CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Deriving Contextual Definition and Requirements from Use Cases of Autonomous Drive

A Qualitative Exploratory Study

Master's thesis in Computer Science and Engineering

JENNIFER LINDER & PADMINI SUBBIAH

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2021

Master's thesis 2021

Deriving Contextual Definition and Requirements from Use Cases of Autonomous Drive

A Qualitative Exploratory Study

JENNIFER LINDER & PADMINI SUBBIAH

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2021

Deriving Contextual Definition and Requirements from Use Cases of Autonomous Drive
A Qualitative Exploratory Study
JENNIFER LINDER & PADMINI SUBBIAH

Supervisor: Hans-Martin Heyn and Eric Knauss, Department of Computer Science and Engineering
Advisor: Stefan Andersson, Oliver Brunnegård, and Olof Eriksson, Veoneer
Examiner: Christian Berger, Department of Computer Science and Engineering

Master's Thesis 2021
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2021

Deriving Contextual Definition and Requirements from Use Cases of Autonomous Drive
A Qualitative Exploratory Study
JENNIFER LINDER & PADMINI SUBBIAH
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Context-aware systems, like Advanced Driving Assistance Systems, need to be clearly defined for which context they can operate within i.e., the Operational Design Domain needs to be clearly defined to ensure safe operation. The complex and dynamic environment that those systems operate in makes it challenging to effectively describe the context and the Operational Design Domain. The lack of common definitions and a standard process for deriving contextual definition and requirements from use cases decelerate efficient progress of the development of safe context-aware systems. There is a clear need for established and common definitions for the key concepts: operational design domain and context. Further, requirements for context-aware systems need to be efficiently related to the context they are valid for, so safety can be ensured during operation. Based on qualitative research, this study interviewed thirteen automotive experts to explore the definition of the key concepts and the process itself. The study provides qualitative models for definitions of the key concepts: use cases, operational design domain, requirements, and context. Further, the current processes and what works well in the industry are explored. Challenges and improvements for the process of deriving contextual definition and requirements from use cases are presented. The result shows that there is a lack of common process across the industry and there are several different understandings of the definitions for the key concepts: operational design domain and context. The challenges identified further shows the need for a more established and common process, and highlights certain areas of concern that industries could improve. The major contributions of the thesis are an eminent understanding of the key concepts for development of context-aware systems and the highlighted challenges areas that should be addressed. The results can be used for creating improved processes, practices, or templates for common definitions of operational design domain and context.

Keywords: Operational design domain, requirements, use cases, context, contextual definition, autonomous driving, qualitative research, process, requirements engineering, thesis.

Acknowledgements

This thesis has been performed within the Department of Computer Science and Engineering, for the division of Software Engineering, at Chalmers University of Technology.

First and foremost, we would like to thank our supervisors Hans-Martin Heyn and Eric Knauss for their immense support, guidance, and showing genuine interest in our thesis. Their constructive feedback and discussions have helped us to progress past the various roadblocks we have encountered in the thesis. We would like to thank our industry advisors, Stefan Andersson, Olof Eriksson, and Oliver Brunegård, for the support provided, interesting discussions, and the great collaboration throughout the thesis. Great thanks to our examiner Christian Berger for taking his time to give us valuable and constructive feedback and evaluating our thesis. Also, a sincere thank you to the participants of the interviews and focus group, who generously shared their thoughts, experience, and expertise for the benefit of study.

I, Padmini, owe thanks to my husband, my kids, my parents and my friends for their immense support and motivation throughout the thesis

I, Jennifer, would like to thank my family and friends, and send a special thanks to Alessandro Bonci, for the immense support throughout the thesis.

Jennifer Linder, Gothenburg, August 2021
Padmini Subbiah, Gothenburg, August 2021

Contents

List of Figures

List of Tables

1

Introduction

The real-world is complex and filled with a lot of unexpected scenarios that a human can quickly react to, but a system needs to be designed to handle. If the system cannot operate accurately or act to mitigate the risk of the scenario, the system is not safe to deploy in the real-world. There are various domains that develop systems that must be defined for a certain context. Some examples of domains are aviation, medical, and home automation. Autonomous driving, within the automotive industry, is particularly a popular and growing field with a high competitive nature for new unique features that will assist in the driving experience. Automated Driving Systems (ADS) are able to monitor context and handle all driving operations by itself in a defined operational context. Advanced Driver Assistance Systems (ADAS) are designed to assist the driving [3] e.g., systems like autonomous parking and Adaptive Cruise Control (ACC) [4].

This study is done in collaboration with Veoneer research[1] and this study utilises the automotive industry and ADAS as a platform to investigate the process and the involved concepts. However, the result of the study is still applicable, relevant, and interesting for other domains that deal with development of context-aware systems.

The growing interest in development for systems that will assist, or even take full control, of the vehicle operations is accompanied by a concern for safety i.e., assuring that the system will be able to operate safely and as expected in the defined operational context [5, 6]. Validation of safety for these systems becomes utmost important [5, 6]. A standardised process for deriving contextual definition and requirements from use cases would ease validation of the system and further ensure that safe systems are being developed.

Awareness and understanding of the context is especially crucial in systems where the context is constantly changing, which is the case for usage of systems like ADAS and ADS. Proper information about the context is important for making the right decisions during operation, especially in uncertain environments, failing to do so can have dangerous consequences. To maintain safety, ADAS and ADS need to be context-sensitive, which means they must monitor the vehicle and its environment in real time [7].

[1] Veoneer. “Veoneer Homepage” Veoneer. https://www.veoneer.com/ (retrieved 2021-08-08)

In order to be able to introduce the ADAS into the market before being able to operate like a human, SAE Standard J3016 introduced Operational Design Domain (ODD) that limits and defines in what context the system is operational [1]. Contextual definition is a broader and more general term, which encapsulates ODD. This thesis will focus on the specific term ODD, since it is the most commonly used term within the automotive industry. The ODD definition helps in identifying scenarios handled by an ADAS [8], but there does not exist a common definition for ODD within autonomous driving [2]. There does not either seem to exist a suggestion on how to approach deriving ODD from use cases.

The context is dynamic, and because of this there is a need to consider which context a requirement is valid for. Otherwise a scenario might occur, while operating, that the system is uncertain on how to, or even cannot, adapt to. These kinds of requirements are defined as contextual requirements [9].

This study investigates deriving contextual definition and requirements from use cases through a qualitative study. The qualitative method enables exploration of the process and its components in regards to structure, definition, challenges, and improvements. The data is collected through interviews with automotive experts. The collected data is analysed through thematic analysis and based on the themes identified, the qualitative models are presented. The result is evaluated by a focus group consisting of some of the automotive experts from the interviews.

This study contributes with models that map the definitions of concepts, especially ODD and context. It presents main challenges within the process of deriving contextual definition and requirements from use cases, along with suggested improvements. The result is not only interesting to the automotive field, but also other domains that deal with context-aware systems.

This thesis will start with, in Chapter 1, introducing the statement of the problem, the purpose of the study, the case company, the research questions, and the scope and limitations of the study. In Chapter 2, background and related work to this study will be presented. In Chapter 3, the research methodology for this study will be presented. The chapter will go through the phases of this qualitative study and further explain the reasoning behind the choices. In Chapter 4, the result is presented, that was gathered and analysed from the interviews, workshop, and focus group. The result will start with presenting definitions identified for the concepts e.g., ODD and context, then goes into the processes described from the interviews and what works well. The next half of the results consists of the challenges and improvements to areas of the process identified. Thereafter, in Chapter 5, a discussion is held about the result, its implications, and the validity of the result. In Chapter 6, a conclusion of the thesis is given.

1.1 Statement of the Problem

Use cases, requirements, and contextual definition are important parts of defining how the system should act in the context. To ensure safety during driving, an ADAS needs to be operational within its context, meaning that use cases for the system need to stay within the defined ODD, and never exit [10]. If the use case exits the ODD, then the behaviour of the system is not guaranteed and that could lead to risky, or even dangerous, situations. Therefore, it is of utmost importance that the ODD and requirements are defined clearly and effectively in order to guarantee safety in the real world.

There is a lack of a common definition for ODD. National Highway Traffic Safety Administration (NHTSA) [11] presents different, and not completely aligned, concepts for ODD definition. The format of the concepts presented ranges from public policy to industry guidelines to research [2]. Gyllenhammar et al. put forth that the closest definition for an ODD is the suggested practice that the SAE Standard J3016 presents [10].

This lack of common definition seems to leave it up to the developers to decide how the ODD should be defined for each system and feature. Koopman and Fratrik find that the ODD often is described as quite simplistic [12]. The SAE Standard J3016 gives examples of quite broad categories that an ODD can limit [1]. The broad categories has been criticised by the National Society of Professional Engineers [13]. NHTSA also presents broad categories, of which one is “other domain constraints” [11].

The use cases for a system should include information about which context it can operate within. However, this description is often quite broad and non-specific. Examples of these use cases are presented in the project ADAS&ME [14]. These use cases again, seems to leave it to the practitioners’ interpretation of how to derive ODD. Deriving context, and in turn ODD, from broad and quite generic use cases might take up more resources, and can be expensive, e.g., by the fact that the practitioners needs to interpret each time how the context is defined from the use case.

For systems where the context is known and constant throughout the systems’ active phase, the need to relate requirements to a specific context is less necessary. However, for adaptive systems, it is important to relate a requirement to a specific context, because the context might change while the system is active [15]. These requirements relating to a specific context, for which they are valid, is defined as contextual requirements [16]. Contextual requirements need to be linked to a specific context when designing the system, but may still need to be managed during the systems’ operational time [9, 16].

In order to safely and efficiently move forward with autonomous driving systems, a common definition of key concepts like ODD and context, and a common practice for deriving contextual definition and requirements is important.

1.2 Purpose of the Study

The purpose of this thesis is to investigate and explore the current status, uncover challenges, suggest improvements, and provide qualitative models for definitions of ODD, context, requirements, and use cases. Investigating and exploring the current status will lay the foundation for a proper understanding of how experts understand and work with contextual definition and requirements related to context. It will, together with learning about the challenges that the experts see and face today, serve as material for creating the qualitative models. The qualitative models will aim to explore the understanding of use cases, contextual requirements, and contextual definition and establish the relationships between them. This would provide the field and industry with further ground material and a common base for discussion and understanding.

1.3 Case Company

This thesis is carried out in collaboration with the company Veoneer, Sweden[1]. Veoneer is a global Tier 1 supplier that supplies state-of-the-art software, hardware, and systems directly to Original Equipment Manufacturers (OEM). Their products are designed and manufactured for occupant protection, ADAS, and collaborative and automated driving.

1.4 Research Questions

A qualitative study is the best fit if the research questions are framed in a general and broad way to understand the participant's knowledge, perceptions, and experiences [17]. The research questions in this study are formulated as open-ended questions and focused on the central phenomenon of the study as suggested by Creswell [18]. Three research questions will guide this study.

The first research question will help with establishing an overview of the current practices that work well in the industry:
RQ1: *Which current solutions work well for deriving contextual definition and requirements from use cases?*

The second research question will explore the current challenges in industry regarding contextual definition and requirements:
RQ2: *What are the challenges with deriving contextual definition and requirements from use cases?*

The third research question will help with providing an idea of what support and improvements is needed in the industry:
RQ3: *Which support would be appropriate for deriving contextual definition and requirements from use cases?*

[1] Veoneer. "Veoneer Homepage" Veoneer. https://www.veoneer.com/ (retrieved 2021-04-13)

1.5 Scope and Limitations

This study provides views and information models on the cornerstones of the process of deriving contextual definition and requirements from use cases. This thesis does not create or construct a full prototype, template, process, or practice. However, the result can be used as a basis for future development of such artifacts. The solutions and refinements presented in this thesis are based on the data collected through the study.

The researchers believe this thesis will provide helpful information for other fields where there is a need to develop context-aware systems, even though the data collection is only done with experts within the automotive domain. Also, the findings of the study is applicable to the autonomous drive in general, and not only to a specific company, because data is collected from experts within various companies.

The results for this thesis are discussed with the gathered literature in relation to the topic, however an in-depth look into other fields in regards to development of context-aware systems are left for other researchers.

2

Background and Related work

This chapter presents concepts and relevant literature for the study. The chapter presents and discusses the literature for the main concepts related to this study: use cases, requirements related to context, operational design domain, and context. The chapter will end with presenting literature in relation to the process of deriving contextual definition and requirements from use cases. The literature was gathered by searching for relevant articles related to the subject. Examples of search-terms were: context aware systems, context for AI systems, and operational design domain. Relevant articles and papers were also provided by our supervisors.

2.1 Use Cases

Use cases are descriptions of a system's responsibilities and behaviour, which is most often written in text format [19]. Cockburn states that use cases want to show how the system reacts to different situations with as little text as possible, but still clearly convey the reactions to situations [19]. Use cases are meant to be clearly understood by anyone i.e., they should not be too technical or detailed [19]. This leaves much room for interpretation of what the context exactly is.

The project ADAS&ME describes 7 use cases, which aim to cover many critical driving scenarios for all types of vehicles [14]. The seven use cases listed are the following:

- Attentive Long Haul Trucking
- Electric Vehicle range anxiety
- Driver state-based smooth & safe automation transitions
- Non-reacting driver Emergency manoeuvre
- Long range attentive touring with motorbike
- Rider Faint
- Passenger pick up/drop off automation for buses

These use cases serve as a base for achieving the goals set up in ADAS&ME to take into account autonomous driving as well as information about the driver's current state.

Maurer et al. presents 4 use cases for autonomous drive that are meant to cover some typical scenarios and be used for further discussion [20]. Two introductory versions of use cases and two widely developed versions of autonomous driving are presented below.

- Interstate Pilot Using Driver for Extended Availability
- Autonomous Valet Parking
- Full Automation Using Driver for Extended Availability
- Vehicle on Demand [20]

The use cases serve as proxies for a multitude of possible use cases. It should be noted that certain assumptions are made in the use cases in order to have them be simplified. They also show identification of scenery derived from the use cases, but only consider the predictable ones, because it was meant to serve as a base for understanding of autonomous drive [20].

Supriya's thesis [21] focuses on cloud-based support for Autonomous Vehicles (AV). The thesis presents artifacts that provide input to an off-board control system for AV, introduces use cases in the CD practices, and defines relationships between different entities e.g., use cases and ODD. These artifacts are a bit too specific to the domain to directly utilise in this thesis, but provides further insight in to the topic. The requirements information model especially provides a better grasp of the relationships between entities like the ODD and use cases. Supriya presents five artifacts: a use case classification, a list of use cases, a list of challenges, a requirement information model, and a use case template for higher level automated systems. The use case classification provides a standard for how to classify control tower use cases. The classification helps with the maintenance, prioritisation, and resolve differences in the level of abstraction of the use cases. The list of use cases provides an initial set of twenty-five use cases that will represent the current existing cloud support functionalities. The list of challenges is presented for potential challenges that can occur during implementation of the control tower. The requirement information model gives an idea about the relationship between ODD and different entities like use cases. It also provides an understanding about the relation between ODD and quality attributes. Supriya states that the main take away from the model is "that the ODD that constrains AV, provides requirements for the control tower"[21]. The use case template for higher level automated systems provided, addresses mainly the problems gathered from the study and is a template for control tower solutions.

2.2 Requirements related to context

There are several different requirements concerning different parts of the system. Examples of these are functional requirements, quality requirements, safety requirements et cetera. For automated systems there is a need to have requirements be related to context, which are called contextual requirements by Knauss et al.[9]. Contextual requirements are related to a specific context for which the requirements are valid [9, 22, 23]. Contextual requirements according to Knauss et al. consist of a tuple of the expected behaviours of the system and the context in which they are valid [15]. There exists some research about how to relate requirements to context, both during design time of the system and the systems operational time, but it is an unrepresented research area according to Knauss et al. [15].

During design time, contextual requirements need to be linked to the context they are valid in. Context attributes that might change during operational time need to be identified in order to avoid uncertainty regarding how the system will adapt to the changes in context when the system is running [9]. Completely avoiding incorrect adaptation of requirements during runtime puts a lot of pressure on the developers to have a full understanding of the operational environment at the design stage [9]. Considering every possible context and avoid missing a single one during design is an unfeasible task according to Ramirez et al. [24].

A way to ease up the pressure during the design phase is to consider and handle changing context during the operational time of the system. Knauss et al. suggest how to deal with this uncertainty during runtime, by proposing an approach that can support the system in maintaining up-to-date knowledge about contextual requirements [15].

It is stated by Ramirez et al. [24] that uncertainties can be introduced in specifically an adaptive system, either in the requirement-, design-, or runtime phase. These uncertainties need to be handled, otherwise they can carry through and spread in the next phase. The uncertainties can come from either the system itself or the execution environment of the system. Ramirez et al. [24] presents a template for how to describe an uncertainty and a taxonomy for each phase. The sources for uncertainties in the requirements phase listed are missing requirements, ambiguous requirements, and falsifiable assumptions. Uncertainties at design level originate from two sources; unexplored alternatives for requirements and untraceable design. Unexplored alternatives are caused by the fact that the developers cannot consider all possible design alternatives for satisfying a given set of requirements. Untraceable design happens if specific requirements are not addressed or documented, that are supposed to be satisfied by design decisions made.

2.3 Operational Design Domain

The SAE Standard J3016 [1] describes a detailed taxonomy of six levels of driving automation, ranging from Level 0 (no automation) to Level 5 (full automation).

Figure 2.1 shows the levels of driving automation with features and the limitations of ODD based on SAE standard J3016. In Level 1 and Level 2, the driver mostly performs all the driving tasks, even if there exists automation systems. In Level 3, the automation system performs driving tasks within its ODD. If the system detects that it is about to exit the ODD, then it alerts the driver to take control of the vehicle. In Level 4, the automation system performs all the driving tasks within its ODD and when the vehicle is about to exit the ODD, the vehicle transitions to a minimal risk condition, such as safe stop. In Level 5, the system performs all driving tasks in all conditions i.e., the ODD is unlimited [1].

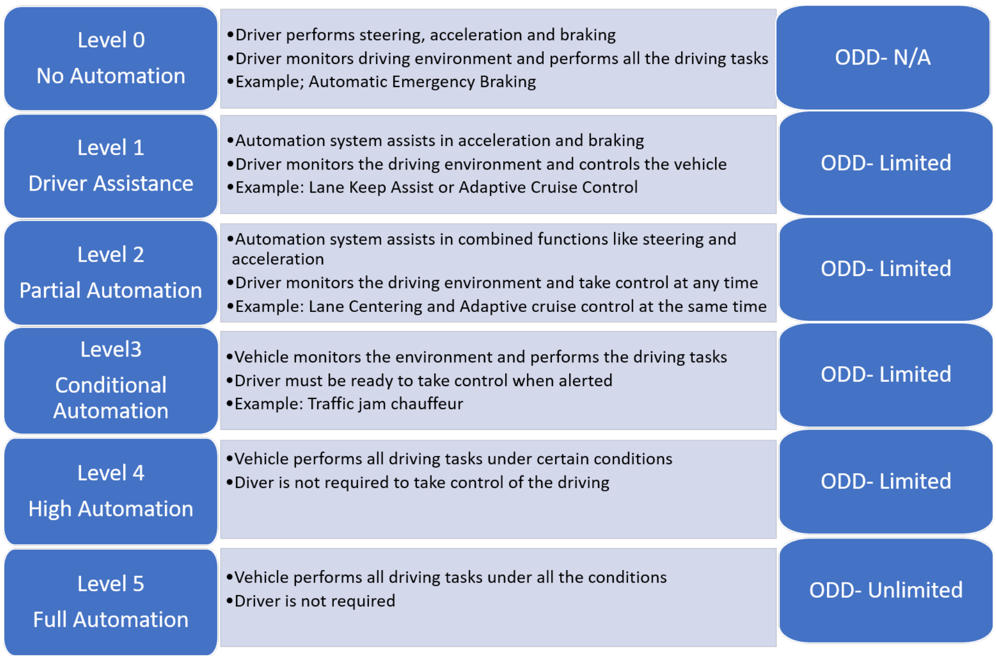

Figure 2.1: Levels of automation related to ODD, adapted from the SAE Standard J3016 [1]

Defining ODD is essential for determining when the AV can operate safely and when it cannot. According to Khastgir [25], the industry, government, and academic experts have differences of opinions on what an ODD is. There are also different interpretations on what an ODD should include. Khastgir explained that the term ‘Scenarios’ has instead been commonly used in the industry. This might be the main cause of confusion among various stakeholders in regards to the term ODD. The ODD and scenarios are related but can not be considered the same [25].

