

The level of automation also has an impact on how validation and verification in the simulation environment is conducted. For lower levels of automation, simulation can be outsourced to suppliers. However, an interviewee notes that they will need to develop in-house simulation for higher levels of automation.

> "*Well, so far, up to level 2 systems according to SAE Level J 3016 standard, there we have a verification and validation flow strategy. Where we ... collect data and use reference sensors. We can ... send over to the tier 1 supplier for up to Level 2, then they can run their simulations. But, in the future now we will invest in simulations by ourselves for higher automation levels.*" - P10

V &V is also affected by standardization, including new and upcoming standards like TS5083.

> "*And also from a verification and validation point of view we are following the new standardization, TS5083 which is gonna be the final version and release around 2023.*" - P10

7 Summary and Discussion

In this section, we summarize our results, answer our RQs, list future directions for research and practice, and discuss threats to validity.

7.1 Requirements Engineering Topics and Challenges (RQ1)

We have identified a number of RE topics and challenges in Section 4, as summarized by Figure 3. These topics and challenges can be seen as a checklist when working with machine learning-based perception systems—a list of issues that should be considered.

Our interviewees emphasize that the definition and limits of ODDs are an integral part of perception systems, and these ODDs have important impacts on data requirements and collection, confirming findings in Heyn et al. [37]. Similarly, perception systems development relies heavily on the use of scenarios and associated edge cases. Such scenarios play a key role in dictating annotation, data collection and simulation.

In terms of challenges, our results indicate that **ODD detection and ODD exit detection** are challenging, as these require information not only about what to detect in the environment, but also how to detect and the accuracy of the detection confirming findings in [86]. In addition, **data requirements** are highly influenced by the content of an ODD, therefore ODDs can be used to evaluate whether a data distribution is sufficient for good machine learning model performance. However, it is not always easy to **collect the data** specified by ODDs. Heyn et al. also emphasized the importance of ODDs in driving automation systems, and noted the lack of a common definition for ODDs [36]. Our participants go further and mention the need for ODD standardization (and efforts in that regard).

One major challenge is that simulations should reflect **realistic scenarios**, echoed by Acuna et al. [1]. To ensure safe perception, the collected data and scenarios must be thorough, and the perception system must avoid failure in all scenarios. In addition to covering normal scenarios, it is important to **specify edge cases** among scenarios, which are then used to determine data distributions. However, edge cases introduce challenges as they create **confusion among annotators** and are challenging to **test in reality** due to safety concerns.

Breaking down requirements for data and annotations can be very difficult, and additional challenges are introduced due to requirements dependencies and the need for multiple teams to collaborate. In general, we believe that the **gap between standard RE methods and machine learning components** is both a technical gap and a gap in training and backgrounds, as the machine learning components are often engineered by data scientists without a software engineering background confirming the results in [4, 80].

Difficulties in breakdown, machine learning opaqueness, as well as the introduction of more elements to trace (e.g., ODDs, scenarios, training data), make it difficult to establish **traceability**. These challenges add to the known challenges with motivating and using traceability in practice [83].

Creating **specifications for data and annotations** is challenging, as it is difficult to have an upfront specification for data classes, e.g., pedestrians and crosswalks. Furthermore, sometimes machine learning components are assigned unrealistic and **unachievable requirements**. Although requirements change is a frequently acknowledged RE problem [45], with perception systems, the **level of uncertainty and change** is particularly high due to uncertainty about the system, including machine learning, and the environmental targets. **Quantifying quality requirements** (e.g., accuracy) is also particularly challenging in perception systems, echoing the results of Vogelsang and Borg [80].

7.2 Quality Topics and Challenges (RQ2)

As part of the Quality theme, our interviewees have identified a number of topics and challenges. Practitioners consider **performance, reliability, robustness, safety, and user comfort** as important quality attributes. It is interesting to note that the space of qualities that interviewees focused on is generally small, compared to the space of NFR qualities explored in past academic work [28].