Furthermore, there does not seem to exist one common definition for an ODD, which becomes obvious in the literature review performed by Thorn et al. [2]. A standardised way of describing ODD is however needed to improve communication and collaboration between manufacturers, developers, transportation agencies, and various other stakeholders. It is also important to build public trust, acceptance, and confidence [26]. Several projects like PEGASUS (Germany), NHTSA, et cetera. and standardisation organisations like ISO, BSI, SAE, et cetera. are working on a standardised way of specifying ODD [25]. From the literature [2, 10, 11, 27], the closest to a common standard seems to be the SAE Standard J3016s definition [1]. It is important to note that the SAE Standard J3016 is not a specification, it is a recommended practice that defines ODD as:

> Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics [1, p. 14].

Khastgir [25] mentioned a concept called Operational Design Conditions (ODC) that has been brought up recently. ODC is a superset of ODD which additionally includes AV's capabilities and driver capabilities [25]. According to Colwell et al.[8], defining ODD will be useful in the design process, testing and verification, and online monitoring. Colwell et al. proposed another concept Restricted Operational Domain (ROD) that combines the ODD monitoring and the functional degradation. The ROD is a restricted subset of ODD in which the ADS can still safely operate [8]. Colwell et al. considered ODD as static one which describes a set of constraints in which the system is designed to operate and ROD is dynamic and a subset of the ODD constraints [8].

Thorn et al. [2] defines ODD as the specific operating conditions and domains in which the system is intended to operate. It is intrinsically complex to get clear and complete information about the intended ODD of an ADS function, due to the emerging and competitive nature of ADS technology [2]. The OEM specifies and defines ODD which causes various OEMs to use various sets of ODDs. Thorn et al. also describes that the ODD for each ADS function is likely to differ. Furthermore, defining and quantifying the variability of the ODD elements is quite challenging, for example, the variability in weather conditions like light snow, moderate snow, heavy snow are hard to define. A structured approach is needed to identify and organise various ODD elements. The structured approach will facilitate negotiation and discussion about requirements, capabilities, and testing among the stakeholders [2].

Thorn et al. [2] devise and categorise ODD elements into a taxonomy, which is presented in Figure 2.2. Based on the taxonomy, a checklist for ODD was presented in [2]. The taxonomy and the checklist together provides a structured approach for identifying the ODDs of an ADS function.

All the operational conditions that a use case entails need to be defined within the ODD for the ADAS to operate safely within that use case [10]. The ODD does not just put limitations on the context, but can also set limitations on the system's behavior and state [27]. Gyllenhammar et al.[10] proposes to model the use cases through a "world model", then have operational conditions (OC) that are quantified by that model, both external and internal. The OCs are also defined by the use case. The ODD then contains and collects the OCs. By encapsulating all the OCs that the use case requires, the ODD will contain what is required for the use case and can

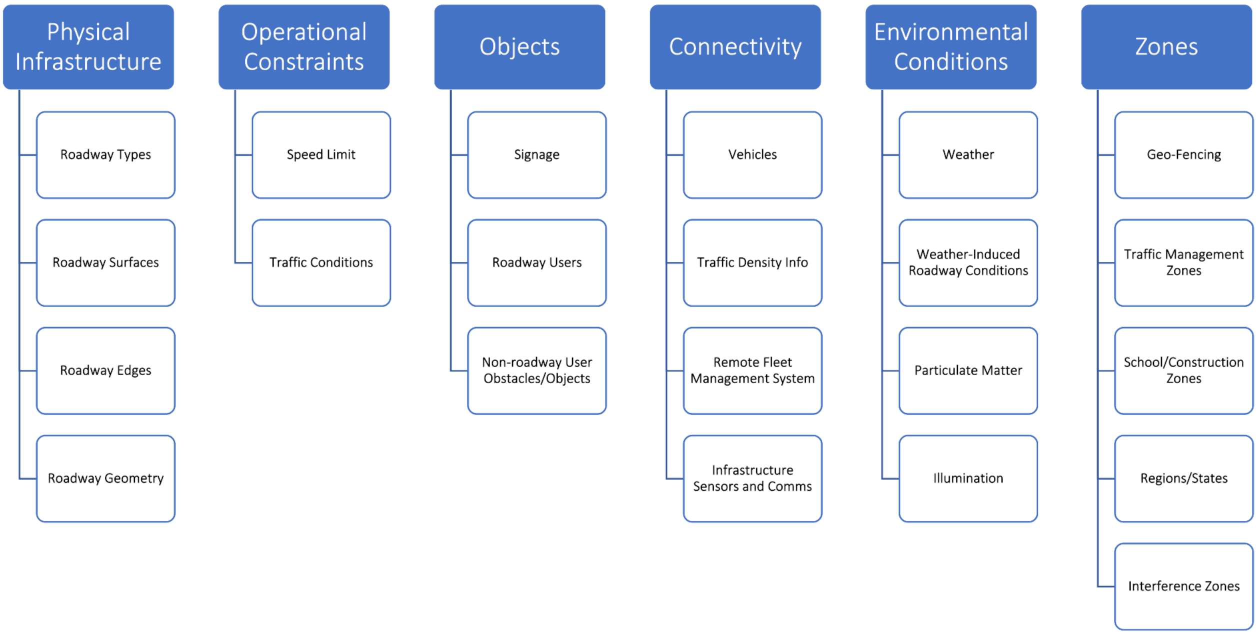

Figure 2.2: Taxonomy of ODD, from Thorn et.al. [2]

therefore operate within the use case. However, it is still said that the ODD should be refined throughout the development process. Gyllenhammar et al. suggests that the primary purpose of the ODD is "to confine the necessary activities of the HARA as well as the verification of the requirements and make them independent of the specific UC considered." [10, p. 7]. However, this puts further pressure on that the ADS never leaves the ODD. Four generic strategies on ensuring this are presented by Gyllenhammar et al.

The first strategy deals with the internal OCs and states that those can be guaranteed to not exits the ODD inherently by the basic feature description. The second strategy deals with some external OCs. The strategy is to only accept trips, or strategic tasks, that can be completed without existing the ODD. In the design phase, these accepted trips or strategic tasks are defined. The third strategy deals with other external OCs. The geographical or temporal triggering conditions need to be identified to ensure not exiting the ODD during runtime. The fourth strategy deals with some external OCs and guarantees them not exiting the ODD by, during run-time, measuring properties related to the OC e.g., rain or road friction. Gyllenhammar et al. also present a framework to categorise these OCs with the intent to ensure that the use case is sufficiently modelled and categories for operational conditions with potential strategies to quantify them.

Defining all the specific operating conditions for which a system is valid can be a taxing task and requires significant experience. But, according to Koopman and Fratrik, it is needed to ensure safe operation in a real-world [12]. Koopman and Fratrik present some important factors that need to be considered when characterising the operational environment. The factors include the following:

- Operational terrain and associated location dependent characteristics
- Environmental and weather conditions
- Operational infrastructure
- Rules of engagement and expectations for interaction with the environment and other aspects of the operational state space
- Communication modes
- Availability and freshness of infrastructure characterisation data
- Other non-stationary objects [12]

These presented factors are not a checklist to guarantee safe deployment of a system in the real-world, but serve as lists to further ensure that no relevant factor is missed during design [12].

The airspace in the aviation industry is classified into different classes, which defines the operational characteristics and procedures for the airplanes [2]. Airplanes need to have some specific equipments and also pilots are required to follow specific regulations and procedures in order to operate in certain airspace domains. Also, missions in space research operate in a specific constrained domain or geographical area, which limits the design. Airplanes and autonomous vehicles have different operational characteristics and procedures. The variations in ODD between autonomous vehicles and airplanes are profound, according to automation specialists in both the automotive and aviation industry. This limits the opportunities for cross-learning [2].

2.4 Context

The term ‘Context’ has been used in various research and engineering fields, and it implies different definitions with regards to different domains. There does not exist a common definition of what a context is [28, 29, 30]. According to Brown and Henricksen et.al. [31, 32], the term ‘Context’ can be considered as environments and situations in which the system will operate. Dey [30] proposed a general definition for context:

> Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves [30, p. 4].

This application-centric definition is widely used in defining the context for context-aware applications.

Zimmermann et al. [28] extended Dey's definition of context by including a formal and an operational part. The authors stated that the formal part of the definition comprises five categories in contexts: individuality, activity, location, time, and relations. Whilst the operational part of the definition entails the dynamic properties of context and the exchange of context among entities. Chen et al. [33] describes context as:

> Information about a location, its environmental attributes (e.g., noise level, light intensity, temperature, and motion) and the people, devices, objects and software agents it contains. Context may also include system capabilities, services offered and sought, the activities and tasks in which people and computing entities are engaged, and their situational roles, beliefs, and intentions [33, p. 1-2].

Nemoto et al.[29] defines context as "a set of spatial-temporal elements related to the person or product. In addition, these spatial and/or temporal elements are called contextual elements" [29, p. 43]

Soultana et al.[7] explained the importance of context for the systems, like ADAS, where the users' context is dynamic. In order to ensure safety, ADAS and ADS needs to be context-sensitive [7]. The authors proposed a taxonomy for context based on the three main categories of context information:

- Driver context (driver behavior, emotion, heartbeat, facial expression, driver profile)
- Physical environment context (traffic conditions, obstacles, weather conditions)
- Car context (speed, alarm signal, state of the tires) [7, p. 2]

According to Soultana et al., ADAS must monitor the vehicle and its environment in real time, as well as sense, analyse, predict, and respond in accordance with the contextual information like vehicle's state, the driver's state, and the physical environment surrounding them.

According to Damak [34], vehicle systems' operational contexts are frequently broad, diverse, dynamic, and uncontrollable. The context includes a variety of things, e.g., traffic signals, traffic signs, road users, weather, and environmental elements.

Damak [34] argued that in order to analyse and define the operational context elements and their dynamic properties, the scenario-based approaches need to be extended. Damak proposed an approach for identifying and defining operational scenarios of the vehicular Cyber Physical System (CPS) based on the ontology of the operational context. The operational context ontology is structured in five levels based on the context elements: Use case, Environment, Road Infrastructure, Traffic Infrastructure, and Traffic Objects [34]. The levels represent different layers for identifying various scenarios and context elements. The scenarios are built upon with each layer.

In level 0 'Definition of use case level', the use case of the vehicle and its attributes are defined. The constraints on the scenarios, behaviours, and road users are introduced from the use cases. These limitations define the boundaries of what operational context items are allowed [34].

In the level 1 'Definition of Environmental level', the environmental conditions in which the vehicle is designed to function must be stated after the use case has been identified. These factors have an impact on the driving conditions and perception capability of the vehicle. For example, the environmental conditions like low standing sun and icy road affects the driving conditions and the sensor's perception [34].

The level 2 'Definition of road infrastructure level' defines the operational context elements and their attributes related to road infrastructure such as road structure, lane structure, road geometry et cetera [34].

In level 3 'Definition of traffic infrastructure level', the design team needs to define the scenarios related to the various traffic infrastructure elements like road markings and traffic signs, that would be perceived by the vehicle [34].

In level 4 'Definition of traffic objects', the design team defines various scenarios based on the traffic objects, which includes elements like traffic participants, maneuver, vehicle riders, and traffic properties [34].

2.5 Process

This section describes the different processes presented in the literature that are related with deriving contextual definition and requirements from use cases. The subsection describes about the challenges with ADS systems.

According to Colwell et al.[8], it is important to have traceability between the ODD and the functional requirements to restrict the ODD for degraded operation. The traceability can be done using both the top-down and bottom-up approach. In the top-down approach, the ODD is defined and then mapped to the functional requirements of the perception system. Whilst in the bottom-up approach, the functionality of the perception system is analysed first and then the ODD is determined [8].

In PEGASUS sub-project 2 [35], it was suggested that the development process need to be modified based on the level of automation and the ODD. The process must be robust and flexible, for applying the functionality in the context of a series development and adapting to the future development needs. The process should also include feedback loops and simulations on various levels [35].

The PEGASUS method [35] defines a six layer model for describing scenarios for ADS features. The layers are:

1. Road- level (road geometry, road infrastructure, etc.)

2. Traffic infrastructure (traffic signals, traffic signs, etc.)

3. Temporary modifications and events (road works, road construction, etc.)

4. Moving objects (other road users like vehicles, pedestrians, etc.)

5. Environmental conditions (light situation, road weather, etc.)

6. Digital information (V2X, digital data/map,etc.)[35]

The PEGASUS method involves a series of steps. The first step is the data processing of the given use case, existing data, and knowledge related to that use case. The outcome of the first step are the logical scenarios [35]. In the second step, the definition of requirements runs in parallel with the data processing step. The requirements for the automated driving function are defined using the knowledge and the use cases [35]. In the third step, the logical test cases with evaluation criteria are created using the requirements and the logical scenarios from the previous step [35]. The assessment of the highly automated driving function is carried out in the fourth step. In this step, simulation is executed with the logical situations from the database, which are then confirmed on the testing grounds. To evaluate the results of the test execution, they are compared to the pass and fail criteria. The results are utilised for risk assessment, which finally contributes to a safety statement [35]. The resulting evidence is then compared to the predefined safety arguments in the final step [35].

Burton[36] in his blog argued that for designing and assuring safety of ADS, the industry must collaborate to develop a common knowledge of the scenarios and safety requirements. Currently various approaches like scenario-based and rule-based approaches have been introduced in the automotive industry. Burton recommended that these approaches should be used as a foundation for developing standardised scenarios and requirements specifications for typical use cases. Burton proposes an approach for deriving safety requirements, which includes the following steps [36]:

1. Define a set of use cases: This step defines the functionalities together with ODD and the level of automation.

2. Identify relevant standards,legal requirements and social expectations: This step determines the appropriate performance goals and limitations or the prerequisites for the use cases.

3. Identify a list of possible scenarios for each use case: The scenarios describe the action, events, scenes and also include dynamic elements like weather, environmental factors et cetera.

4. Identify and evaluate hazards for each scenario: Identification and assessment of potential hazards that could arise from vehicle failure, malfunction, traffic situation, or interactions with the driver or other road users are performed in this step for each of the possible scenarios. Also, defining safety goals and criteria should be done in order to specify the performance level required in order to reach the safety level defined in steps 1 and 2 [36].

Burton further explained that the scenarios and the safety requirements derived through this method will serve as a base for the detailed system analysis to identify conditions that lead to safety goals violation as well as to determine countermeasures to prevent them [36].

The Netherlands Aerospace Center (NLR) [37] organised a workshop to transfer lessons from automation in aviation to the road transport system. In the aerospace domain, even though they have automated systems, the pilots are mostly responsible for decision making and planning. According to Roosien and van Birgelen [37], the highest level of aviation automation including drones is equivalent to the level 3 automation defined by the SAE. Furthermore, the aviation industry has an extensive certification procedure that includes significant testing before a new system can be put into service. Roosien and van Birgelen mentioned that the aviation industry has developed a design methodology “Total system”. This method serves as a foundation for the development of the operational design domain as well as for the assessment of the AV’s safety in its operational context. The design method “Total system” involves the following steps:

1. Set the objective of the operation

2. Define the scope and content of the technical system

3. Determine the required technical functions of the system

4. Describe the operational environment

5. Describe the operational concept

6. Describe how the system will be monitored and controlled

7. Define the transition at the beginning and endpoint of the operation [37, p. 16]

As part of the workshop conducted by NLR, a case study is conducted by applying this methodology to SAE level 4 automated vehicle and presented the results of identified elements. Roosien and van Birgelen further argued that there is no standardised way for the assessment of safety for automated vehicles and believe that this method will be a promising solution to fill that gap.[37]

2.5.1 Challenges with Automated Driving Systems

Yurtsever et al. [38] presents general high-level challenges of driving automation and unsolved problems. It was stated that less than ideal road conditions, and dealing with intemperate weather, is still a tricky and unsolved problem. Human-machine interaction has many problems present, and the topic is not researched enough. Centralised cloud-based information management has yet to be implemented due to the complex infrastructure required and the vehicle-to-vehicle communication is in the early stages.

There are a lot of challenges to overcome before introducing autonomous driving into society. Jackson, from the company Altran, [39] presents five challenges to overcome in regards to autonomous driving. The first challenge stated regards 'assurance of systems and software'. Jackson considers two aspects: safety and security. The safety of operation needs to be considered when developing automated systems. There are several factors that are being considered for today's vehicles on the market that should also be taken into consideration for automated systems to provide assurance of safety during operation. It is stated that safety can only be judged by examining dynamic interactions between components and effects outside the system boundary, which raises a concern about bounding responsibility. Another difficulty arises for characterising the environment for development or testing. Difficult ones to characterise could be temporary conditions e.g., snow on lane markings or redirection of traffic. The environment for automotive is also more prone to single-point failure i.e., one vehicle misbehaves, in comparison to other domains like railway and aviation. A contributing factor is that the road environment is less regulated than in those other domains. It is also stated that the mistakes for automated vehicles will be different then for humans, since they sense the environment differently. A concern raised, when a system operates instead of the driver, of how to handle and effectively have the user take over for situations where the system suddenly cannot function. Jackson expresses that to get these systems in to the market they need to be accepted and a decision must occur on a societal and political level. However, the responsibility still lies on other involved parties to be transparent about decision and arguments made.

The challenge 'sensing and connectivity', regards ensuring that the vehicle has the right relationship to the environment, so the vehicle can operate efficiently and safely. Direct sensing builds a model of the environment and may utilise wireless communication to get further environmental information. In systems, especially safety critical systems, there is an emphasis on ensuring that the communication between the system and sensors are dependable. They state that key technologies moving forward will be signal processing and analytics in support of sensor interpretation. Some examples of these technologies are vision processing and sensor fusion i.e., taking advantage of multiple input sources. These techniques, like machine learning, take a lot of computational power and are difficult to verify by traditional means. They state that to bring them into the market, and mass production, further advancements need to occur both in implementation and verification. Jackson states that, despite the challenges, the autonomous driving demands state-of-the-art technology that can handle the complex operational environment and the information required.

The third challenge regards 'Judgement'. This considers how automated systems manage expectations and make decisions. It is especially discussed how they will be able to make ethical judgements and who is responsible for decisions made by the automated system/function. Concerns listed regards three areas: decision-making, data and change, and human interaction. In regard to decision-making, it is stated that in domains like railway or aviation, the environment the systems are operating in is either limited or rigorously controlled by highly trained and monitored staff. These domains can have algorithms and measures used that are absolute, whilst the same cannot be used for the automotive domain. The road environment can be more complex and cannot be monitored to the same extent. Therefore, the algorithms adopted, and the measures used to assess them should be statistical, according to Jackson[39]. According to Jackson, autonomous vehicles will be more reliant on available data to ensure safe operation. This data is prone to change, because of the complex and changeable operating environment. Therefore, it is crucial to consider the quality, and maintenance of quality, of that data. Lastly, there is a concern about the human interaction with the autonomous systems and functions. Specifically, they bring up the tricky situation of when there is a need to pass control from the system to the user. Two main points are brought up to consider. First, the time needed to alert the user to take over and if the autonomous function can still operate safely during that period. The second point regards how to transfer the control to the user without causing the user to panic or overreact to the situation. Jackson states that user-centered approaches will be required moving forward, because the interaction between humans and machine will be of utmost importance for development and deployment into market.