Interviewees found that for driving automation systems, **performance was difficult to measure accurately**. As a means of ensuring both performance and safety, redundancy of algorithms and sensors was important, but interviewees noted that **redundancy must be carefully designed**, particularly in terms of data and algorithms, and redundancy as a principle for system design has both limitations and trade-offs.

In the context of driving automation systems, safety is particularly critical, as also noted by [82]. Safety assurance is already challenging for conventional driving automation systems software, and becomes even more challenging with the inclusion of machine learning. Practitioners set safety goals, often in negotiations with component suppliers. **Safety negotiation** with suppliers has been a challenge, including issues of **trust in components and suppliers**. In addition, ensuring driving automation systems safety requires collaboration and effort from different parties—is challenging, confirming findings in [60].Interviewees also acknowledged that safety does not operate in a vacuum, recognizing **safety trade-offs** with, for example, security, availability, and functionality. Safety cases are a critical element of assuring that the safety goals are met, but are also more complex given the uncertainty of machine learning.

To ensure safety, practitioners must comply with **evolving safety and AI standards**, including established standards such as ISO26262 [42]—which are not sufficient to account for the incorporation of machine learning. This confirms the findings of [21, 46]—and underlines the need of newer machine learning-specific standards such as TS5083 [43]. In general, given the wide range of standards, there have been challenges in understanding, managing and conforming with the relevant standards. Although there are significant costs associated with safety incidents, conforming to standards is also costly.

The large input space and non-determinism of machine learning complicate quality assurance. Instead of specifying concrete expected behaviors in key scenarios, quality assurance is performed by **tracking critical KPIs** during the execution of catalogs of scenarios and performing statistical analysis on captured observations. KPIs can be **defined on multiple levels**, including **driving automation systems-specific KPIs** and **standard KPIs for machine learning components (e.g., precision and recall)**. KPI assessment is affected by **training and validation data**, **uncertainty**, and **simulation realism** confirming the results of [82].

7.3 Systems and Software Engineering Topics and Challenges (RQ3)

Our interviews also revealed topics and challenges related to systems and software engineering development methodologies for perception systems. Our findings show that the presence of **machine learning adds further complexity to agile ways of working**. Existing traditional and agile methodologies are not sufficient to meet the needs of large-scale machine learning, echoing the findings of [26]. Our interviewees apply a **mix of methods** using more traditional, top-down engineering in some areas and more iterative, bottom-up development in others. In this way, from a methodological point of view, a **continuous feedback cycle** is key to successful delivery.

The focus on safety further complicates development methods, confirming the findings in [2], as many **safety methodologies do not adequately address machine learning**. The importance of data to perception systems requires changes in development practices. In particular, practitioners need **data quality methods**.

Our findings show that **verification and validation is more challenging and data-intensive in the presence of machine learning**. Data selection and consideration of data quality are required to ensure effective verification. The use and acquisition of **synthetic data** is an important topic, but raises data quality issues. Rather than comparing observed behavior with specific, expected outcomes, **V&V is based on statistical analyses** of quantitative metrics. To gather sufficient observations and limit the risk to vehicle operators, verification and validation uses simulation. However, it is a challenge to have **realistic simulation**, and to **determine in which situations simulation can replace real verification**.

7.4 Future Directions in Research and Practice

Some of the identified challenges in RQ1-3 are relatively new from an RE and SE perspective (e.g., ODD detection, missing edge case, the proliferation of machine learning-related safety standards, machine learning verification and validation), while others have been long recognized (e.g., traceability [83], specification changes [45], and quality trade-offs [17]). Our findings point to a number of new research topics. We outline these areas, highlighting examples of existing state-of-the-art work on these topics.

Although the focus of our work has been on perception systems, we believe that many of the topics and challenges found apply more generally to other domains reliant on machine learning. For example, challenges breaking down specifications would hold due to the volatility and opaqueness of machine learning. Future work should contrast RE challenges and practices in other machine learning-enabled domains.

ODD Methods: Our findings illustrated the importance and challenges associated with ODD development as part of complex machine learning systems. Existing work has focused on various aspects of ODD development as part of autonomous driving, e.g., [20], including a consideration of safety [27, 37], but our industrial partners still find this topic a challenge. ODD can be linked to broader work on context in RE, e.g., [8,15], but future work should explore what aspects of ODD context are domain-specific or general.

Data Requirements: How to capture and define requirements over data is an issue that should be a focus of future work. Although previous work has looked at data-driven RE, e.g., [58], this focuses more on gathering standard requirements from sources such as social media, rather than requirements for the data needed for machine learning. Other work has looked at data quality, but from an era before the rise of machine learning, e.g., [12].

Requirements Traceability with Machine Learning: Tracing requirements to system elements is essential for safety argumentation and change management, but becomes challenging when traces must include machine learning components like models and data. Although traceability has been heavily investigated from a requirements perspective [78], traceability for machine learning is only just starting to be explored, e.g., [62]. Other works look specifically at traceability from the perspective of data provenance or data lineage as part of machine learning [52].

Scenarios as Specifications: Given the challenges of defining up-front, complete requirements for driving automation systems, practitioners have turned to scenarios both for specifying data, and verification, including simulation. Scenarios have been an active topic in RE, e.g., [72], but mainly for improving the quality of gathered requirements, including completeness, and not as a stand-in for traditional requirements. Using scenarios as specifications data-driven development requires further attention.

Quantifying Machine Learning Requirements: Interviewees expressed difficulties with placing specific targets on quality requirements for driving automation systems, e.g., performance requirements, echoing some of the challenges found in recent work [28]. Although quantifying quality requirements, or metrics, has been an active area of investigation for many years, e.g., [13], most work on metrics for machine learning is specific to particular qualities, e.g., uncertainty [67]. Setting targets for such metrics is particularly difficult and context specific.

Redundancy: Redundancy as a means to improve performance and safety arose as a prominent issue. Redundancy has been studied in general for systems engineering, e.g., [18], and has been studied from the perspective of multiple machine

[12] `https://iso25000.com/index.php/en/iso-25000-standards/iso-25012`

learning models, e.g., ensemble learning [69]. However, more consideration can be made at combining these perspectives, considering machine learning redundancy at the system level.

Safety Standards and Machine Learning: Safety standards and machine learning is an active topic, e.g., [33], but we see in our findings a proliferation of many possible standards or frameworks which can be applicable for driving automation systems, but the selection or integration of these multiple standards has not been extensively explored.

Large-scale Agile with Machine Learning: Much work has been dedicated to reporting and making recommendations concerning transitions to agile methods for large-scale systems, e.g., [48]. However, most work does not consider the challenges introduced with machine learning. In terms of machine learning development, methods like CRISP-ML [76], and a more recent focus on MLOps [51] attempt to guide development. But the combination of these machine learning and data-driven methods with established, large-scale agile methods like SAFe[13] is still mainly unexplored.

Large-scale Agile with Machine Learning and Safety: Adding to the complexity of the previous direction, such large-scale, agile, machine learning-enabling methods should also be usable in safety-critical contexts. Safety challenges as part of large-scale agile have been investigated [75], but not in an explicit machine learning context.

Verification and Validation for Machine Learning: Testing and related activities for machine learning is already an active area of investigation [84]; however, we feel we should highlight this direction, in particular the areas of synthetic data curation and simulation, as it was raised by several interviewees. Others have begun to investigate the utility of synthetic data, e.g., [40], but further investigations in a driving automation systems or safety-critical contexts are needed.

7.5 Threats to Validity

Internal Validity: We internally peer-reviewed the interview guide and conducted a pilot interview to improve the guide and process. We sent a preparation email to all the interview participants with the details and purpose of the interview study. To maintain consistency in the interview process, at least three authors conducted each interview, with two authors present in all interviews.

All interviews were conducted in English, and the auto-generated transcripts were 'fixed' by authors by listening to audio recordings and correcting any transcription errors. Note that the working language of each company was English, so the language should not have created barriers.

Although qualitative coding always comes with some bias, we mitigated this threat by following established literature [71], coding in multiple rounds, using

[13] `https://scaledagileframework.com/`

inductive and deductive codes, and having multiple authors participate in each round of coding, with in-depth discussion on code meanings and assignments.