The fourth challenge concerns architectures for managing complexity. It is important to understand the complexity of the autonomous control of a vehicle and the challenges that come with it. Autonomous control of a vehicle consists of a large number of cooperating functions e.g., an environmental perception system, a vehicle context system, and an active safety system. The complexity of such a system creates major challenges with e.g., cost and time of integration, testing, and ensuring consistent adequate performance and confidence throughout its life-cycle. The architectural methods used needs to keep this manageable and plausible.

The fifth, and final, challenge stated by Jackson [39] concerns verification and validation. The challenge regards how to decide and know what is sufficient amount of testing to ensure the level of required safety and confidence. Simulations and modelling can help make the testing more manageable, but according to Jackson a lot of dynamic testing will still be required. The challenge is to cover the complete complex environment that ADAS and autonomous systems operate in, to achieve the required level of confidence. Currently the process of labelling scenes and objects is manual, which causes it to be slow and expensive. It was stated also that it is hard to recognise the desired behaviour. The testing will further be complicated and increased, because of the higher presence of interacting functions within an ADAS or autonomous vehicle. For example, sensor fusion and centralised logic has algorithms that will demand testing and verification against known scenarios. Jackson states the need to have test approaches that can support massive and well characterised test programmes and that evidence should be gathered from a wide range of assurance methods.

The lack of a scientific approach to testing has resulted in the automotive industry's inability to meet the challenges regarding testing posed by ADAS and ADS [40]. Khastgir et al. [40] suggested two approaches for testing the ADAS and ADS and the generation of test scenarios: Requirement Based Testing (RBT) and Hazard Based Testing(HBT). Khastgir et al. suggested to focus on how the system fails while testing ADAS and ADS. The authors also mentioned that several fault analysis tools have been used in the automotive industry and the research community to identify hazards, but those tools are inadequate in analysing ADAS and ADS systems. Khastgir et al. found two main themes in their study that are related to challenges with the test methods. The themes were the definition of pass/fail criteria and the completeness and the consistency of requirements [40].

Zhang et al. [41] mentioned that the different stakeholders have different requirements and levels of abstraction. As a result, there is a lack of a common and understandable language for describing scenarios which in turn leads to misinterpretations. So, the authors proposed a two level Scenario Description Language (SDL), SDL level 1 and SDL level2 for describing scenarios for ADS. This enables describing and communicating scenarios among various stakeholders in a common language [41]. In SDL level 1, the scenarios are described in textual language at a higher abstraction level whilst SDL level 2 is a formalised description of scenario and mainly used for simulation and testing [41].

Thorn et al.[2] recommends using an iterative way or a combination of modelling and simulation, test track and on-road testing for assessing the performance of ADS. It was suggested that modelling and simulation should be primarily conducted since they can offer a substantial coverage of a broader range of test conditions in an efficient way. It is possible to test a wider variety of scenarios with the simulation. Moreover, simulation environments are faster to deploy and execute. In comparison to developing one or more fully functional test vehicles, simulation may be a less expensive choice for initial testing [2].

3

Research Method

This chapter introduces the applied research methodology in the study. We conducted a qualitative exploratory study. This chapter begins with rationale for the chosen qualitative research method, followed by a more detailed overview of the different phases and detailed methods applied during the course of the study.

3.1 Qualitative Research Method

The qualitative study is the process of understanding and exploring the problem by collecting data from individuals and analyse them inductively to create themes and making interpretation of the findings [42]. Creswell also suggests that qualitative study is useful when the researcher tries to investigate and explore a problem without knowing the variables of the problem.

The qualitative research method was evaluated to be a good fit for this study, because of the need for further exploration and a deeper understanding of the research topic, and to achieve study results in a realistic environment.

A qualitative study tends to focus on different perspectives of different people by incorporating the real world context [43]. It also attempts to broaden our understanding of participants' experience with the subject area, their views and opinions, and it explores the issues in the subject area that are not yet identified [43].

As stated by Creswell [42], the qualitative research process is emergent which means that the phases are not strictly followed in a sequential manner. Some of the phases might need to be revisited or altered when the researchers starts collecting the data. The process of our research study involves six phases that are represented in Figure 3.1: planning, preparation for data collection, data collection, data analysis, evaluation, and reporting of the study results.

A successful study necessitates proper planning and there are several elements that must be planned ahead of time. The preparation for the data collection phase involves activities like sampling strategy, ethical consideration, and preparing interview guides that are needed for data collection. An initial literature review was performed to review existing research related to the topic of the study. In qualitative

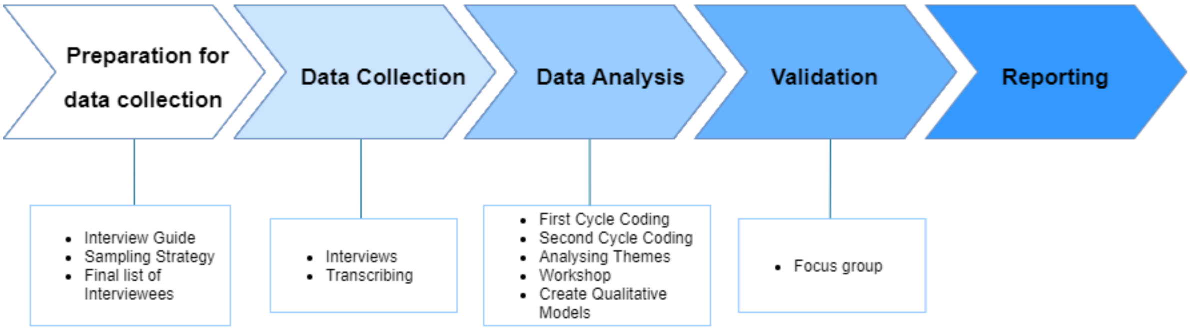

Figure 3.1: Phases in our study

study, the literature plays a less substantial role [44]. It served therefore more as a preparation for the data collection, by deepening the knowledge and broaden the understanding of the topic.

The data collection phase includes collecting and recording information from the participants through semi-structured interviews. Then the collected data is analysed inductively using thematic analysis to develop themes and the themes are interconnected to form a qualitative model. The validation phase will be carried out in order to know about the correctness and the credibility of the findings. After validating the findings, the results of the study are reported.

3.2 Preparation for Data Collection

This section describes and motivates the preparation for data collection of this qualitative study. The subsections will describe the process and material created for the interviews, the early literature review, and motivate the samples used in this study. The main documents created for the interviews were the consent form and the interview guide. The consent form ensured the participants of confidentiality and informed them about how data would be handled and stored. This was sent out before each interview. The interview guide created contained the script of the interview and more specifically consisted of important information that should be given to the interviewee and the questions.

3.2.1 Sampling

In qualitative studies, purposeful sampling is a common technique to use for sampling [18]. Participants for this study were purposefully chosen with the use of the maximum variation strategy [18]. To properly investigate and explore the topic it was important to interview experts, within autonomous driving, with different experience and perspective on the topic. According to Creswell [18], maximum variation is a good strategy to pick, because it increases the chances of collecting data that will reflect the different perspectives. The participants were chosen based on their role, company, and availability to participate.

To get samples that cover the different parts of the process, four different kinds of roles were identified. The four different kind of roles are as follows:

- People working on system/architecture level i.e., with an overview perspective. Examples of these are System Manager, System Engineer, System Architects, and System Designer
- Requirements Engineer
- People on the customer/user side. Examples of these are product owner and function owner.
- Developers, which will develop the system based on the specification

The case company was responsible for contacting and finding interviewees for this study, based on the sampling strategy provided by the researchers. However, the researchers and academic supervisors were also part of finding relevant participants for the study. The different kinds of roles requested were communicated to the case company, alongside with some additional restrictions. At least ten interviewees were requested for the study to get a good and sufficient amount of data that explores the topic. It was requested to have about half of the people interviewed from the case company and one from each level from the case company. This was requested in order to get a good idea of how the case company viewed the topic and also to be able to compare the other data towards that data. The rest of the interviewees was requested to be from various relevant automotive companies. Two interviewees should also not be from the same company and have the same role. These requests were however affected by the availability of the potential participants.

Marshall and Rossman [45] states that a researcher needs to be flexible when it comes to sampling, because it can change during the study. Since the samples were also given during the data collection phase, adjustments to samples were also made during the data collection phase to cover the spectrum of participants that were requested. A certain role was also stopped from being interviewed when the data collected was very similar or the same. The sampling consisted of seven participants from the case company, and the rest of the participants was from various OEMs with a similar profile as the case company. The final list of the participants, both from inside and outside the case company, are shown in Table 3.1.

3.3 Data Collection

This section discusses the data collection procedures and methods in the qualitative study. The subsections will describe the method for the interviews that has been used, how the interviews were conducted, and how the data was transcribed and organised for the next phase of the the study.

Creswell [44] describes five steps in data collection. The first four steps describe how to prepare for the data collection. The first step is to decide the sampling strategy,

Name	Role	Experience
Interviewee A	Group Manager	ADAS features and collision avoidance features
Interviewee B	Functional Developer	ADAS features
Interviewee C	System Engineer	Planning and control for safety critical issues
Interviewee D	Researcher	Research and innovation group for sensors and systems
Interviewee E	Developer	Develop Algorithms for obstacle detection, blind spot
Interviewee F	Functional Developer and Functional Safety Engineer	ADAS features
Interviewee G	Project Manager (European Transport Administration)	ADAS vehicles
Interviewee H	Researcher	Automotive, Energy-management, Data management, Computer vision
Interviewee I	Product Owner	Ground truth system
Interviewee J	Technical Lead	Artificial and Machine learning projects
Interviewee K	Technical Specialist	Sensor technologies for blind spot detection, lane changing, adaptive cruise control, and forward collision avoidance
Interviewee L	Functional Safety Manager	Functional safety methods
Interviewee M	Researcher	Standardization of AD safety assurance methodologies

Table 3.1: Participants of the interviews

that is deciding on what type of people will be needed for the study. In the second step, the access to site and permission for interviewing people is obtained. The third step is to decide which data collection methods to use. The methods need to be chosen based on the type of data that is needed to answer the research questions. The fourth step is designing the research instruments, such as interview protocols and procedures for collecting and recording data. The final step, is to conduct the data collection process in combination with ethical consideration. In this section, the last step in the process will be discussed further. Step one to four however are described in the previous Section 3.2.

There are various data collection procedures in qualitative study and the appropriate methods are chosen based on the data that is expected to address the research questions [46]. Creswell [42] presents four basic types of data collection procedures in qualitative research: qualitative interviews, qualitative observation, qualitative documents, and qualitative audio visual materials. Lethbridge [47] describes three categories of data collection methods. The first degree methods will collect data directly with the subjects by having direct interaction in real time (e.g., through interviews). In second degree methods, the data is collected without having direct contact with the subjects (e.g., by observing through video recordings). In third degree methods, the data is collected through analysis of artifacts and documents.

In this study, a first degree method namely interview was used as the main source for collecting data from the participants. This method allowed the researcher to have control over the type and quality of data collected, and the data collection process itself.

3.3.1 Interviews

Brinkmann and Kvale describe qualitative interviews as "attempts to understand the world from subjects' point of view" [48, p. 3]. Interviews are interactions in which the researcher asks the participants a set of questions about the central phenomenon of the study. The direct interaction with the participants allows the researcher to control the data collection by formulating the interview questions based on the central phenomenon of the study [46]. Robson and McCartan [49] differentiate between unstructured, semi-structured, and fully structured interviews. In this study, the interviews were semi-structured. In semi-structured interviews, the researchers have planned and prepared the interview guide with a set of questions beforehand. But still, the order of questions can be altered during the conversation with the participants. The researcher can also ad hoc include additional questions to allow for deeper understanding and exploration of the topic of the study [46].

The main objective of the interviews in this study is to get an understanding of the process of deriving contextual definition and requirements from use cases and investigating about the parts that are currently working well and challenges with the process. The participants for the interviews were selected within the automotive domain from the case company locating at different sites and also outside the case company. Thirteen individual interviews were performed with the experts from the automotive domain. Before each interview, the participants received a consent form with the information on how we collect and handle data with anonymity and confidentiality, along with a short description about the thesis. The interviews were conducted remotely using Zoom[1] and Microsoft Teams[2] because of restrictions due to an ongoing medical pandemic. Each interview session took about one hour of time. This study has been carried out by the two authors of this thesis, who alternated between being the interviewer and observer of the interview. At the start of each interview, the interviewer presented some background details and outlined the study's objective, as well as received consent for recording the interview.

The interview guide's questions were formulated based on the research questions, with the intent to find answers to the research questions. The interview guide was divided into three different sections, each with a set of questions. The first section consisted of questions aimed at learning about the participants' current role and experiences. The second section focused on establishing some ground concepts with the participants, in order to prevent any misunderstanding about the concepts. This section included showing a description of ODD to get the participants' view on ODD and to find out if the description is missing any vital information. For some questions examples were given to assist participants in answering the question. The third section explored the process of deriving contextual definition and

[1] An online communication and meeting tool , `https://zoom.us/`

[2] An online communication and meeting tool , `https://www.microsoft.com/en-ww/microsoft-teams/group-chat-software`

requirements from use cases. It aimed at investigating the multiple facets of the process, including finding out what the main concerns are, what works well, and how it can be improved. Based on the flow and direction of the interview, the order of the questions were altered and some additional follow-up questions were included during the interview. At the end of each interview, the participants were informed that potential follow-up questions could be sent by email. They were also made aware of the focus group, that would be carried out as part of the validation phase.

Most of the interviews were recorded and field notes were taken by the observer. For those interviews that were not recorded, more detailed and comprehensive notes were taken by the observer. Each recorded interview was manually transcribed into a text file. The VLC Media player was used to watch and listen to the recording while manually writing down each word. Timestamps and unclear words were specifically marked in the transcripts. For two interviews, Zoom's[3] automatic tool for transcription was utilised, however the researchers did still manually listen through the interviews to ensure no wrong transcription was made. The transcripts were anonymous and were created to be used for the analysis. The transcripts made us familiar with the data and also helped in refining the interview questions for the forthcoming participants to increase the quality of the data. The interview questions were modified to a lesser degree.

3.4 Data Analysis

This section describes and motivates the data analysis of this qualitative study. In this study, the transcribed and organised data was analysed by the coding practices suggested by Saldaña [50]. The data analysis process consists of four phases, as shown in Figure 3.2.

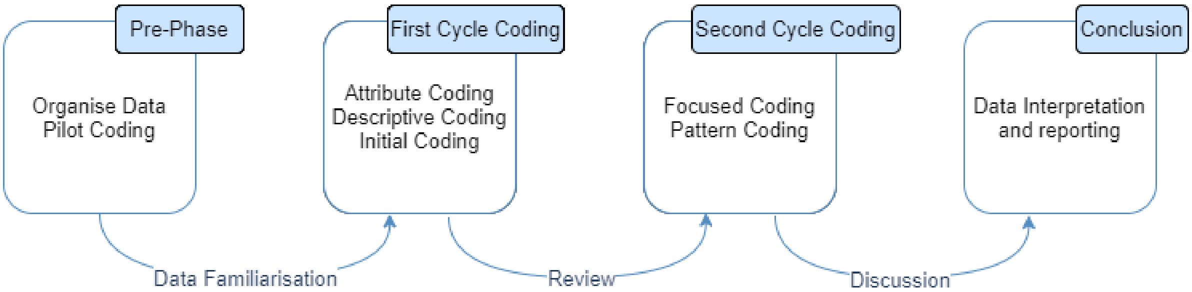

Figure 3.2: Method for conducting Data Analysis

3.4.1 Pre-Phase

In the Pre-Phase, pilot coding was performed, as suggested by Saldaña [50], with some data to get a feel for which coding methods would be suitable. Saldaña suggests that novice coders start with "generic" coding methods, like Attribute Coding and Descriptive Coding. The intention is to try if the coding methods give sufficient

[3]An online communication and meeting tool , `https://zoom.us/`

discoveries. Considering Saldaña's advice, and which methods were compatible with each other, it was decided to test the coding methods Attribute-, Descriptive-, and In Vivo Coding. The statements of the transcript were divided beforehand, and then both researchers of this thesis coded separately. The codes were then compared, evaluated, and a final decision on the codes for each statement was made. As a result, from the pilot coding, In Vivo Coding was replaced by Initial coding for the first cycle. Initial Coding creates the opportunity to code more freely, whilst still being able to employ In Vivo Coding.

3.4.2 First Cycle Coding

In the first cycle, three coding methods have been used: Attribute Coding, Descriptive Coding, and Initial Coding. The Attribute Coding method was used to note down the meta information such as interviewees' role, interviewees' experience, and date of the interview. These codes created a reference for tracking data and helped to differentiate between the interviews. The Descriptive Coding was used to develop some codes which represents the summarised topic of the statements. The Initial Coding has been used to a great extent in the first cycle. It involves an open ended approach and is a combination of In Vivo and Process Coding [50]. Some codes were generated using the In Vivo method, in which the terms or phrases of text expressed by the interviewees were quoted as codes. These codes helped to highlight the interviewees thoughts. Therefore, the codes rely on the interviewees for giving meaning to the data.

3.4.3 Second Cycle Coding

Saldaña expresses the primary goal of the second cycle coding to be "to develop a sense of categorical, thematic, conceptual, and/or theoretical organisation from your array of First Cycle codes" [50, p. 207]. The aim for this cycle was therefore to further organise and analyse the initial and descriptive codes, that was the outcome of the first cycle coding. By further working with the codes, the codes could be improved upon, combined if essentially the same, or even removed if deemed out of scope or redundant [51].

In this cycle, two coding methods was used: Focused Coding and Pattern Coding. Both of these coding methods are appropriate for second cycle coding, after performing Initial coding [50]. Focused Coding organises the data into major categories that are identified as the most salient [50]. Notably, the coding method do not pay attention to codes' properties or dimensions. Pattern Coding is used to identify emergent and meaningful themes amongst codes [50]. Both of the chosen coding methods bring different helpful ways of organising and analysing the codes' from the first cycle coding. Focused Coding was chosen to organise the data into categories and to help make working with the codes manageable. The Pattern Coding was, however, chosen to bring forth themes with meaning that could be used to create models for the final result.

The steps for the second cycle can be seen in Figure 3.3.

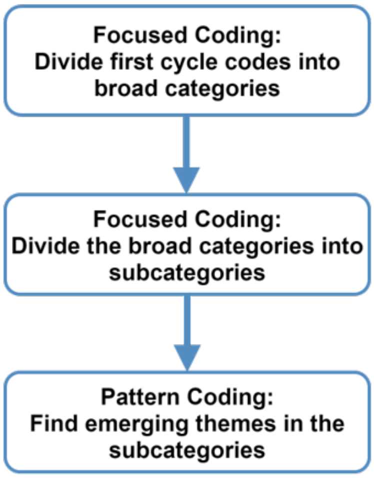

Figure 3.3: The steps of the second cycle coding

The second cycle coding started with Focused Coding, were the Initial Codes was divided into broad categories. An overview of the broad categories can be found in Appendix A.2.

In the next step those broad categories were divided into subcategories, see Appendix A.2, since the broad categories were not specific enough about the content. The subcategories were created before assigning them to the statements. The categories were named into what the researchers thought to be good subcategories.

Assigning the subcategories to the statements was done by going through all the broad categories one by one. The filter function in Excel was used to filter out all but one category. Each statement was then assigned to one of the existing subcategories. The statements where it was unclear which subcategory it belonged to was marked as 'unsure'. Later, these statements were revisited and discussed further in order to assign them to a subcategory that fit well.

3.4.4 Workshop

After the themes were created in the second cycle coding, a workshop was organised to review the themes to discuss possible bias. The workshop consisted of four tasks that would be accomplished in three hours. The schedule for the workshop is shown in Figure 3.4.