External Validity: We used a mixture of purposive and snowball sampling. As our study needed a certain set of expertise to answer our questions, we could not conduct random sampling, using our networks and their contacts. Still, due to the size of the study, with participants covering a wide variety of roles with varying experience levels, covering differing company roles and sizes in the perception system ecosystem, we believe we have a relatively representative sample. Furthermore, we argue that we reached a sufficient point of saturation with our interview data, as we noticed a sharp decline in emerging codes after analyzing the fifth group interview.

Note that one cannot link participants to interviews and companies, this is done deliberately to protect the anonymity of our participants. Although this may affect transferability of our results, we feel this level of anonymity does not greatly hurt our results. Though our study results are limited to perception systems in DAS, we argue that some findings can apply to other safety-critical or perceptions systems. This applicability should be explored in future studies.

8 Conclusion

Our study investigated requirements engineering, quality, and systems and software engineering topics and challenges during the development of DAS. We interviewed 19 participants from five companies and identified a number of topics and challenges that have a major impact on the specification, development, and quality of DAS. The results of this study offer guidance to practitioners and suggest future research directions in the intersection of requirements engineering, software quality, development methodologies, and machine learning to help mitigate the challenges practitioners are facing.

Acknowledgements: Support for this project was provided by Vinnova prestudy 2021-02572. We also thank all interview participants.

References

1. Acuna, D., Philion, J., Fidler, S.: Towards optimal strategies for training self-driving perception models in simulation. Advances in Neural Information Processing Systems **34**, 1686–1699 (2021)
2. Adler, R., Feth, P., Schneider, D.: Safety engineering for autonomous vehicles. In: 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W). pp. 200–205. IEEE (2016)
3. Ågren, S.M., Knauss, E., Heldal, R., Pelliccione, P., Malmqvist, G., Bodén, J.: The impact of requirements on systems development speed: a multiple-case study in automotive. Requirements Engineering **24**(3), 315–340 (2019)
4. Ahmad, K., Abdelrazek, M., Arora, C., Bano, M., Grundy, J.: Requirements engineering for artificial intelligence systems: A systematic mapping study. Information and Software Technology p. 107176 (2023)

5. Ahmad, K., Bano, M., Abdelrazek, M., Arora, C., Grundy, J.: What's up with requirements engineering for artificial intelligence systems? In: 2021 IEEE 29th Int. RE Conf. (RE). pp. 1–12. IEEE (2021)
6. Al Islam, M.N., Ma, Y., Alarcon, P., Chawla, N., Cleland-Huang, J.: Resam: Requirements elicitation and specification for deep-learning anomaly models with applications to uav flight controllers. In: 2022 IEEE 30th International Requirements Engineering Conference (RE). pp. 153–165. IEEE (2022)
7. Ali, M.A., Yap, N.K., Ghani, A.A.A., Zulzalil, H., Admodisastro, N.I., Najafabadi, A.A.: A systematic mapping of quality models for AI systems, software and components. Applied Sciences **12**(17), 8700 (2022)
8. Ali, R., Dalpiaz, F., Giorgini, P.: A goal-based framework for contextual requirements modeling and analysis. Requirements engineering **15**, 439–458 (2010)
9. Allmann, C., Winkler, L., Kölzow, T., et al.: The requirements engineering gap in the oem-supplier relationship. Journal of Universal Knowledge Management **1**(2), 103–111 (2006)
10. Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., Zimmermann, T.: Software engineering for machine learning: A case study. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). pp. 291–300. IEEE (2019)
11. Arpteg, A., Brinne, B., Crnkovic-Friis, L., Bosch, J.: Software engineering challenges of deep learning. In: 2018 44th euromicro conference on software engineering and advanced applications (SEAA). pp. 50–59. IEEE (2018)
12. Belani, H., Vukovic, M., Car, Ž.: Requirements engineering challenges in building AI-based complex systems. In: 2019 IEEE 27th Int. RE Conf. Workshops (REW). pp. 252–255. IEEE (2019)
13. Boehm, B.W., Brown, J.R., Lipow, M.: Quantitative evaluation of software quality. In: Proceedings of the 2nd international conference on Software engineering. pp. 592–605 (1976)
14. Borg, M., Englund, C., Wnuk, K., Duran, B., Levandowski, C., Gao, S., Tan, Y., Kaijser, H., Lönn, H., Törnqvist, J.: Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry. arXiv preprint arXiv:1812.05389 (2018)
15. Broens, T., Quartel, D., Van Sinderen, M.: Capturing context requirements. In: Smart Sensing and Context: Second European Conference, EuroSSC 2007, Kendal, England, October 23-25, 2007. Proceedings 2. pp. 223–238. Springer (2007)
16. Chan, F.K., Thong, J.Y.: Acceptance of agile methodologies: A critical review and conceptual framework. Decision support systems **46**(4), 803–814 (2009)
17. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-functional requirements in software engineering, vol. 5. Springer Science & Business Media (2012)
18. Coit, D.W.: Maximization of system reliability with a choice of redundancy strategies. IIE transactions **35**(6), 535–543 (2003)
19. Cooling, J.: The The Complete Edition–Software Engineering for Real-Time Systems: A software engineering perspective toward designing real-time systems. Packt Publishing Ltd (2019)
20. Czarnecki, K.: Operational design domain for automated driving systems. Taxonomy of Basic Terms “, Waterloo Intelligent Systems Engineering (WISE) Lab, University of Waterloo, Canada (2018)
21. Falcini, F., Lami, G.: Challenges in certification of autonomous driving systems. In: 2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). pp. 286–293. IEEE (2017)

22. Farrell, M., Mavridou, A., Schumann, J.: Exploring requirements for software that learns: A research preview. In: International Working Conference on Requirements Engineering: Foundation for Software Quality. pp. 179–188. Springer (2023)
23. Felderer, M., Ramler, R.: Quality assurance for ai-based systems: Overview and challenges (introduction to interactive session). In: Software Quality: Future Perspectives on Software Engineering Quality: 13th International Conference, SWQD 2021, Vienna, Austria, January 19–21, 2021, Proceedings 13. pp. 33–42. Springer (2021)
24. Fisher, M., Collins, E., Dennis, L., Luckcuck, M., Webster, M., Jump, M., Page, V., Patchett, C., Dinmohammadi, F., Flynn, D., et al.: Verifiable self-certifying autonomous systems. In: 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). pp. 341–348. IEEE (2018)
25. Fujii, G., Hamada, K., Ishikawa, F., Masuda, S., Matsuya, M., Myojin, T., Nishi, Y., Ogawa, H., Toku, T., Tokumoto, S., et al.: Guidelines for quality assurance of machine learning-based artificial intelligence. International journal of software engineering and knowledge engineering **30**(11n12), 1589–1606 (2020)
26. Giray, G.: A software engineering perspective on engineering machine learning systems: State of the art and challenges. Journal of Systems and Software **180**, 111031 (2021)
27. Gyllenhammar, M., Johansson, R., Warg, F., Chen, D., Heyn, H.M., Sanfridson, M., Söderberg, J., Thorsén, A., Ursing, S.: Towards an operational design domain that supports the safety argumentation of an automated driving system. In: 10th European Congress on Embedded Real Time Systems (ERTS 2020) (2020)
28. Habibullah, K.M., Gay, G., Horkoff, J.: Non-functional requirements for machine learning: An exploration of system scope and interest. In: 2022 IEEE/ACM 1st International Workshop on Software Engineering for Responsible Artificial Intelligence (SE4RAI). pp. 29–36. IEEE (2022)
29. Habibullah, K.M., Gay, G., Horkoff, J.: Non-functional requirements for machine learning: Understanding current use and challenges among practitioners. Requirements Engineering pp. 1–34 (2023)
30. Habibullah, K.M., Heyn, H.M., Gay, G., Horkoff, J., Knauss, E., Borg, M., Knauss, A., Sivencrona, H., Li, J.: Requirements engineering for automotive perception systems: An interview study. In: International Working Conference on Requirements Engineering: Foundation for Software Quality. pp. 189–205. Springer (2023)
31. Habibullah, K.M., Horkoff, J.: Non-functional requirements for machine learning: understanding current use and challenges in industry. In: 2021 IEEE 29th Int. RE Conf. (RE). pp. 13–23. IEEE (2021)
32. Hawkins, R., Paterson, C., Picardi, C., Jia, Y., Calinescu, R., Habli, I.: Guidance on the assurance of machine learning in autonomous systems (amlas). arXiv preprint arXiv:2102.01564 (2021)
33. Henriksson, J., Borg, M., Englund, C.: Automotive safety and machine learning: Initial results from a study on how to adapt the iso 26262 safety standard. In: Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems. pp. 47–49 (2018)
34. Hesenius, M., Schwenzfeier, N., Meyer, O., Koop, W., Gruhn, V.: Towards a software engineering process for developing data-driven applications. In: 2019 IEEE/ACM 7th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE). pp. 35–41. IEEE (2019)
35. Heyn, H.M., Knauss, E., Malleswaran, I., Dinakaran, S.: An investigation of challenges encountered when specifying training data and runtime monitors for safety