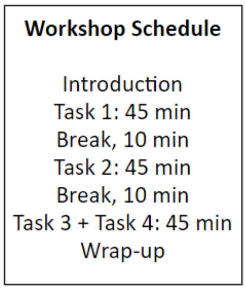

Figure 3.4: Workshop Schedule

To accomplish the tasks the tool Miro[4] was used, which is a online collaborative whiteboard platform. The themes from the second cycle coding were organised in the Miro board as separate note cards. The supervisors of this thesis, partners from the company supporting this thesis and from academia,

[4]An online collaborative whiteboard tool, `www.miro.com`

took part in this workshop. The researchers guided them through the workshop and provided further information and clarification about certain themes, when needed.

As seen in Figure 3.5, the first task was to get input regarding certain themes that was found in regards to the current process. For some of the themes they were specifically asked to give input about if the theme was out of scope of the study. The second task, see Figure 3.6, was a bit more interactive and utilised the

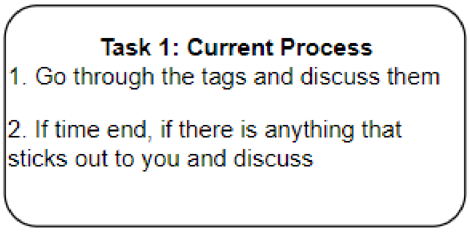

Figure 3.5: Task 1 for the workshop

functionalities of the Miro board more. The participants were asked to find more general themes amongst the challenges. The participants were presented with all the challenges gathered from the interviews, each one on individual notes, which enabled the participants to move each individual one freely around on the Miro board.

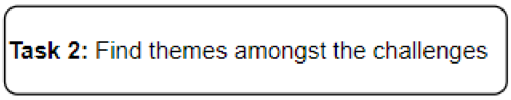

Figure 3.6: Task 2 for the workshop

In the third task, see Figure 3.7, they were asked to prioritise the themes they found in task two. The goal of the fourth task was to connect refinements and solutions to specific themes of challenges created in task two. The refinements and solutions presented for them to connect to the themes were all collected in the interviews.

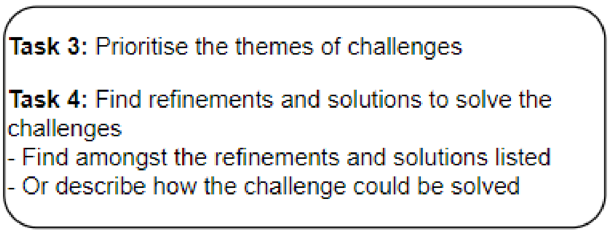

Figure 3.7: Task 3 and Task 4 for the workshop

3.5 Validation

This section discusses the validation procedure for determining the accuracy of the findings in our qualitative study.

Creswell and Poth [18, p. 406] states that the "validation process in qualitative research entails assessing the accuracy of the results as best described by the researcher and the participants". Creswell [42] discusses various strategies such as triangulation, member checking, peer debriefing for assessing the accuracy of the results. This study uses the "member checking" validation strategy. This strategy entails that the accuracy of the results are assessed by presenting the major results or themes to the interview participants, asking them if they agree on the accuracy of the results and receiving feedback from them. The approach can also involve follow-up interviews with the participants in order to provide the opportunity for the participants to reflect on the results [42]. In this study, a focus group is used as a validation strategy instead of individual follow-up interviews due to the time and availability restrictions.

3.5.1 Focus Group

The focus group was organised to elicit shared understanding from a group of people, along with specific viewpoints from individuals [44]. The four participants that took part in the focus group were interviewees from the data collection phase. One of the researchers led the focus group discussion as a moderator whilst the other one took field notes. The focus group was conducted remotely using the Mentimeter [5] tool. This tool made the session more interactive and served as an easy way for the participants to answer directly. The guidelines and the questions for the focus group were created by the researchers beforehand. The questions were organised based on the research questions and the themes that were created during the workshop subsection 3.4.4.

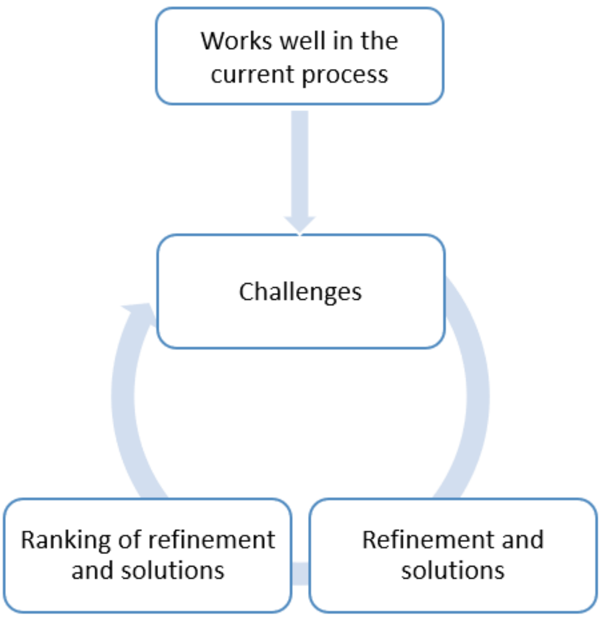

Figure 3.8: Focus Group Discussion work flow

[5] An interactive online survey tool, `www.mentimeter.com`

The focus group was given four different tasks, as seen in Figure 3.8. The first task aimed at assessing whether the participants agreed with the themes for what was working well in the current process. In the second task, the themes related to challenges were presented and the participants were asked to discuss whether they agree with the challenges or not. The third task aimed at assessing the themes related to refinement and solutions. The participants were asked if they feel that the refinement and solutions would solve the challenges discussed in the previous task. In the fourth task, the participants were asked to rank the themes related to refinement and solutions. The answers of the focus group, together with the discussions were used to validate and improve the results of this study.

4

Results

In this chapter, the results for the three research questions stated in Chapter 1 are reported in detail. The chapter begins with the results for the concepts and the current processes, which is presented in Section4.1. Following that, the results for working well in the current process (RQ1) are presented in Section 4.3. Then the results for the challenges (RQ2) and the validation results from the focus group are presented in Section 4.4. Finally, the chapter ends with the results for the improvements (RQ3) followed by validation results of it, which are presented in Section 4.6.

4.1 Concepts & Current Process

This section will present the result in regards to each individual concept and the current process. The concepts that result are presented for are Requirements, ODD, Context, and Use Case. The result for the current process will discusses the different processes currently in use in the industry and research field for deriving contextual definition and requirements from use cases.

4.1.1 Requirements

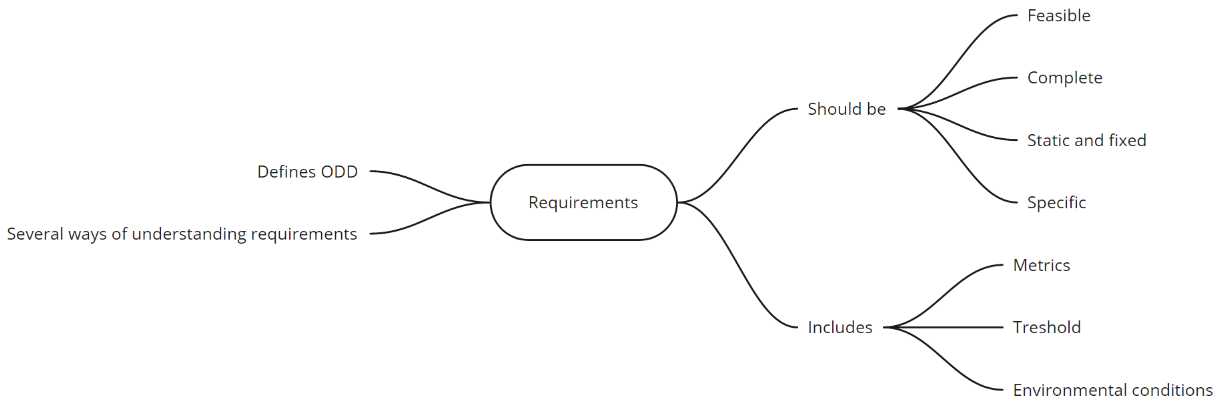

Figure 4.1: Themes identified for requirements

The mind map in Figure 4.1 shows themes collected on how requirements should be defined, what kind of information it should include, and additional themes identified about requirements. Requirements can be interpreted in several ways, therefore it is important to clarify what is meant by requirements, as well as to state what the goals of having requirements are. Requirements can be formulated in different ways, depending on the context for which they are written for.

It was stated that a specification, i.e. requirements, needs to be complete and feasible. Requirements was deemed not useful, if they were unfeasible. It was brought up that it is especially important to be able to argue for completeness of safety critical requirements and to be thorough when creating them.

Today, requirements are viewed as being static and quite fixed. There are, however, plans to have requirements be handled in a more agile way, meaning that some of them would be able to be changed during development. The importance of traceability was also lifted with the statement of having the requirements being less static.

The need for the requirements to be specific is motivated by a need to clearly know what needs to be fulfilled. The consequences of requirements not being specific enough will be that it leaves room for interpretation on how the requirement are to be fulfilled or that they might not be fulfilled in the intended way. Functional requirements are specified to contain the thresholds(e.g., 80% accuracy), metrics(e.g., accuracy), and then also environmental conditions(e.g., rain).

Other thoughts in regards to requirements are listed on the left side of the mind map in Figure 4.1. The theme 'Defines ODD' stems from the initial code 'ODD like conditions are usually defined in requirements'. It was described that a lot of the environment or given conditions would be inside the requirements and based on the environment, the metrics e.g., range accuracy for a sensor, will adjust according to that. For example, the accuracy for a sensor will be different if the distance to a vehicle is fifty meters in comparison to ten meters. 'Several ways of understanding requirements' touches on that there are a lot of different kinds of requirements e.g., quality, functional, legal, et cetera. In this study, the main focus is on functional requirements.

	Responsible for deriving requirements	**Responsible for maintaining requirements**
System Engineer	X	X
Design Lead		X
Product Owner	X	
Function Owner	X	X
Feature owner in collaboration with team	X	
Relevant stakeholders e.g., component owner and hardware owner	X	
Quality Department	X	
Sensor Owners, System Engineers, Developers in collaboration	X	

Table 4.1: Responsible for deriving and maintaining requirements from use cases

A few different roles were identified to be responsible for deriving the requirements from the use cases. The roles that were mentioned during the interviews, when asked the question 'Who is responsible for deriving requirements?', can be seen in Table 4.1.

It was mentioned that system engineers and the quality department should be responsible for deriving the requirements, as seen in the following statements collected from the interviews:

> *"System engineers will definitely be the best ones to do that."*

> *"[...] the requirements engineering is done in the quality department[...]"*

Further, several interviewees emphasized the collaboration aspect of the process. The product owner was mentioned to be responsible for deriving requirements together with all relevant stakeholders.

> *"[...] all the relevant stakeholders are in the loop together with the product owner,[...]"*

Relevant stakeholders are for example component owners, hardware owners, customers etc. All the relevant stakeholders are involved in deriving the requirements. They negotiate what requirements are needed and can be defined. The negotiation was mentioned to be trickier, because there are several different stakeholders involved. The consequences of the negotiation not being done well is expressed in the following statement.

> *"[...] if those are not negotiated well, you might end up, even with contradictory requirements."*

Additionally, it was stated that experts should be involved in the process, with emphasis on the initial phase of the development process.

> *"[...] you have to have a collaboration between a main team or main person, that is mainly responsible for the function, and the set of experts that you can get, especially during first initial development of your use case and your first requirement set that you can get as experts into your team, maybe not for the entire development process but especially in the initial phase."*

The function owner defines requirements together with people who are in contact with the customers. Here follows some statements that are motivating why the function owner should be responsible for deriving the requirements.

> *"[...] function owner of course, should be involved because he or she has the best overview of the function."*

> *"[...] he or she is responsible to break it down into requirements[...]"*

In the interviews, information was also collected in regards to which role was responsible for maintaining the requirements. It was mentioned that if the requirements are static and fixed, they do not have to be maintained. This is because they would not change during the development.

Two interviewees brought up that the one who define the requirements should also be the one maintaining it. Another interviewee opposed that statement. It was stated that the people who define it do not necessarily have to be the ones defining it.

If the requirements can change during development, i.e. not being static and fixed, then it was stated that someone needs to keep track of the changes. It is also important to keep track that the changes made are acceptable. The person responsible for checking the requirements needs to ensure that quality concerns are still satisfied, for example, that the use case is still fulfilled, safety is still guaranteed, and that the security of the function is still guaranteed.

Three different roles were brought up to be responsible for maintaining requirements during development, as shown in Table 4.1. The system engineer and design leads needs to be informed about changes and approve those, as motivated by the following statement:

> *"[...] if there is any change they need to know, whether it will work for the system or not."*

Function owners were also mentioned to be responsible for maintaining the requirements. However, there might be more people involved in maintaining them, but the responsibility would lie with the function owner.

4.1.2 ODD

The ODD is an important factor in autonomous drive that defines where the autonomous vehicles can operate safely. However, there exists a challenge in that different people have varied thoughts and perspectives about what the ODD is. Therefore, the interviewees were asked to share their thoughts and opinions regarding the ODD definition. The themes identified in relation to the ODD definition are depicted in the mind map in Figure 4.2.

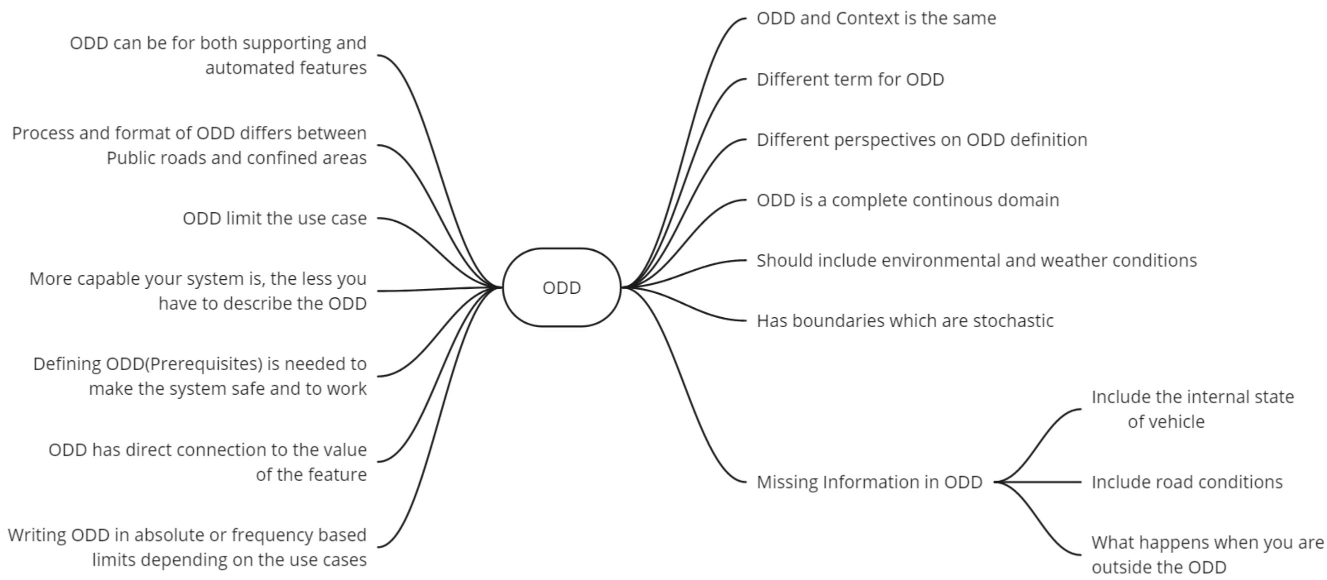

Figure 4.2: Themes identified for ODD

Most of the interviewees agreed with the J3016 definition for the ODD, which is:

> Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics [1, p. 14].

Given the J3016 definition the interviewees reflected on different things that an ODD definition should include. It was expressed that the ODD should include the internal state of the vehicle, capability of sensors, and driver behaviour. The internal state could be e.g., the status of the actuators. They stated that it should also include the road and lane conditions such as lane markings and traffic signs.

The interviewees mentioned that the important elements to be included in the ODD definition are the environmental and the weather conditions as these elements are dynamic and unpredictable.

The ODD definition were declared to be incomplete in terms of describing what happens when the system is not in compliance with the ODD. This was reflected in the interviews by the following statements.

> *"[...] what I'm missing [...] what happens when you allow or beyond the border and that has to be included in the ODD"*

> *"[...] in the ODD you have to describe it as sort of a graceful degradation of the system when you go outside and it's never included"*

It was revealed from the interviews that there exists inconsistent terms that are being used in lieu of the ODD. For example, one of the interviewee mentioned a more broader term called 'scope', which was used instead of ODD. It also emerged, that the term ODD tend to be interpreted in different ways. For example, the different perspectives were mentioned in the following statements.

> *"[...] where the system can operate and it is common to refer to as ODD"*

> *"[...] describe in what context the system should remain productive"*

> *"[...] what conditions is the system safe"*

The interviewees stated that the ODD is defined as a set of prerequisites that are required to fulfill the use cases. Furthermore, the ODD is seen as a design document or specification that describes the conditions within which the autonomous driving function can operate safely. The ODD is also expressed to have boundaries which are stochastic.

It was also mentioned that the ODD is evolving, and that it is a continuous and dynamic domain. For example, the ODD varies depending on various factors such as the time of day, traffic density.

The process for deriving ODD will differ depending on the use cases i.e., what functionality are being developed. The format of the ODD will also differ, due to the intent of the use cases being different. For example, the ODD for public roads and confined areas are different.

The interviewees, who are working closely with the standardisation organisation, described that the use cases are getting restricted by the ODD. The ODD limits the context which in turn also limits the functionality that the use cases are describing.

During the interviews of this study, it was found that most of the interviewees were uncertain about the roles and responsibilities for deriving ODD. There were a wide range of responses from the interviewees regarding who is responsible for deriving ODD. The reason was due to the lack of standard and common process for deriving ODD. The roles and responsibilities differs depending on different companies because companies have very different organisational structures. The responses included quite a wide range of different roles as seen in Table 4.2

Responsible for deriving ODD
System engineer/system team
Safety engineer/safety team
Vehicle Manufacturer
Business architect, solution architect, and enterprise architect
Supplier
Standardisation organisation
Teams for different components
Team of people from different disciplines

Table 4.2: Responsible for deriving ODD

4.1.3 Context

The interviewees were asked to describe context, elaborate on what the context should include, and encouraged to share other thoughts and opinions about the concept. The themes identified from the interviews can be seen in Figure 4.3. The outlined boxes around 'Three aspects of context' and 'Context relating to levels of automation' indicates that there are an extension attached to it, see Figure 4.4 and 4.5. For the full view, where everything is connected, see Figure A.3 in the Appendix

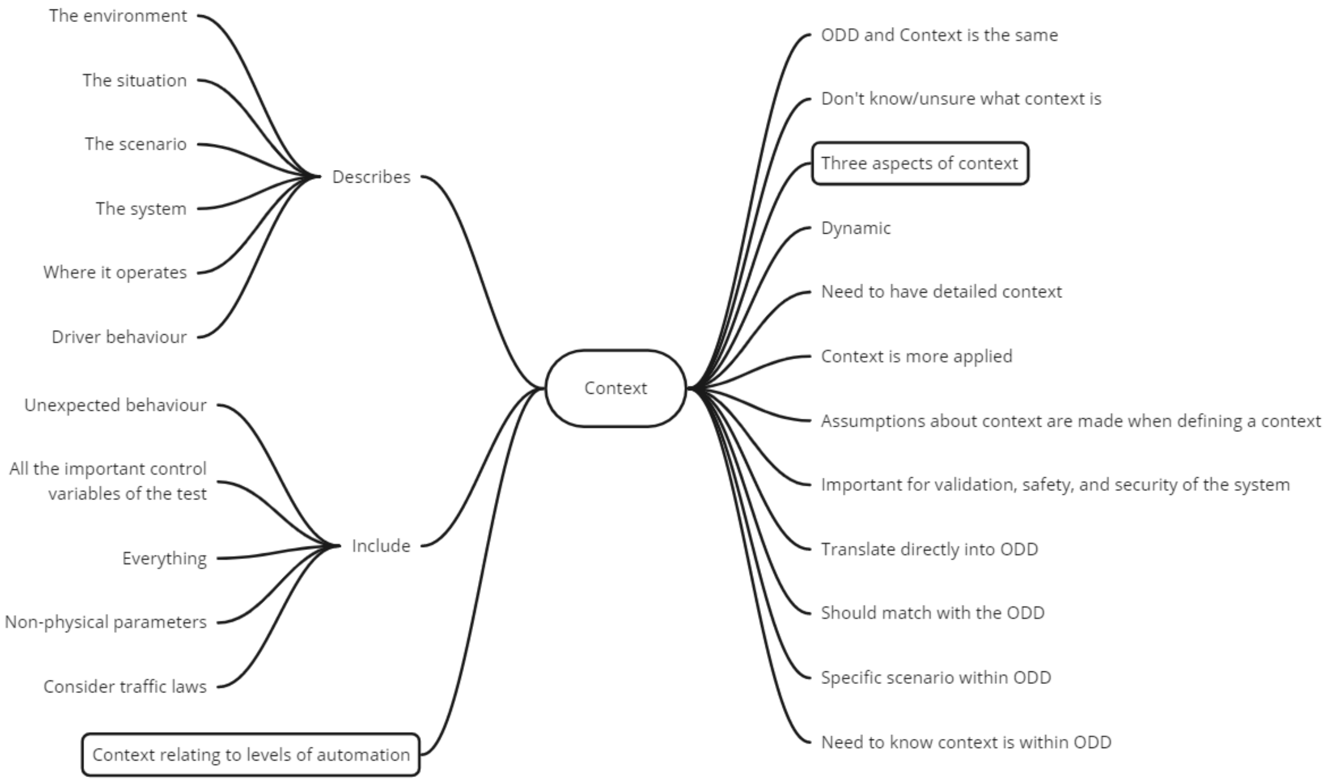

Figure 4.3: Themes identified for context

First, the interviewees were asked if they are familiar with the concept of context. The answer varied. Four interviewees were unfamiliar with context or unsure of the meaning behind it. The rest were familiar with the term context. One from each group, familiar and unfamiliar with the term context, considered context to be the same as ODD. They described the ODD and context as environmental conditions in which the system is designed to operate.