critical ml applications. In: International Working Conference on Requirements Engineering: Foundation for Software Quality. pp. 206–222. Springer (2023)
36. Heyn, H.M., Knauss, E., Muhammad, A.P., Eriksson, O., Linder, J., Subbiah, P., Pradhan, S.K., Tungal, S.: Requirement engineering challenges for AI-intense systems development. In: 2021 IEEE/ACM 1st Workshop on AI Engineering-SE for AI (WAIN). pp. 89–96. IEEE (2021)
37. Heyn, H.M., Subbiah, P., Linder, J., Knauss, E., Eriksson, O.: Setting AI in context: A case study on defining the context and operational design domain for automated driving. In: Int. Working Conf. on RE: Foundation for Software Quality. pp. 199–215. Springer (2022)
38. Heyn, H., Habibullah, K., Knauss, E., Horkoff, J., Borg, M., Knauss, A., Jing Li, P.: Automotive perception software development: Data, annotation, and ecosystem challenges. In: 2nd International Conference on AI Engineering–Software Engineering for AI. IEEE (2023)
39. Highsmith, J.: Adaptive software development: a collaborative approach to managing complex systems. Addison-Wesley (2013)
40. Hittmeir, M., Ekelhart, A., Mayer, R.: On the utility of synthetic data: An empirical evaluation on machine learning tasks. In: Proceedings of the 14th International Conference on Availability, Reliability and Security. pp. 1–6 (2019)
41. Horkoff, J.: Non-functional requirements for machine learning: Challenges and new directions. In: 2019 IEEE 27th Int. Requirements Engineering Conf. (RE). pp. 386–391. IEEE (2019)
42. ISO: ISO 26262:2018: Road vehicles — Functional safety. International Organization for Standardization, Geneva (2018), `www.iso.org`
43. ISO: ISO/CD TS 5083: Safety for automated driving systems - Design, verification and validation, under development. International Organization for Standardization, Geneva (2023), `www.iso.org`
44. Jain, A., Patel, H., Nagalapatti, L., Gupta, N., Mehta, S., Guttula, S.C., Mujumdar, S., Afzal, S., Mittal, R.S., Munigala, V.: Overview and importance of data quality for machine learning tasks. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2020)
45. Jayatilleke, S., Lai, R.: A systematic review of requirements change management. Information and Software Technology **93**, 163–185 (2018)
46. Jenn, E., Albore, A., Mamalet, F., Flandin, G., Gabreau, C., Delseny, H., Gauffriau, A., Bonnin, H., Alecu, L., Pirard, J., et al.: Identifying challenges to the certification of machine learning for safety critical systems. In: European Congress on Embedded Real Time Systems (ERTS 2020) (2020)
47. Johansson, R., Sivencrona, H.: Developing seooc-original concepts and implications when extending to ads. In: CARS 2021 6th International Workshop on Critical Automotive Applications: Robustness & Safety (2021)
48. Kasauli, R., Knauss, E., Horkoff, J., Liebel, G., de Oliveira Neto, F.G.: Requirements engineering challenges and practices in large-scale agile system development. Journal of Systems and Software **172**, 110851 (2021)
49. Knight, J.: Fundamentals of dependable computing for software engineers. CRC Press (2012)
50. Koopman, P.: Ul 4600: What to include in an autonomous vehicle safety case. Computer **56**(05), 101–104 (2023)
51. Kreuzberger, D., Kühl, N., Hirschl, S.: Machine learning operations (mlops): Overview, definition, and architecture. IEEE Access (2023)
52. Kästner, C.: Machine Learning in Production: From Models to Products (2022)

53. Liebel, G., Tichy, M., Knauss, E., Ljungkrantz, O., Stieglbauer, G.: Organisation and communication problems in automotive requirements engineering. Requirements Engineering **23**(1), 145–167 (2018)
54. Luo, Y., Zhang, X.Y., Arcaini, P., Jin, Z., Zhao, H., Zhang, L., Ishikawa, F.: Hierarchical assessment of safety requirements for configurations of autonomous driving systems. In: 2022 IEEE 30th International Requirements Engineering Conference (RE). pp. 88–100. IEEE (2022)
55. Lurie, Y., Mark, S.: Professional ethics of software engineers: An ethical framework. Science and engineering ethics **22**, 417–434 (2016)
56. Lwakatare, L.E., Raj, A., Bosch, J., Olsson, H.H., Crnkovic, I.: A taxonomy of software engineering challenges for machine learning systems: An empirical investigation. In: Int. Conf. on Agile Software Development. pp. 227–243. Springer, Cham (2019)
57. M. Mahally, M., Staron, M., Bosch, J.: Barriers and enablers for shortening software development lead-time in mechatronics organizations: A case study. In: Proc. of the 2015 10th Joint Meeting on Foundations of SE. pp. 1006–1009 (2015)
58. Maalej, W., Nayebi, M., Ruhe, G.: Data-driven requirements engineering-an update. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). pp. 289–290. IEEE (2019)
59. Mallozzi, P., Pelliccione, P., Knauss, A., Berger, C., Mohammadiha, N.: Autonomous vehicles: state of the art, future trends, and challenges. Automotive Systems and SE pp. 347–367 (2019)
60. Nahar, N., Zhou, S., Lewis, G., Kästner, C.: Collaboration challenges in building ml-enabled systems: Communication, documentation, engineering, and process. In: Proceedings of the 44th International Conference on Software Engineering. pp. 413–425 (2022)
61. Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of migrating to agile methodologies. Communications of the ACM **48**(5), 72–78 (2005)
62. Njomou, A.T., Africa, A.J.B., Adams, B., Fokaefs, M.: Msr4ml: Reconstructing artifact traceability in machine learning repositories. In: 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). pp. 536–540. IEEE (2021)
63. Pei, Z., Liu, L., Wang, C., Wang, J.: Requirements engineering for machine learning: A review and reflection. In: 2022 IEEE 30th International Requirements Engineering Conference Workshops (REW). pp. 166–175. IEEE (2022)
64. Pernstål, J., Gorschek, T., Feldt, R., Florén, D.: Software process improvement in inter-departmental development of software-intensive automotive systems–a case study. In: Int. Conf. on Product Focused Software Process Improvement. pp. 93–107. Springer (2013)
65. Poth, A., Meyer, B., Schlicht, P., Riel, A.: Quality assurance for machine learning–an approach to function and system safeguarding. In: 2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS). pp. 22–29. IEEE (2020)
66. Pradhan, S.K., Heyn, H.M., Knauss, E.: Identifying and managing data quality requirements: a design science study in the field of automated driving. Software Quality Journal pp. 1–48 (2023)
67. Psaros, A.F., Meng, X., Zou, Z., Guo, L., Karniadakis, G.E.: Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. Journal of Computational Physics p. 111902 (2023)