Some interviewees stated that the context is not the same as ODD. It was described that context feels a bit more applied, whilst ODD is something that is written on paper. The context was further explained with the following statements.

> *"The reality that the system operates within."*

> *"[...] it's kind of the same things as the ODD is describing but context contains all the possible combinations, if you like, of where this is going to operate."*

The interviewees expressed that the context should match with, and be within, the ODD. When asked about how to move from context to ODD, it was described that parts of the context are selected to be translated directly into the ODD. One interviewee described the context to be a specific scenario within the ODD. The context was further described as discrete points in the ODD. They view it as one point in time and in one condition, whilst the ODD would be the sum of all those contexts.

> *"...it's almost like to say it's discrete points in the ODD if you want this really specific conditions that you define around a use case..."*

Amongst the themes about how context should be described, it is identified that context is dynamic and should include the situation, scenario, or the environment. The following statements support the identified themes:

> *"But for me context is a bit more wide and not that specific but it is describing the whole situation and the whole environment as a whole."*

> *"But it is a way to define a situation, or define a system, or defining... yeah a scenario, or an environment, I would say. So that is what I would call a context."*

> *"I mean the context would be the definition of you know the environments and the environment could be of course environmental factors ah.. lighting, rain and whatever."*

> *"Immediate surrounding, where you are right now"*

These statements describe context as a wide concept that includes environmental factors and describes the surrounding. It was also stated that the context should describe the setup of the overall scenario of function of the vehicle. To clarify, an example scenario was given. The scenario included a vehicle, or system, on a highway with dense traffic i.e., other cars in close proximity. The scenario then should include a description of the road (highway), traffic (dense traffic), but it should also describe how the system should function and what the driver is doing. There were more things that should be considered and could be included in the context like road surface conditions, speed limit, what is known about the road, availability of map data, and vehicle dynamics like how hard can the car brake or steer.

The context was presented to be divided into three aspects: the user, the actor, and the system, as seen in Figure 4.4.

The context of the user would be the user experience inside the vehicle. For example, as a user inside an autonomous vehicle and using the system, the context would be how the vehicle is driving, how safe it is, how much trust the user should have, and what kind of experience the user should exhibit. The context of the user is to reach from point A to point B safely with good experience e.g., the user does

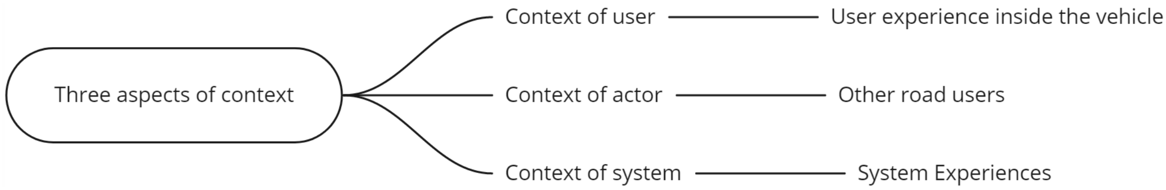

Figure 4.4: Three aspects of context: User, Actor, and System

not have to worry about a lot of things like unintended stopping, breaking down of the vehicle, and so forth. The context of the actor is the one who is directly or indirectly influenced by the autonomous system. Actors could for example be pedestrians, other road users, people on the sidewalk, and other authorities. The system context is defining how the system should function. The system will be interacting with the road and going to different urban areas, it must consider and take care of energy, wear and tear, and so on. The context is for example how it should drive in the roundabout, who should do what, who should go first, should the user overtake, and so on. These things should not influence the user experience.

Some interviewees mentioned that the system and the levels of the system should be described in the context.

> *"[...] the different levels of system that we are talking about [...]"*

However, the system and the user needs to know in which context it is operating in. The complete necessary knowledge about the context is important for validation, safety, and security of the system. Because of the importance of this, the context needs to be properly defined. Additionally, they also expressed that the context needs to be defined with enough details in order to be able to repeat or reproduce the results. However, it was expressed that context might not go into the in-depth details. There always will be assumptions when defining a context. For example, if the vehicle is driving on the motorway or highway, then it is assumed that there will not be people crossing the road or bikes suddenly being between vehicles. This is assumed because, by law, pedestrians and bikes are not allowed on the motorway. It was, however, expressed that there is still a need to develop safety functions that can see and do an emergency brake. The probability of those scenarios occurring in the context of a motorway is much lower than for the context of city traffic, where pedestrians and bikes must be expected at all times.

The interviewees also presented information about what the context should include. Some wanted the context to include and describe non-physical parameters such as the driver's behavior. The following statements describes what should be included in the context, there amongst the behavior or the state of the driver.

> *"That means also that you have to describe the fact that the driver might be more stressed in the morning, maybe, something like that, but ok so it's non-physical parameters that should be included in context [...]"*

> *"Another includes also things that are not physically related, I should say. So here it actually includes behavior of driver and people outside of the car, and even animals which is never a part of the ODD."*

Context should, also, include unexpected behavior, in order to be closer to describing the real world. The unexpected behavior is not included in the context today, because it is complicated to include. It is motivated by the following statement:

> *"[. . .] normally, we don't do that because it's so complicated when you start thinking about these things, but it actually need to be there."*

It was explicitly mentioned in one of the interviews that context should include everything. This was further clarified by explaining that it should include everything that will affect, impact, or vary over the use case. There are, however, some limitations where the systems should work. For example, if the system should only work on roads, this limitation should be stated in the context.

> *"So, the context should say this will only work on roads, that's all I have to care about but what can happen on the road."*

There is still a lot to consider on the road like people, animals, bicyclists, etc. More things that are needed to be considered for the context were traffic laws, different weather, and whether it is night or day because the road will be different.

The researchers investigated if the interviewees thought context was limited by the levels of automation. The answers varied, as shown in Figure 4.5. Four out of

Figure 4.5: Themes identified in regards to context relation to levels of automation

twelve agreed, with some uncertainty, that the context is limited by the level of automation. The uncertainty was from the fact that the interviewees felt that it was quite difficult to be sure, and they expressed that the question was a new one that needed some further reflection. However, they did express some motivation to why they agreed. The context was described to influence the vehicle and that it grows, i.e., becomes wider, with the levels of automation. For example, level two automation, according to the SAE definition of automation, needs clear definitions of the context in which the function can be active (e.g., only motorway, with clear visibility, driving more than 50 kilometers per hours, etc.), whilst for an automated vehicle level four there is less need of describing every detail because the context should be so much wider. However, there is a need for transition zones in level four, since it would be unacceptable for an autonomous vehicle to suddenly want the driver to take over because they are out of their context. In those zones the driver can be prepared to take over. There are other expectations on a level two function, for those it should be no issue that the function suddenly wants the driver to take over.

One interviewee did not want to describe the context as being limited by the levels of automation, but rather that the context is different for the levels of automation. On the contrary, two out of twelve thought the context is not limited by the levels of automation and two out of twelve thought that the level of automation is limited by the context. It was described that one must consider the level of automation based on the context. The rest, four interviewees, were not asked this question, due to time restrictions.

The prominent theme from trying to identify who is responsible for deriving the context was that it is unclear or interviewees are unsure who should be responsible. However, some suggestions of roles were discussed in the interviews, see Table 4.3. It was discussed by an interviewee that the supplier had partly split responsibility

	Responsible for deriving context	**Responsible for maintaining context**
Supplier	X	
Customer	X	
Relevant stakeholders with Product owner	X	
Quality department	X	X
The ones who defines the product functionality	X	
System Engineer/System Team and Design Leads	X	
Function owner	X	

Table 4.3: Responsible for deriving and maintaining the context

with the customer. The supplier would be responsible for ensuring that the prerequisites are fulfilled, whilst the customer would be more responsible for analysing and defining the use case and context. The customer was also mentioned by others as being responsible for defining the context together with others. Their motivation was that since the customer is the one who orders the functionality, then there is a big responsibility on them to ensure that they have the proper input, the proper context, and the prerequisites. It was described that the customer together with relevant stakeholders and the product owner will be in the loop of defining the context and requirements. Another expressed it as the ones who define the product functionality should be responsible.

However, one person specifically oppose that the customer should be responsible. The interviewee stated that the customer needs the use case, and will provide the company with the use case, but not the context. With some uncertainty, it was suggested that the responsibility would fall on the quality department, since the requirement engineering is done today in that department. It was more specifically motivated by the fact that they develop the method for requirements handling. It was also mentioned that they might be suitable to be responsible for maintaining the context.

System engineering and design leads were also mentioned to be responsible for defining the context, in collaboration with the customer, because they are architecting the whole system. They were described to have the best idea of where the sensor and the output will fit in the system. However, another interviewee explained that the system team or the team who design the system should use the context and design the system, but not be responsible for defining the context.

The function owner was described to be the one responsible for deriving the context. They are described as suitable, because they have the best overview of the function. However, they should be collaborating with a team when deriving the context, in order to further ensure that they derive the proper context for the system. The team should consist of experts, like functional safety experts or security experts. The additional expertise was mentioned to be especially important in the initial phase, when the first set of requirements and the context is derived.

4.1.4 Use Case

In order to investigate the process of deriving contextual definition and requirements from use cases, there is a need to understand and explore how the use cases are defined for Automated Driving Systems (ADS), and what is the most important element to define in the use cases. The mind map in Figure A.3 shows the themes identified with regards to the use case definition.

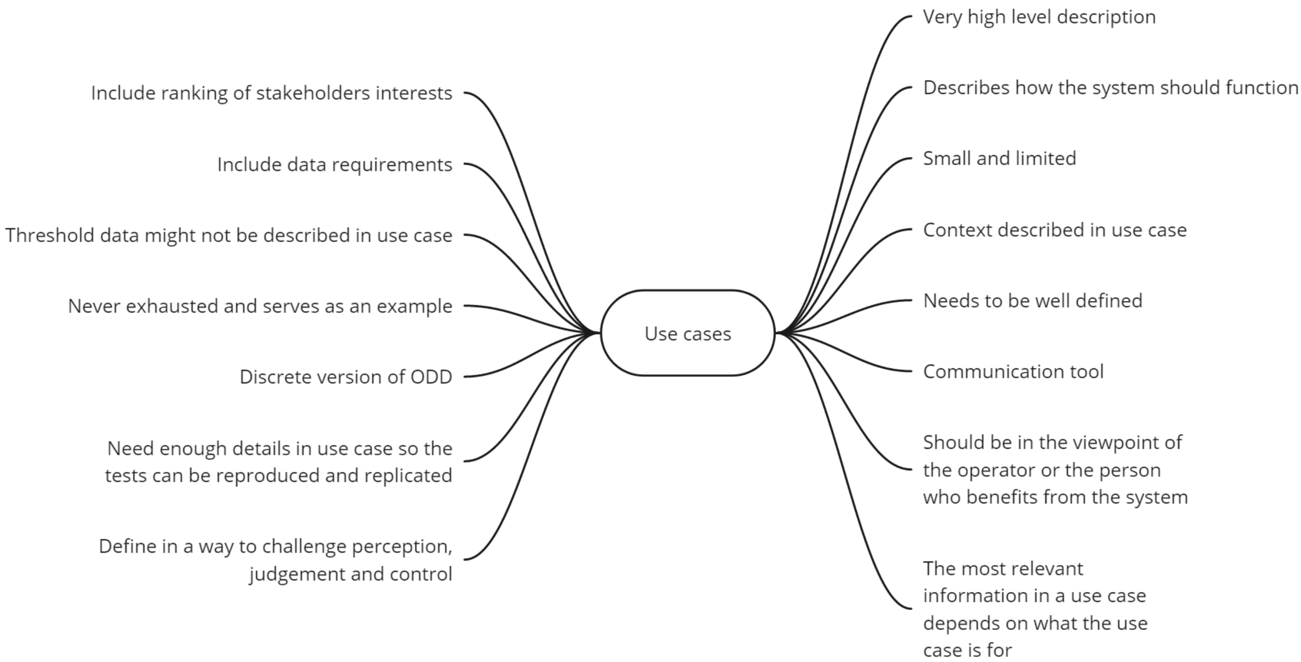

Figure 4.6: Themes identified for use cases

The use cases describe the system behaviour and the context of the system. They promote an easier and more effective way of communicating the intent of the system amongst the stakeholders, because the use cases are written in an common language. The use cases for ADS provide a higher level description of the system functionality and has limited information about the context in which the system is supposed to operate. This has been reflected by several interviewees. It was also mentioned that the use cases need to be well defined, so that it includes as much information about context as possible. Having well defined use cases helps to reproduce tests in an effective way. It was gathered from the interviews that the customer for whom the system is designed for, should be responsible for defining the use cases.

One of the interviewees mentioned that the use cases should be described in the perspectives of the operator or the person who benefits from that function, because use cases are descriptions of the system's functionality from the end users' point of view. This helps to ensure that the system will have the required functionality needed by the users. For example, use cases should be considered in the viewpoint of the driver and the other stakeholders surrounding the vehicles.

> *"[...] the driver benefits from having an automatic cruise controller and having higher comfort when driving. So that is the first perspective that*

> *has to be taken into account what is the benefit for the person using the function. [...] has to be considered in the use case [...]. "*

One interviewee suggested to include the ranking of stakeholders' interests in the use cases. That ranking would be based on the priorities of the stakeholders who has more influence in the use case. The use case should encompass the information about which stakeholders gets priority.

> *"[...] the pedestrian as a stakeholder has a higher interest of surviving, than the stakeholder, who is the driver. His interest would be to have a comfortable function and not having to steer the vehicle or accelerate and brake manually. But in that case, that vehicle has to do an emergency brake because the stakeholder pedestrian [...] outside of vehicle has a higher priority than the comfort of the driver. And that should be negotiated in the use case and defined in the use case as well "*

4.1.5 Current Process

The interviewees were asked to describe the current process for deriving requirements from use cases and encouraged to share their thoughts and opinions about the current process. The themes identified from the interviews are further elaborated in this section.

The interviewees were asked first if they were familiar with the current process. Four out of twelve were unfamiliar. Some admitted that they did not have so much insight on the process, not the level of expertise needed, and are unaware of the core parts in the process. The interviewees expressed that the processes differ between different countries and organisations. Even within some organisations, the process varies between different teams and components. The main underlying challenge that was expressed is that there is no established or common process for deriving contextual definition and requirements from use cases. This challenge was further explained in section 4.8. An interviewee described that it is hard to know which process is the best, since the process can be done in different ways. Furthermore, it is difficult to say if the process they are currently doing is good or not, since there are a lot of uncertainties about whether the process they are doing will work all the way through.

Interviewee C explained a process for more advanced automation systems. It was described that the process starts with deciding on validation strategy and safety cases to prove that the system is safe. Then they design backwards and structure the system in a way which is compatible with safety.

A process for deriving ODD was explained by interviewee F, which begins with analysing the use case to find out the context. Then define the context in which the use case is actually happening and that gives the first idea of what the context is. Then they analyse what are the internal conditions of the vehicle and the environment and then derive ODD from that context.

Interviewee A described a process for deriving requirements and ODD. The process starts with defining the use case that needs to be fulfilled. Afterwards, one continues with an analysis of what is required to fulfill the use case. Through the analysis, a lot of prerequisites will have been identified, which would be defined as ODD. The interviewee mentioned to consider all kinds of information and the inputs needed to the vehicle such as GPS signals, information from the road, when defining the prerequisites for the use case to be valid and safe. Through the analysis of use cases, the requirements on the ODD and also requirements on the actual vehicle are obtained.

Interviewee H discussed a process for deriving ODD and context. In this process, the use cases set the frame for the ODD. From the use cases, the constraints for the vehicle to operate i.e., the ODD are obtained. The additional constraints on the ODD can also be obtained from the sensors that are being used. Finally, the information about the context is derived from the ODD.

Interviewee K described a process for how ODD, context, requirements, and use cases are derived. An overview of the process is that the ODD is analysed, interpreted, and the use cases and requirements are derived from the ODD. Thereafter the ODD would be refined based on the use cases. It is an iterative process where both the use cases and the ODD influence each other back and forth. It was more specifically explained that they start with the ODD and then derive use cases from ODD. They reduce the number of use cases to the most relevant ones, after the initial phase where they have gotten a better understanding of the capability of the system or function. Then they derive requirements and get into the requirement management process. During this process, they bring back some potential deviations or issues where they can not comply and there will be a discussion with system, design leads, customer, and project manager. Thereafter, the requirements specification is defined and that goes into the design- and test phases. These were also described as iterative processes. In addition, it was mentioned that they rely more on simulations to cover the gaps between the use cases which are the discrete points in ODD. In the test phase, they start with maybe a hundred tests and at the end, de-serialise some of the cases and find the most problematic and critical ones. The key aspect from the tests is that they must be repeatable or reproducible, otherwise it is not so objective and not so useful. The use cases should have enough details to reproduce or replicate the test and the use cases are like instrumentation requirements for verification. The repeatability aspect of the test helps them to know about the different precision and accuracy metrics of the components.

A process for level 2 automation systems was explained by interviewee B. They start with getting requirements from the customer and breaking them down. Then they look into the current implementation and try to figure out what can be done with the current code. Then, they analyse if the current code can be adjusted, or if any conditions in the code can be changed, or write a completely new piece of code. It is important to understand the impact to the existing parts when changing or modifying the code. Therefore, the main focus is on analysing that the changes will not influence the performance of other parts. The interviewee also mentioned an example of AEB systems to explain how the changes in the code influence the performance. With the AEB system, they have a fear that a code change can trigger false braking requests in the system and that could be very problematic. This results in a lot of old cars starting to brake a lot more when they should not, after a software update has been made.

Interviewee E described a process for handling requirements. They get a set of requirements from OEMs, after which they figure out which requirements are too difficult for the prototype or which ones are unreasonable. Based on this, the requirements are pruned and then they negotiate with the OEMs on what would work now for this maturity level of a product.

Interviewee F described a process for obtaining safety requirements. A safety goal basically comes directly after defining the use cases. Then, they perform hazard analysis and consider what could go wrong with the function and what are the potential hazards. Based on this analysis, they define safety goals to prevent these hazards from occurring. Then, from these safety goals, they break down functional safety requirements that are on a functional level, as well as technical safety requirements that go into software component level and hardware level.

Interviewee K explained the process for sensor development. They target the ODD and they define the sensors based on the capabilities of the sensors. After that, they map these capabilities to ODD and then adjust the ODD and negotiate.

Interviewee E explains that the process for sensor development should start with a test plan that incorporates all the use cases, including corner cases. Then a lot of brainstorming is done to determine what will and will not work with the sensors. However, even before creating the test plan, some initial requirements could be derived. Based on the experience with the sensors and theoretical studies, they can decide on which metrics should be included, as well as write down some initial requirements. It was stated that when the product reaches a good stage, then they can test them out by having a test plan that covers the requirements. According to the interviewee, the sensors are not going to work in all conditions, therefore the requirements need restrictions. The requirements are not usually set, so they begin with requirements that sort of match the ideal ODD. Furthermore, the requirements are always related to how well the sensors should work. It was described that they need to understand the sensor's capability by knowing the sensor's quality metrics like accuracy, precision etc. It was also said that the verification and validation phases run in parallel with the requirements and design phases. It is important to test that the system is fulfilling the requirements. Interviewee E further explained that the road is very dynamic and in order to build confidence for what is being developed, the number of miles tested on the car makes a huge difference. The data collected will increase the confidence in the requirements, however often this is not enough because the data also needs to be diverse enough.

Interviewee D described their process of the deriving contextual definition and requirements from use cases. It starts directly from the use case, followed by a top-down and bottom-up approach. Ultimately, the aim is to make them meet in the middle. However, it was expressed that the two approaches never meet in the middle in a good way. As mentioned, they start off with the use case, then look at the specification of the sensor with respect to the environment and other things. Then they try to combine these, and from that they create the ODD directly. It was expressed that with this approach they never really do any context description.

Interviewee J described that the process of defining context and requirements uses the agile methodology 'Scrum', which allows it to be adaptive to the customers wants and needs. The system or feature being developed will be improved further with each iteration. It was suggested that it will help make it clearer to the customer what use cases they want. It was further stated that the process does not depend on individuals, instead it is a collaborative process with many stakeholders involved.

Interviewee I described a quite iterative process for deriving requirements. They start with making their best attempt, then refine iteratively, if needed. It was explained that they start off with what needs to be there, and then expand upon that along the way. The system engineers come from the way down. They get the use cases and define high level requirements. The interviewee I described, that from the specification of the sensor, the perception group can decide which high level requirements are out of the realm of possibility for the system or function. The developers will meet the system engineers at a certain point and interact, then the developers continue downwards from that level where they intermingled. To figure out what does or does not work when digging deeper into the requirements, the architecture is considered: Typical questions asked are whether or not failures occur too often, whether or not the availability is super limited by the ODD etc are looked into. Typically, they would derive both the use cases and ODD simultaneously. It was mentioned that they start with a strict ODD and expand it based on the data that prove safety. The HARA is then performed and some of the assumptions are used. Then, they figure out what needs to define the ODD from a safety standpoint. It was further explained that if restrictions are added to the ODD it might cause the need to create additional use cases that supports those restrictions. While refining the use cases, one might discover that situations are outside of the ODD, and the ODD needs to be refined. Then it goes back to adding more use cases, that informs what should happen in another scenario. However, it was described that it is not always necessary to add use cases, sometimes requirements would be enough to add. For example, if it was found out through a simulation that the system is bad at detecting ice. Then instead of having the ODD limit the driving, depending on if ice is detected, it could be limited by other input. For example, if it rained two days before the day or yesterday and the temperature is below zero then there is a high probability of ice. That could go into the ODD and then requirements are added which checks the weather and temperature, to figure out if the system is in the ODD.

Interviewee M explained how the ODD is limited. They need to have some preliminary image of the ODD before finding the technical requirements. After defining this preliminary domain, they start designing the system in a way that if it cannot handle certain driving conditions, then technical limitations are adapted and then feedback is given to the ODD. More limits are put in the ODD to secure safety. An example for, when limitations to the ODD needs to be added was given. The ODD for this example is that the car can work in all conditions on highways. The ODD is not feasible and needs to be limited because in some countries there is no speed limit for highways and the car cannot handle such conditions. They could then put a limit in the ODD for the maximum speed and say within that speed the car can operate safely.

Interviewee B and Interviewee F mentioned some important considerations when deriving ODD from use cases. The important considerations are:

- Sensor characteristics and performance
- How much computation to spend on the feature. For example, path planning takes a lot of computational resources

- Need to tune or modify ODD to handle all possible scenarios
- Input data from the sensors and data quality. For example, if the sensor data is under a certain quality threshold, then the system leaves the ODD and cannot ensure safety anymore
- Overview of the internal status of the system
- Monitoring the performance of the system and sensor data
- ODD might not always be limited by technical aspects, might also be due to budget and resources

4.2 Traffic Administration's interaction with OEMs

We got the opportunity to have an interview with a person from an european traffic administration and we will just put forth some interesting information from that conversation.

During the interview, when asked about the familiarity of the terms 'Context' and 'ODD', the interviewee was unsure about the context but familiar with ODD and agreed with the SAE J3016 definition of ODD. The interviewee stated that ODD is not generic and there are different ODDs for different functions and brands. Also the structure and format of the ODD depends on the OEM's perspectives, for example, some manufacturers have control towers for their vehicles and some do not, so they have different ODDs.

The interviewee further explained that the traffic administration does not deal with the ODD and they are mainly focused on the physical and the digital infrastructure. The interviewee mentioned that it is the OEM who defines the ODD, but was unsure regarding the responsibility for maintaining ODD. The OEM is responsible for ensuring that the different components of the system are working properly to ensure safety e.g., that they are able to read road signs or road markings. However, the transportation agency will question how they will make sure that the vehicle is safe, the features, and the functions they develop are safe. The transportation agency describes what features are acceptable to put on the market, and then the traffic administration must monitor if any problems occur with the approved vehicles on the road.

The interviewee further explained that the traffic administration cooperates with the OEMs on pilots for high-level vehicles that drive 10 kilometers in a confined area, like from a factory to the harbour. They discuss with the OEMs about potential needs for changes in the physical infrastructure. It could be about, for example, a roundabout or traffic signals that the OEM's current technique cannot handle and then the traffic administration could contribute with some changes if deemed beneficial. The hope is to learn what are necessary and beneficial contributions in the future.

However, serial production vehicles are not bound to a certain area and the traffic administration has identified what their contribution could be for those. The interviewee mentioned that the traffic administration does not believe that infrastructure modifications can be made to the entire network as of today. Especially not, when the requests would be based on assumptions about what impact these vehicles would have. It was stated that infrastructure changes last for 100 years, so the investments in adjustments today need to be useful and beneficial in the long run. It was further mentioned that in the case of serial production vehicles, the OEMs should not depend on road authority for making any special changes. This is because those changes might only be applicable in one country i.e., that their vehicles might not be able to operate safely in other countries, because what they are designed for is not there. If there are some minor changes requested in the physical infrastructure that is considered an improvement then the traffic administration would look into that. Also, if serial production vehicles are approved to be used in the communal road network, then the transport administration will help to enhance the traffic safety.

The transport administration view is instead that their main contribution is to digitise the current information that they are responsible for conveying to the human drivers, to enable the systems to utilise that information.

4.3 RQ1: Which current solutions work well for deriving contextual definition and requirements from use cases?

This section presents the themes identified from the interviews for the first research question (RQ1). The themes were identified with regards to what is working well in the current process of deriving contextual definition and requirements from use cases. The validation result is presented in Figure 4.7 and is explained for each theme.

Use cases as a communication tool

Use cases serve as a framework for gathering customer needs. It was explained that use cases help to easily communicate the system behaviour to the stakeholders, for example by sharing the design information. Use cases prevent the misinterpretation of languages among the stakeholders, because they are written in a common textual or visual format from the end users' perspective.

It can be seen in Figure 4.7 that all the four participants felt that use cases serve as a good communication tool in their current process. One of the comments was that the use cases are only effective if their description is complete and correct. It was also stated that requirement engineers should figure out if the use cases are complete and correct.

Simulations

In order to develop safe and reliable ADS, the autonomous vehicles must be tested to react to all types of driving situations, which entails lots of scenarios. Simulations play an important role in testing various scenarios in a more efficient and cost-effective manner. The interviewees described that simulations are used to catch errors as early as possible, to accelerate the development process, and improve the product quality. Simulations enable testing with complex scenarios for ADS. The interviews reflected that the simulations are an effective way to analyse the performance of autonomous vehicles in order to assure safety. The validation results for this research question are presented in Figure 4.7. The results show that two out of four participants felt that simulations are working well in their current process.

Iterative process and feedback loop

It was stated during the interviews that the process of deriving contextual definition and requirements is iterative and it is working well in the interviewees' organisation and teams. They have a good feedback loop built into the process for deriving requirements and refining ODD. It was explained that the iterative process allows to add more and more features depending on what the customer needs. It allows getting feedback early on from the customer by delivering valuable increments. This makes the development process a little faster and builds confidence. The interviewees also mentioned that the feedback loop in the process allows for better collaboration between teams and also that teams can act instantly on the feedback or adding it to the next increment. Most of the participants agreed to this theme, which is also visible in the validation results presented in Figure 4.7.

Process of deriving contextual definition and requirements from use cases

Some of the interviewees stated that the current process for deriving contextual definition and requirements from use cases is quite good and clear. However, there are still some assumptions that are misinterpreted, which results in a system that is either overly safe or unsafe. Two participants agreed that the process is a well established one. Also, it was discussed in the focus group that even though the process is an established one but it is inefficient and not that useful.

Breaking down into requirements

The interviewees stated that the current process for breaking down use cases into requirements works well. As a result, their current process has a greater traceability between use cases and requirements which aids in the understanding of requirements at any phase of the software development process. The results, given in Figure 4.7, shows that half of the participants agreed that the process of breaking down use cases into requirements works well.

Works well to test first and build an hierarchy of priority based on lessons learned

Testing early on builds confidence in developers and also helps in identifying the most critical and challenging functionalities early in the process. It helps in finding the most relevant and important functionalities from the lessons learned from the past experiences. The Figure 4.7 shows that two participants agreed that the testing early on is helpful in their current process.

Traceability

In terms of the process for developing systems with level 2 automation, one interviewee stated that they had good traceability between the changes in requirements. Traceability provides information about linked and reliant artifacts when a requirement changes. Furthermore, traceability makes it easy to track the impact of changes, and update the artifacts, if necessary. Figure 4.7 reveals that one participant found the traceability as something that works well within the process.

Too early to tell what is working well in process

It was explained by the interviewees that it is hard to prove whether the process is good or bad until ADS reaches a certain level of penetration into the market and proving that the system is really safe. It was also described that the autonomous drive is a new field and the international community tries to define the minimal requirements and rules so that the systems can be safely released. The validation results in Figure 4.7 show that it was agreed by one participant.

Identifying distinct properties in ODD

It was mentioned in one of the interviews that certain different properties should be identified and defined in the ODD. Those properties are as follows:

- Distinguish lanes
- Distinguish road types
- Distinguish city road from highways
- Identify environmental factors (external factors)
- Kind of objects in the road
- Properties of road users
- Traffic Density
- Traffic Rules

Identifying these properties helps to generalise and handle the process in a structured and standardised way. This theme was chosen by one participant, as shown in Figure 4.7.

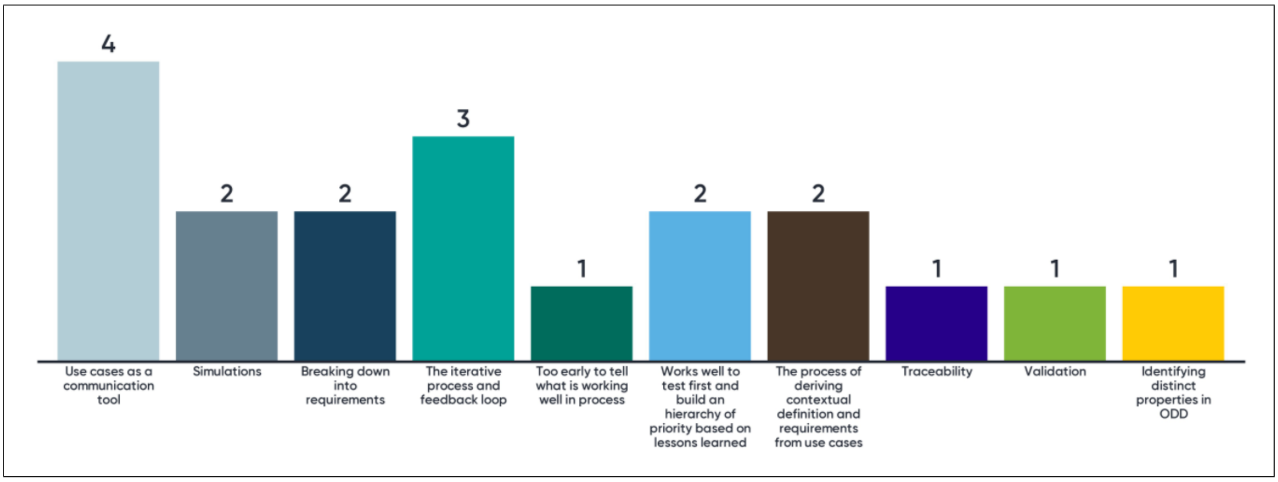

Figure 4.7: Validation results for RQ1

4.4 RQ2: What are the challenges with deriving contextual definition and requirements from use cases?

This section presents the challenges identified from the interviews. Six different areas of challenges were identified: 'Process, Responsibility, and Communication', 'Deriving requirements from use cases', 'Defining ODD', 'Deriving context', 'Validation', and 'Standards and Regulations'. These are collectively presented in a Ishikawa diagram, see Appendix A.5. The sub-sections will present each challenge connected to respective challenge area.

4.4.1 Challenges with process, responsibility, and communication

This section describes the themes related to process, responsibility, and communication, that were identified from the interviews. The challenges identified are presented in Figure 4.8.

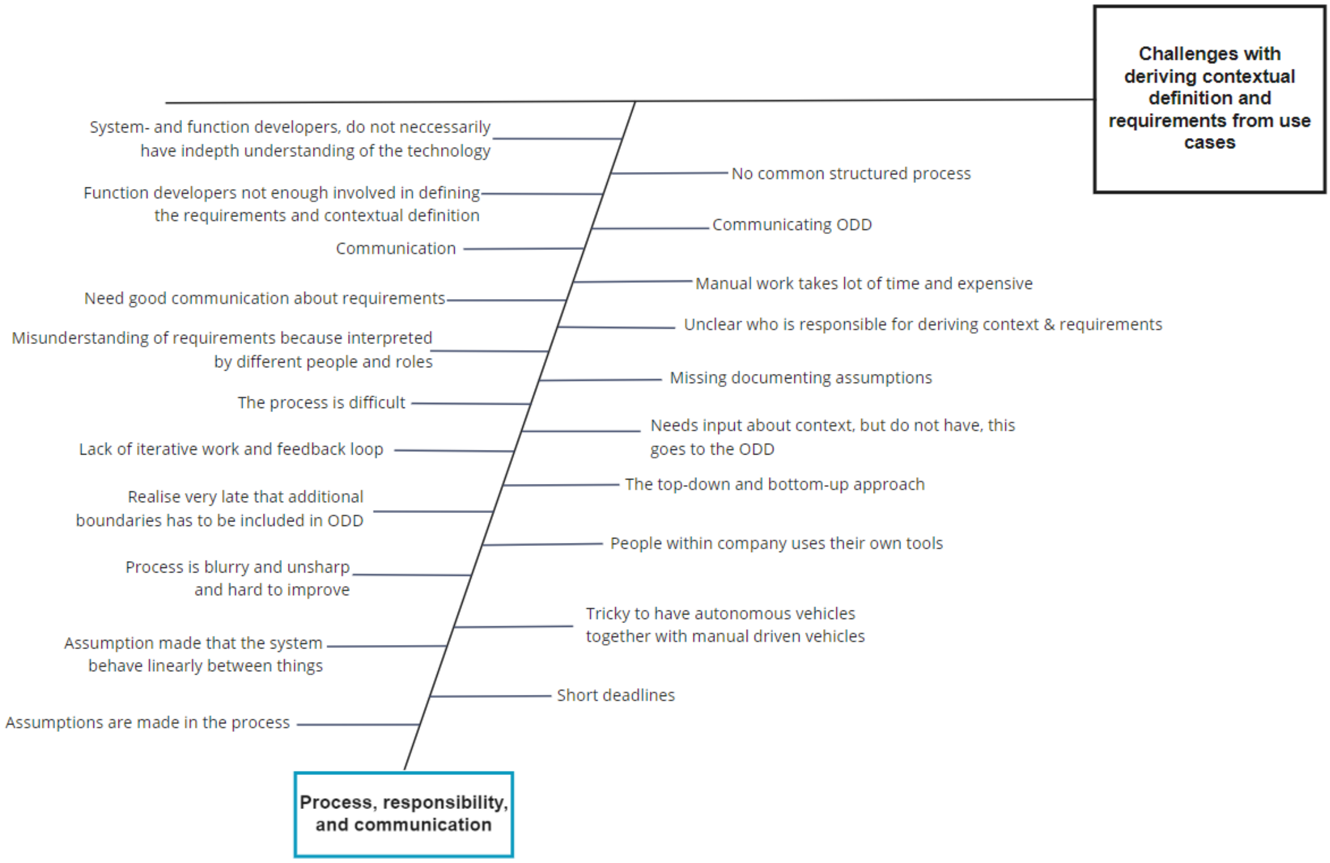

Figure 4.8: Part of the Ishikawa diagram, for challenges with Process, Responsibility, and Communication

System- and function developers, do not necessarily have in depth understanding of the technology

This challenge was mentioned as a main concern. The concern is that the people who develop the system and function do not have an in-depth understanding of the technology and how it affects other things. Without this understanding, it can be more easy to overlook important parameters, miss an assumption, or not properly interpret for example requirements.

Function developers not enough involved in defining the requirements and contextual definition

It was stated that in some organisations the functional developers are not involved enough in defining requirements and context definition. The lack of involvement makes them a bit detached from the overall picture of why certain requirements are there. It makes it harder for them to understand and see the purpose of certain requirements. It is important to know why certain requirements are there to avoid strange constructions in the system or software.

Communication

It was stated that it is important to communicate and make sure that everyone is still on the same page regarding things between steps in the process. Misalignments and decisions between levels and documents should also be properly documented and communicated. For example a decision made at the implementation level, might never be communicated back to the design level. So even if the implementation level documents have this information, it is missing in higher levels. This should be avoided.

In the top-down and bottom-up approach, it was expressed that no full context description is done and that this description is missing for more effective communication between customers and developers. Additionally, it was expressed as not only a technical problem but also a communication problem. This is because they need to communicate to the customer and the car manufacturer, and make them understand how they are going to describe where the system is going to work. Overall, it was identified that the interviewees expressed a need for good communication within the process in order to achieve success.

Misunderstanding of requirements because interpreted by different people and roles

The challenge raised was that different people and roles can interpret the same requirements differently. It is important to avoid or mitigate misunderstandings. It is therefore important that people defining the requirements interpret them the same way as e.g., the developers.

No common structured process

One main concern raised in the interviews was that there is no common structured process for deriving requirements and contextual definition. It varies between organisations, but also within an organisation. One stated that there are some suggestions on how it should be done, but hardly anyone uses it.