68. Ribeiro, Q.A., Ribeiro, M., Castro, J.: Requirements engineering for autonomous vehicles: a systematic literature review. In: Proc. of the 37th ACM/SIGAPP Symposium on Applied Computing. pp. 1299–1308 (2022)
69. Sagi, O., Rokach, L.: Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **8**(4), e1249 (2018)
70. Salas, E., Goodwin, G.F., Burke, C.S.: Team effectiveness in complex organizations: Cross-disciplinary perspectives and approaches. Routledge (2008)
71. Saldaña, J.: The coding manual for qualitative researchers. The coding manual for qualitative researchers pp. 1–440 (2021)
72. Seyff, N., Maiden, N., Karlsen, K., Lockerbie, J., Grünbacher, P., Graf, F., Ncube, C.: Exploring how to use scenarios to discover requirements. Requirements Engineering **14**, 91–111 (2009)
73. Sommerville, I.: Software Engineering, 9/E. Pearson Education India (2011)
74. Staron, M.: Requirements engineering for automotive embedded systems. In: Automotive Systems and SE, pp. 11–28. Springer (2019)
75. Steghöfer, J.P., Knauss, E., Horkoff, J., Wohlrab, R.: Challenges of scaled agile for safety-critical systems. In: Product-Focused Software Process Improvement: 20th International Conference, PROFES 2019, Barcelona, Spain, November 27–29, 2019, Proceedings 20. pp. 350–366. Springer (2019)
76. Studer, S., Bui, T.B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., Müller, K.R.: Towards crisp-ml (q): a machine learning process model with quality assurance methodology. Machine learning and knowledge extraction **3**(2), 392–413 (2021)
77. Sutcliffe, A.: Scenario-based requirements engineering. In: Proc.. 11th IEEE Int. RE Conf., 2003. pp. 320–320. IEEE Computer Society (2003)
78. Torkar, R., Gorschek, T., Feldt, R., Svahnberg, M., Raja, U.A., Kamran, K.: Requirements traceability: a systematic review and industry case study. International Journal of Software Engineering and Knowledge Engineering **22**(03), 385–433 (2012)
79. Villamizar, H., Escovedo, T., Kalinowski, M.: Requirements engineering for machine learning: A systematic mapping study. In: 2021 47th Euromicro Conf. on SE and Advanced Applications (SEAA). pp. 29–36. IEEE (2021)
80. Vogelsang, A., Borg, M.: Requirements engineering for machine learning: Perspectives from data scientists. In: 2019 IEEE 27th Int. Requirements Engineering Conf. Workshops (REW). pp. 245–251. IEEE (2019)
81. Warg, F., Skoglund, M., Thorsén, A., Johansson, R., Brännström, M., Gyllenhammar, M., Sanfridson, M.: The quantitative risk norm-a proposed tailoring of hara for ads. In: 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). pp. 86–93. IEEE (2020)
82. Winner, H., Lemmer, K., Form, T., Mazzega, J.: Pegasus—first steps for the safe introduction of automated driving. In: Road Vehicle Automation 5. pp. 185–195. Springer (2019)
83. Wohlrab, R., Steghöfer, J.P., Knauss, E., Maro, S., Anjorin, A.: Collaborative traceability management: Challenges and opportunities. In: 2016 IEEE 24th Int. RE Conf. (RE). pp. 216–225. IEEE (2016)
84. Zhang, J.M., Harman, M., Ma, L., Liu, Y.: Machine learning testing: Survey, landscapes and horizons. IEEE Transactions on Software Engineering **48**(1), 1–36 (2020)
85. Zhang, R., Albrecht, A., Kausch, J., Putzer, H.J., Geipel, T., Halady, P.: Dde process: A requirements engineering approach for machine learning in automated

driving. In: 2021 IEEE 29th International Requirements Engineering Conference (RE). pp. 269–279. IEEE (2021)

86. Zhang, X., Tao, J., Tan, K., Törngren, M., Sanchez, J.M.G., Ramli, M.R., Tao, X., Gyllenhammar, M., Wotawa, F., Mohan, N., et al.: Finding critical scenarios for automated driving systems: A systematic mapping study. IEEE Transactions on Software Engineering **49**(3), 991–1026 (2022)