Communicating ODD

ODD definitions are described, according to an interviewee, as being written with a specific purpose. The challenge occurs when there is a need to communicate about the ODD. Even if the ODD is written with a specific purpose, it can be used for a multitude of things. It is important, especially in large organisations, to be clear about what type of ODD is being referred to. There is a challenge in communication between departments when they have different viewpoints on, in this case, the ODD. There was an emphasis on language, specifically how things are written when defining the ODD. The words used should be clear, so the user and system will know if they are supposed to be working in a context or not. A concern raised was regarding that the ODD should not only be written so the system or car developers understands. There are other people that the interviewee considered should be able to understand what is written, there amongst users, lawyers, insurance providers, and policy makers.

Manual work takes lot of time and expensive

A lot of manual work is involved in the process. It takes a lot of effort and is not that effective. When implementing new functions, a lot of manual work with checking the impact of the changes in the code is done, which takes a lot of time. It was expressed that there should be more systematic ways of doing things, and that they would like to rely more on testing and simulation.

Unclear who is responsible for deriving context & requirements

According to the interviews, it was quite unclear who is and should be responsible for deriving context and requirements. Some suggestions were discussed, but all with some level of uncertainty. This discussion is shown in Section 4.1.3 and Section 4.1.1.

Need good communication about requirements

The people who define the requirements need to be able to communicate very efficiently with each other to prevent that the requirements are contradictory or get hard to implement.

Missing documenting assumptions

A main concern mentioned was that assumptions made in requirements and ODD are not being documented properly. It is important to make sure no assumption is forgotten, because that might create gaps e.g., that there is a lack of requirements that fulfill the whole use case.

Needs input about context, but do not have, this goes to the ODD

The concern is that the proper limitations for the context should still be described in the ODD, even if the input for the context cannot be retrieved by the system. For example, when the system does not have input about 25 millimeters of rain, then the context should describe that the input cannot be measured or obtained. Based on that information in the context, the limitation that the system can operate as long as it rains less than 25 millimeters per hour should be added to the ODD. The concern is to find a good way to describe the context and add the limitations in the ODD automatically, when the input could not be retrieved.

The top-down and bottom-up approach

In the top-down and bottom-up approach, described in Section 4.1.5, it was described that even if the work from top and bottom are supposed to meet in a good way in the middle and integrate, they never do in reality. It causes the development time to be prolonged and it gets expensive. As well it was described that this process kind of forces a choice about which sensor to use early, which results in too conservative requirements that are applied to the sensors, which results in an overdesigned, and therefore unnecessarily expensive, sensor system. If the sensors could be chosen later, the system could avoid being overdesigned.

Lack of iterative work and feedback loop

The challenge described is to have a good feedback loop between stages of the process. There is a need to make sure that changes that occur later in the process of developing a system are being reported back and adjusted accordingly. It was specifically mentioned to be a lack of feedback and iterative work with ODD and requirements in the top-down and bottom-up approach.

People within company uses their own tools

It is mentioned that different departments, within a large organisation, use their own tools. It is also quite a sensitive subject, according to the interviewee. This was brought up to shine light on the fact that it might be difficult to have standard tools to be used in the process, since people are used to working with certain tools.

Process is blurry and unsharp and hard to improve

The process of deriving ODD and requirements from use cases was described as being blurry and unsharp by nature. It was described to be hard to improve.

Tricky to have autonomous vehicles together with manual driven vehicles

It was described that it is tricky to integrate autonomous fleets together with the common manual driven vehicles. It is believed that the autonomous vehicles will, more or less, be standing still quite a lot. Therefore, this is believed to be a big challenge. The new transportation implemented still needs to be efficient, to be useful.

Assumption made that the system behave linearly between things

There are assumptions made in the development process that the system is going to behave somewhat linearly. This is not just assumed for the system, that assumption also applies to everything in the context. The reality is that not everything follows linear behaviour, some things are unexpected. There is a big challenge on how to consider and include these uncertain trajectories in the context description.

Short deadlines

It was described that even if the process was considered good, having short deadlines was described to be challenging because it gets harder to document in a proper and systematic way. It was also brought up that tracing back requirements to use cases takes a lot of time.

Assumptions are made in the process

It was described that assumptions might not always reflect reality, which makes them challenging in the process. Assumptions about movement, in context, are often made and are often more simple in comparison to reality. An assumption of movement is described in the challenge 'Assumption made that the system behave linearly between things'.

The process is difficult

The iterative process with a focus on trial-and-error was described as being not an easy process. It was mainly difficult because there are quite a number of things to consider and a lot of assumptions have to be made during the process.

Realise very late that additional boundaries has to be included in ODD

It was described that if there is a lack of a complete ODD, and the necessary additional boundaries are only discovered during the development of the system, then those have to be included in the ODD very late. This challenge was raised in connection with the challenge 'Communicating ODD'.

4.4.2 Challenges with deriving requirements from use cases

This section described the themes related to challenges with deriving requirements from use cases, that were identified from the interviews. The challenges identified are presented in Figure 4.9.

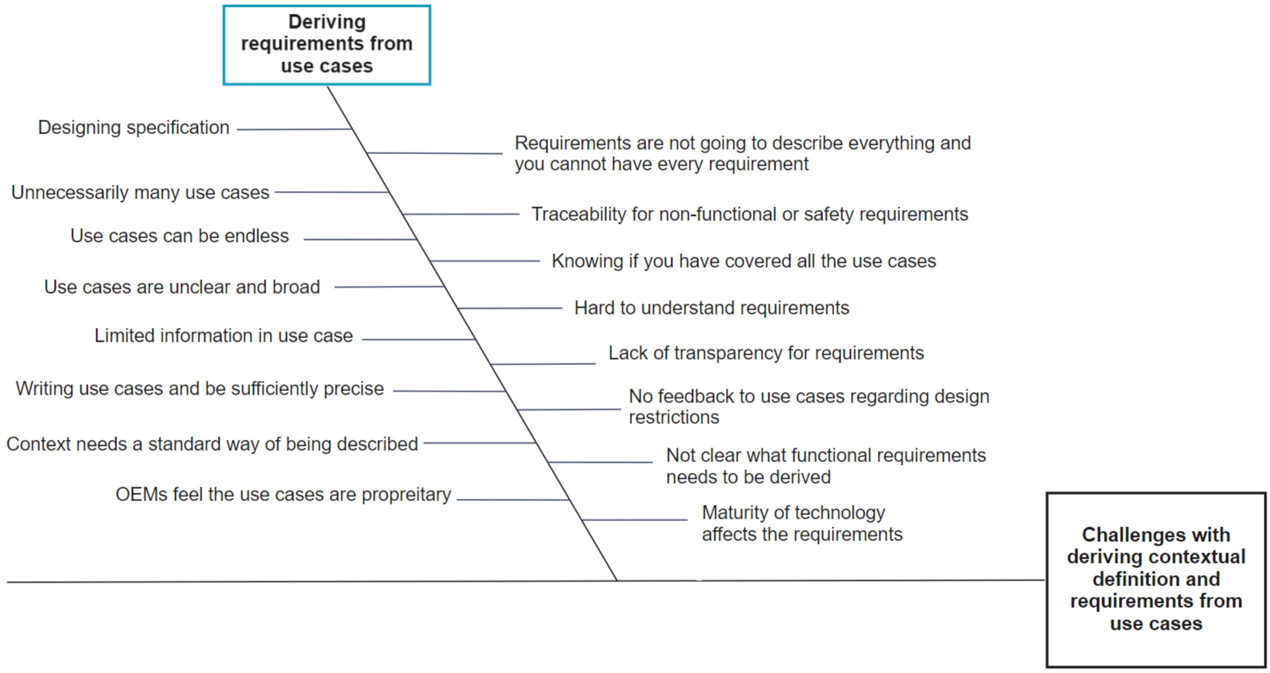

Figure 4.9: Part of the Ishikawa diagram, for challenges with Deriving Requirements from Use Cases

Unnecessarily many use cases

The challenge is to avoid having unnecessary many use cases. It was described by Interviewee I, that one do not want to have too many use cases, because with the number of use cases increases the number of requirements one have to derive and validate. The focus should be to make the use cases simple yet encompassing.

Use cases can be endless

Driving is described as an extremely dynamic task, and that it is hard to foresee every use case. In fact, in practice the use cases are described to be endless. The people creating the use cases might not even know all the use cases that exist.

Use cases are unclear and broad

A main concern raised was about being able to understand the use cases. It was mentioned that use cases should be simple yet encompassing, but that is hard. The reality is that use cases can be unclear and broad. The consequence of having unclear use cases is that when deriving other work products, e.g., requirements and ODD, from use cases, information is easily missed or misinterpreted. It was stated that it is important to clearly understand which main requirements are needed to achieve a minimal viable product in the initial stages.

Limited information in use case

In Section 4.1.4, the use case is described as a very high level description. The challenge with this is that the use cases will have limited information available. That limited information leaves room for assumptions to be made.

Writing use cases and be sufficiently precise

Use cases can be difficult to write and another concern was stated, that it is challenging to be sufficiently precise i.e., it is sometimes challenging to provide specifications only in written text. It was brought up that use cases might be easier to simulate and model instead of writing the information down. An example of what could be harder to write down is properties like staying within a certain distance.

Context needs a standard way of being described

This challenge relates to the challenge ‘Some parameters are hard to describe in context’ described in Section 4.4.4. It states the need for a standard terminology for parameters, and similarly this challenge presents the lack of a standard way of describing the context. It is expressed that the context needs to be formulated especially when the complexity increases and for safety critical systems. There is a risk to misunderstand the specific context that the requirements are written for, if the context is not properly described and cannot be clearly understood. Therefore, the need for some sort of standardisation for the context was emphasized.

OEMs feel the use cases are proprietary

It is described that the OEMs can feel that use cases are part of their intellectual property, which is expressed as a challenge. If the OEMs feel ownership over certain use cases, then when another customer comes in, the same use cases cannot be reused.

Hard to understand requirements

Some requirements could be quite hard to understand according to the interviewees. The reasons mentioned were the lack of transparency behind them, and that requirements could also be understood and interpreted in several ways. This difficulty does make it harder to make proper design consideration on what can be done and what can be delivered in a certain time frame.

Lack of transparency for requirements

The lack of transparency for requirements was raised by developers. They stated that the lack of transparency makes it more difficult to implement the requirements in a good way. It might even lead to a strange or inaccurate construction of the system. It was explained that sharing the justification for the requirements would help greatly in the development and was even considered to be important for safety reasons.

Requirements are not going to describe everything and you cannot have every requirement

The challenge here is making the decisions on what requirements to include, because a system cannot deal with everything. The requirements to include can be limited by for example the maturity of technology and resources. For example, it could be that one cannot have cameras that have the same accuracy during night and day, because the context is different, which requires different technologies, and thereby affect the accuracy that can be expected in different contexts. Another concern mentioned is that the requirements are not going to cover every context or scenario.

Not clear what functional requirements needs to be derived

In the iterative process where one starts creating the minimal viable product and having a more trial-and-error approach, it is described that it is not clear what functional requirements need to be derived. It is described to be quite a challenging task and that a lot of assumptions must be made.

Maturity of technology affects the requirements

It is described that the maturity of technology will affect the requirements. The design document might contain requirements about that the system must handle certain things in a given way, but there might be technology restrictions that affect how the requirements are written. For example, the operation design document can state that everything should work during the day and in some specific weather conditions, but that might not be possible with the current available technology e.g., there could be limitations to the sensors. The requirements will then be written out slightly differently from what the operation design document states.

Designing Specification

It is a difficult task to design the specification, because there are a lot of people involved in the process and a lot of negotiation.

Traceability for non-functional or safety requirements

The requirements management process was described and that in the end of it one would have certain safety requirements. It was noted that it is very important to be able to trace back those safety requirements to the safety goal that one defined very early in the development process. This was described as being quite challenging, because the tree that spans from the safety goal down to the technical safety requirements can be extremely big. It gets difficult because one will need to be able to test everything that is safety relevant for the function. Therefore, it is important to have an overview of all the safety requirements according to the interviewees. The challenge and concern were described to be viewed similarly for the functional requirements.

Knowing if you have covered all the use cases

A main concern stated is that one might not have included all the use cases that can come up. Driving is an extremely dynamic task, which makes it harder to foresee all the use cases. If not all use cases are covered, then that can result in not all the proper requirements being derived and that the ODD is not set up correctly. One interviewee stated that *"you are never sure you got it all, but have to try and that is kind of the only thing you can do."*.

No feedback to use cases regarding design restrictions

A concern was raised that information about misalignment between use cases and other documents might not be communicated back to the higher levels of the design process. It was specifically described that if certain scenarios are disregarded further down in the process there is no feedback given about it back to the use case.

4.4.3 Challenges with defining ODD

This section describes the themes related to challenges with defining ODD that are identified from the interviews. Figure 4.10 presents the challenges identified for defining ODD.

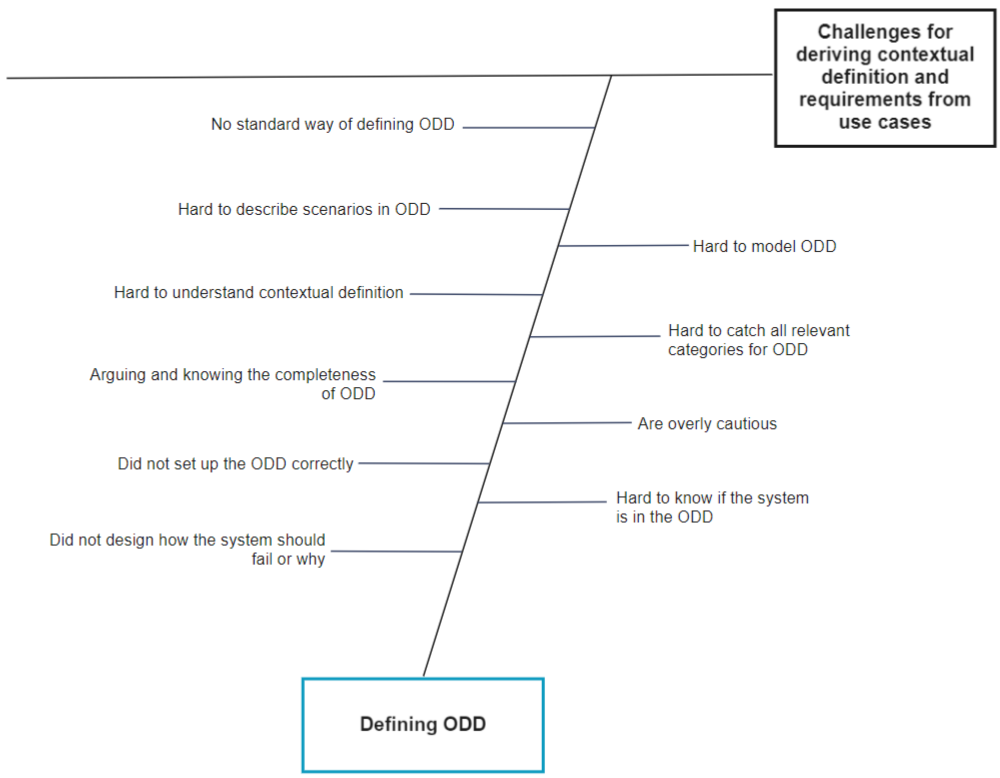

Figure 4.10: Part of the Ishikawa diagram, for challenges with defining ODD

No standard way of defining ODD

Interviewees reported that the ODD is a concept that has been adopted by a lot of disciplines, and everyone tries to customise it to their own use cases. So when individuals say they have an ODD, it tends to mean different things and makes it difficult to define. To minimise unnecessary customisation, it would be beneficial to have some form of standardised ODD.

It was mentioned that the different OEMs have different approaches to define the ODD, revealing that a standardised approach does not exist. Even within an organisation, different teams and roles have distinct perspectives on the ODD definition. Other issues that emerged from the interviews were that there exists no clear view in the industry on what to include in an ODD, how to set up and fill an ODD, and most importantly a lack of standardisation for the definition of an ODD. The lack of a standardised approach is considered by the interviewees as one of the major challenges. It leaves room for the companies to have their own interpretation on how to define the ODD, which can be costly, ineffective, and prolong the time it takes to launch a product on the market.

Hard to describe certain scenarios in ODD

Autonomous vehicles are designed to operate in any driving situations within the ODD, which can include any number of different scenarios. The scenarios need to be precisely described in order to develop safe and secure autonomous vehicles and it can be difficult or even impossible, to enumerate all possible scenarios.The interviewees expressed that it is hard to predict and describe all the scenarios within the ODD. The scenarios can also differ a lot between different countries, regions, and cities. This increases the difficulty in properly describing all the possible scenarios that need to be covered to ensure safety in the real world.

There are also some scenarios that are more difficult to foresee and express in the ODD. These could, for example be weather related scenarios and specific combinations of scenarios, like low standing sun with snow on the road and scenarios during dusk and dawn.

Hard to model ODD

It becomes quite complex to model the ODD, if all kinds of behaviours and scenarios described in the ODD should be included. The interviewees described that the edge case scenarios are especially tricky to model, since predicting what kind of edge case scenarios should be explained in detail is challenging. The scenarios in which the vehicle violates the ODD, i.e., in which it operates not safely, are considerably much more difficult to model.

Hard to understand contextual definition

An interviewee mentioned that function developers are not involved enough in the process of defining the ODD, which makes it difficult to understand the ODD. As a result, they are unsure what is described and included in the ODD. In addition, it is considered more difficult to implement a system in a good way, when it is not known exactly what the purpose of the requirements are and why they are stated. As a result, the system or software will probably be implemented in unexpected and inefficient way.

Arguing and knowing the completeness of ODD

The interviewees mentioned that it is important to define the ODD in such a way that, depending on the intended use case, completeness can be argued. However, it is hard to understand and determine whether the ODD has included all the sufficient elements. They also stated that there is no standardised method or template for determining that the ODD is complete. Since, currently given is a list of situations or context in which a function or autonomous function can operate.

Are overly cautious

One interviewee expressed a main concern for being overly cautious when defining the ODD. The interviewee explained that this concern is raised for when starting with a strict and limited ODD, and then only expands it once they have proved that the system can operate safely within the extension. Another stated that the main concern is the ability to prove safety. To prove safety, one could rely on a lot of data to know where the focus on proving safety should lie.

Did not set up the ODD correctly

One interviewee stated that their main concern is that the ODD is not set up correctly because all the use cases were not included, and that they sometimes forgot to add use cases. The underlying problem is a lack of process for deriving the ODD from use cases as well as for ensuring that the ODD is correctly set up.

Did not design how the system should fail or why

An interviewee explained that the industry does not spend effort to define when the system should fail, i.e., when the system is in the non operative domain. The non operative domain is not defined currently in the industry due to liability issues as well as a reluctance to do so. However, in order to design safe and secure systems, it is important to understand where the system will function and where it will not operate.

Hard to know if the system is in the ODD

It was mentioned that it is challenging to decide if the system is within the ODD and to decide if the system should continue operating. It was also expressed that the industry is lacking in measuring if the system is still within the domain.

Hard to catch all relevant categories for ODD

The ODD has different categories which can entail road conditions, lane conditions, road users, traffic environment, and other external factors. Identifying these categories helps to handle properties of the ODD in a similar way and generalise the process of defining the ODD. However, it is difficult to identify all the relevant categories and to know if all the categories have been covered.

4.4.4 Challenges with deriving context

In this section challenges identified from interviews for deriving context are presented. Figure 4.11 presents the four specific challenges identified for deriving context.

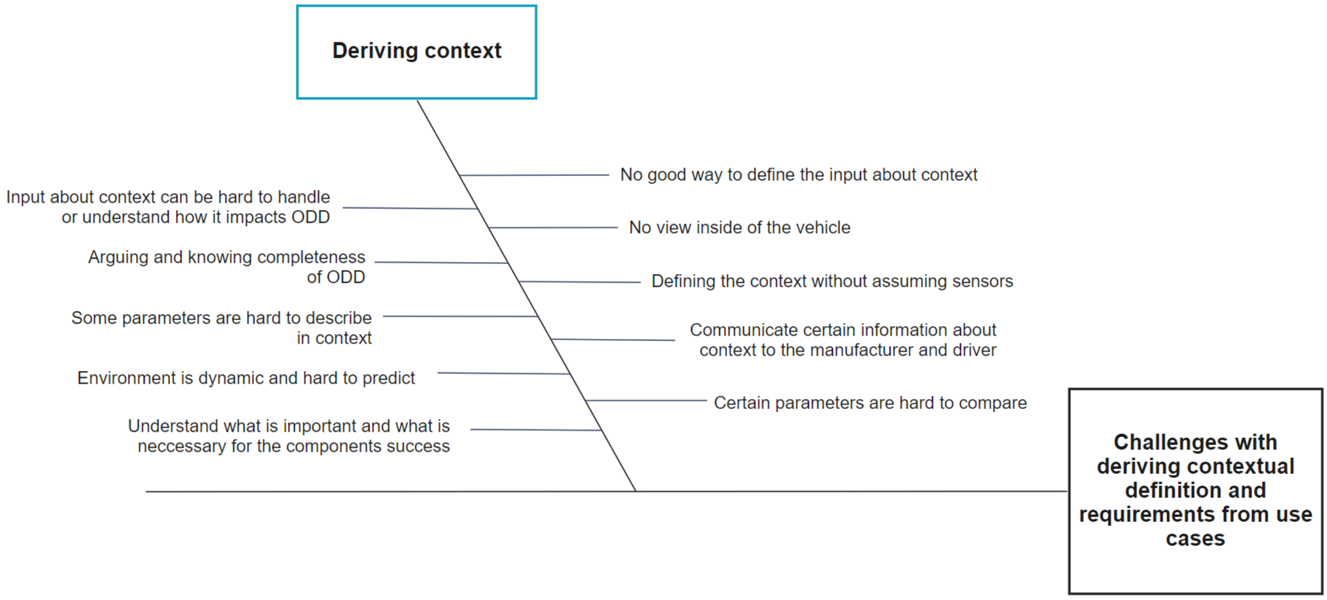

Figure 4.11: Part of the Ishikawa diagram, for challenges with Deriving Context

Arguing and knowing completeness of ODD

The challenge of arguing and knowing the completeness of ODD is described in Section 4.4.3 where challenges when defining the ODD are presented. This part will investigate the underlying challenge, in regards to context, that affects the identified challenge. ODD is described to be the limitations set up to limit the context that the system/function can operate within. Therefore, it can be said that, if it is hard to argue and know if the whole context is described in a proper way, then that can affect how well the completeness of the ODD can be argued.

Some parameters are hard to describe in context

There is a lack of terminology for describing parameters within the context. It was stated that some parameters are easy to describe, and easily understood by everyone. An example of such a parameter is weather, like for example rain, snow, precipitation, et cetera. However, many parameters are more complex, and thus require some form of nomenclature.

Environment is dynamic and hard to predict

The real world is complex and there are a lot of unknowns when the car is driving out on the road. The environment is extremely dynamic and it is challenging, maybe even impossible, to foresee everything. The context, in which the use cases are valid, is hard to predict. The context changes around the use case, for example, when going to another country the surroundings changes.

Understand what is important and what is necessary for the components success

In the development of a system or function, there always will be negotiation about what needs and can be developed. It was described that there is not enough transparency on the reasoning behind the requirements, which creates a challenge when making design considerations. To be able to make proper design considerations, it is important that the people involved understand what is important and necessary for a component's, system's, or function's success.

Input about context can be hard to handle or understand how it impacts ODD

Some input about context can be easier to handle, whilst some can be more complicated. An example given was joint input or the result of joint input. For those it can be quite a challenge to handle or understand how that input impacts the ODD.

Communicate certain information about context to the manufacturer and driver

This challenge was brought up in addition to speaking about the challenge 'No good way to define the input about context'. The situation described was that the sensors are not able to get input about certain context, i.e, it is difficult to monitor the current context. The system needs to know that their context is within the ODD, so it can operate safely. The challenge brought up specifically was that even if the ODD could be set up accordingly, it will be hard to communicate that information to the car manufacturer and driver. The driver, for example, would need to get the information that they had to watch out for certain scenarios where the system is not going to operate.

Certain parameters are hard to compare

It was expressed that requirements, and context, need to be formalised further so those could be comparable. Certain parameters like distance to an exit on a highway or number of pedestrians is easy to compare, but there exist harder ones. Parameters that are harder to compare are for example weather.

Defining the context without assuming sensors

It was stated that it would be good to do system level simulations early. The challenge is that, in order to do that, one needs to know the context, and not assume any type of sensor. For the simulations it was stated that agnostics sensors should be used, i.e., the input to the simulations should be some type of data from the context.

No view inside of the vehicle

It was expressed that it is an issue that an ADAS does not have a view inside of the vehicle. For certain levels of automation there might be a need to know certain information about the context within the vehicle to ensure that the system is within the ODD.

No good way to define the input about context

The challenge is when a scenario occurs and the sensor system has no way of providing input that tells the system it is in that context. An example stated to further clarify the challenge was that the context could be that there is low standing sun and ice on the road. The issue is that only one of these can be detected and the other one, which cannot be detected, still impacts how well the road can be seen with the camera. Furthermore, there is no way of knowing how the different aspects of the context (low standing sun, in combination with ice on the road) together impact the camera. Even if these limitations are stated in the ODD, the difficulty is that the system cannot understand that this is the context that it is operating in. The ODD knows the limitation, but the sensor system has no way of providing that input which could tell the system they are in that scenario.

4.4.5 Challenges with validation

This section describes the themes identified for challenges with validation and the themes were presented in Figure 4.12.

Validation and testing

Some interviewees stated validation as a main concern. It was further explained that checking against context is hard and expensive. To allow for testing against context, the internal states of the vehicle and some additional information from sensors, for example processing power need to be known.

An interviewee described that the validation strategy is the hardest part, since they start with proving that the system is safe, then go backwards and structure the system in a way which is compatible with that.

It was expressed that it is extremely difficult to specify the behaviour of other vehicles and especially the tail behaviours are hard to quantify. As a result, it is more challenging to simulate or test these behaviours using formal methods.

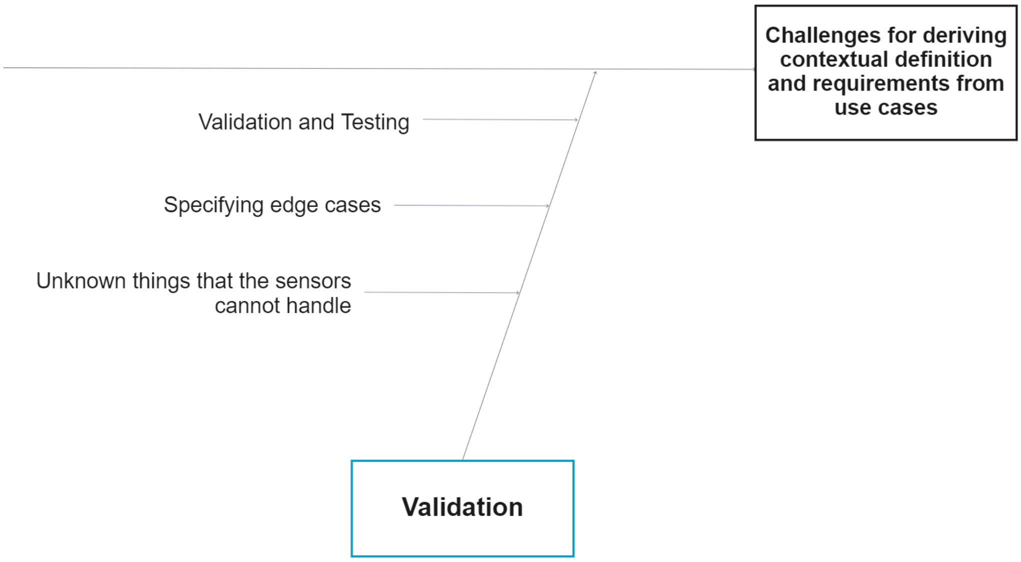

Figure 4.12: Part of the Ishikawa diagram, for challenges with Validation

An interviewee explained that in the current process, the sensors are deployed on the car and tested directly on the system to see that the system actually fulfills the use cases. The interviewee considered it as an expensive and inefficient way of testing. It was also mentioned that doing system level simulations without assuming any sensor or processor performance would be preferable to ensure that the system fulfills the use case.

It was stated that testing the self-driving car by driving a certain amount of miles on the same road on a highway is not useful. It was further explained that when deploying self-driving cars in other countries, there are a lot of unknowns and different scenarios, which are not as obvious for the sensors as they are for humans. Because, the sensors have not been tested under a lot of different conditions and scenarios in different countries. In addition, it is important to have a diverse test set, otherwise, the sensors will not work on the cases it has not seen.

The testability of context and ODD is expressed as a main concern by an interviewee. It is considerably challenging to test that the ODD is complete. Furthermore, it is challenging to test what happens if the system leaves the ODD; that the system can be able to detect that it leaves the ODD, such that the autonomous vehicle can come to safe stop. It was specifically mentioned that one should have test cases that not only test if the system/function is within the ODD, but also those that test if the system/function leaves or is outside of the ODD.

The challenge is to do objective testing when the driver is in the loop and that there is no good way to do that. Also, defining pass or fail criteria is more challenging when doing objective testing.

Specifying edge cases

The challenge is to specify the tail behaviour in a way to prove that it is correct. There is a need to realise what kind of edge cases to explain in detail and what kind of edge cases to be abstract about. The more abstract the edge cases are, the more likely it is that behaviours that will actually never occur has been taken into consideration. In addition, the edge case behaviours are more challenging to describe and there is no straightforward process to describe them in a good way.

Unknown things that the sensors cannot handle

The sensors are not tested in all different conditions and situations since it is time consuming and expensive. The sensors can be tested hundreds of hours to ensure that they fulfill all the use cases. Yet, there are some unknowns that the sensor will not be able to handle if they are exposed to conditions for which they have not been tested in. It was mentioned that despite guarantees from the suppliers and understanding from theories, there are still some unknowns that the sensors cannot handle.

4.4.6 Challenges with standards and regulations

The section describes the themes identified for challenges with standards and regulation which is presented in Figure 4.13

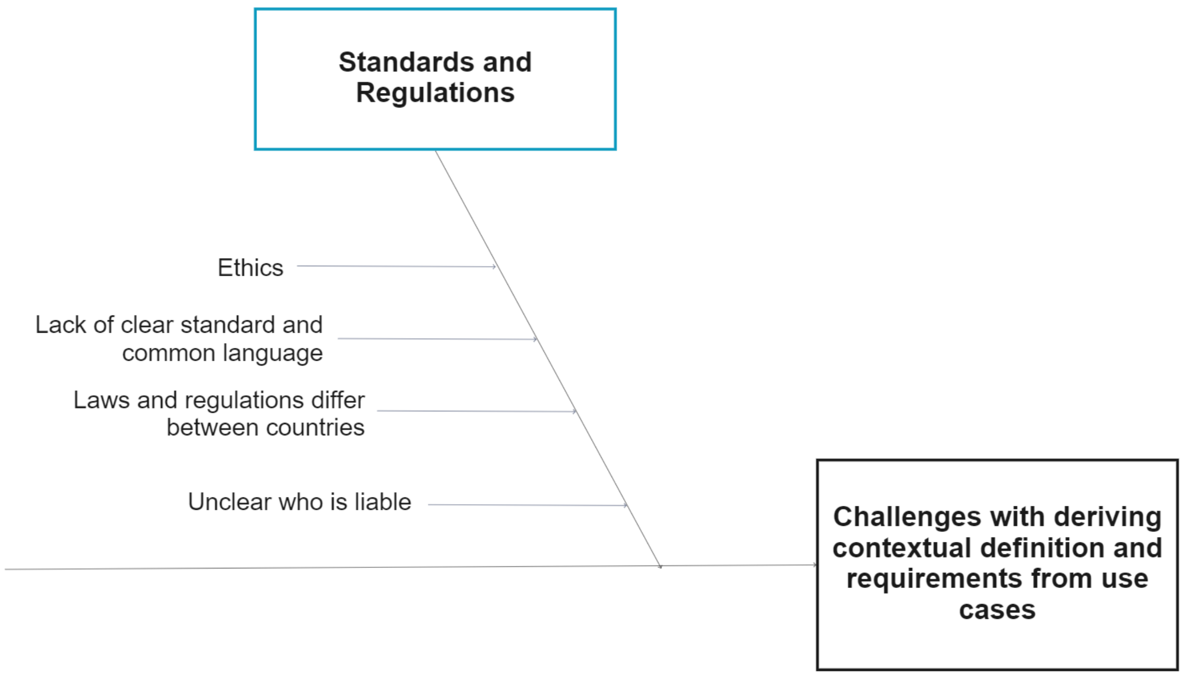

Figure 4.13: Part of the Ishikawa diagram, for challenges with Standards and Regulations

Ethics

One interviewee stated ethics as a critical challenge. Varied countries, individuals, and environments may have different ethical standards. Even within a country people may think differently, have different thoughts and opinions, and no one will accept the same thing. It was further explained that the major issue with ethics is what is acceptable or not for a new technology in terms of safety.

Lack of clear standard and common language

Some interviewees expressed that there is no clear standard on how to derive contextual definition and requirements from use cases. The process of deriving is carried out differently between different countries, companies, and teams. Even within a company, different products are using different languages. As a result, it is more difficult to ensure that the process carried out by the companies is good enough to produce safer vehicles in the society. A lot of new initiatives for a common standard is being developed now in order to have safer vehicles and the environment, for example SAE J3016 [1] and BSI PAS 1883 [52].

Laws and regulations differ between countries

Different countries have different laws and legislation. Some countries legally accept some technologies, some countries do not. For example, in some countries, level 3 and level 4 automated vehicles are not legally accepted to put on the public roads whilst accepted in public traffic by others. Because of the diversity of legislation between countries, and sometimes even between states and regions within the same country, it is difficult to develop a standard approach at the international level, resulting in a variety of approaches and standards among organisations.

Unclear who is liable

There exist a lot of unclear legal requirements about who is responsible and liable, in case of an accident, or other technical failures, occur while self-driving.

4.5 Validation Result for Challenges

In this section the validation result for the challenges are presented. There are four challenge areas that validation results will be presented for. The four are 'Process, Responsibility, and Communication', 'Deriving requirements from use cases', 'Defining ODD', and 'Deriving context'. There is no validation result for the challenge areas 'Validation' and 'Standards and Regulations', because from the workshop these challenges were deemed less interesting and important for this thesis.

4.5.1 Validation of the challenges with process, responsibility, and communication

Input regarding the challenges identified for 'Process, Responsible, and Communication' was collected from the focus group. The focus group was faced with the question 'What are the challenges with deriving contextual definition and requirements from use cases?' and then they could confirm the challenges presented that they thought was a challenge for this area. The result is shown in Figure 4.14 and Figure 4.15 and further explanation of the challenges is found in Section 4.4.1.

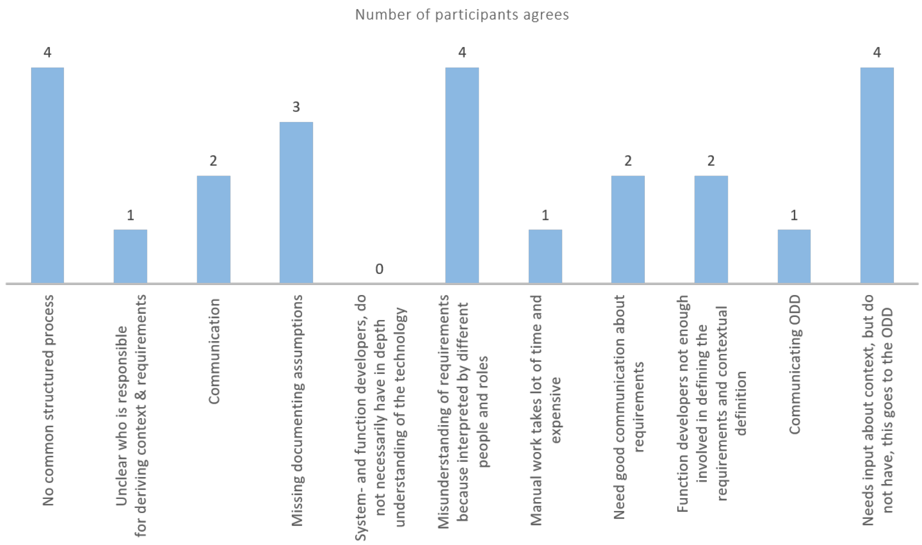

Figure 4.14: Validation result diagram for the challenges in Process, Responsibility, and Communication with four participants

It is notable that all participants in the focus group agreed on that the following three challenges from the area 'deriving contextual definition and requirements from use cases' are relevant: 'No common structured process', 'Misunderstanding of requirements because interpreted by different people and roles', and 'Needs input about context, but do not have, this goes to the ODD'. The motivation behind the challenge 'No common structured process' was that the participants thought that the current process is generic, hard to use, outdated, and not useful. One of the challenges, 'Missing documenting assumptions', three out of four agreed was a challenge.

There are three challenges that the focus group had divided opinions about. These are 'Communication', 'Need good communication about requirements', and 'Function developers not enough involved in defining the requirements and contextual definition'. The challenge 'Communication' was discussed and it was stated, by the ones who agreed it was a challenge, that communication is always an issue. In any way, it is difficult for the stakeholders to communicate and capture the contextual definition.

There were three challenges that only one participant thought was a challenge. Those are 'Unclear who is responsible for deriving context & requirements', 'Manual work takes lot of time and expensive', and 'Communicating ODD'. The challenge 'Unclear who is responsible for deriving context & requirements' was motivated as a challenge because it is unclear if a part of the process should have someone responsible for it. However, it was described as not being a big challenge and that it is easily solvable.

There was only one challenge that everyone agreed was not a challenge and that was 'System- and function developers, do not necessarily have in depth understanding of the technology'. The participants opposed that statement completely, but also mentioned that it can be hard to have a proper understanding of the system that is built. In the context of AD, the tail of the events are so long and they need to have a complete description. It is hard but solvable.

The validation result in Figure 4.15 is separated from Figure 4.14, because there are only three participants validating those results. This due to time restrictions and participants availability.

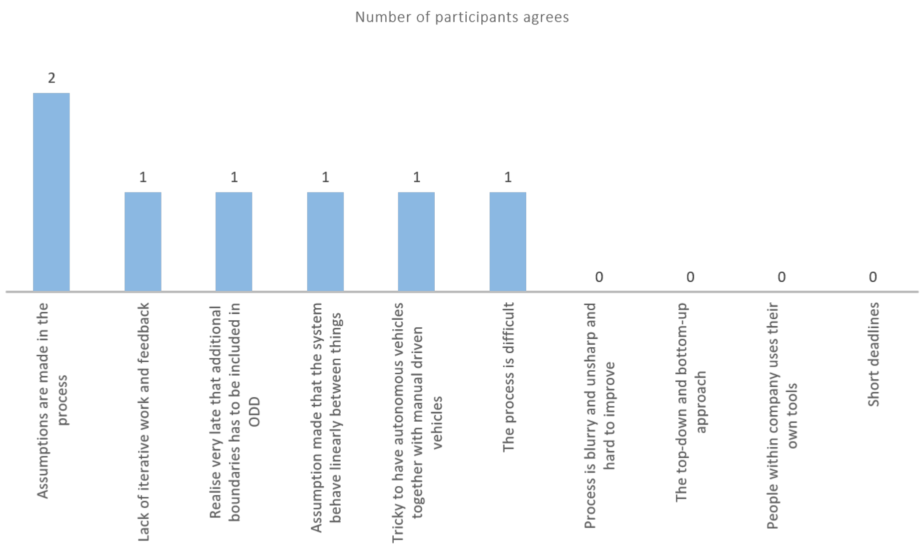

Figure 4.15: Validation result diagram for the challenges in Process, Responsibility, and Communication with three participants

The challenge 'Assumptions are made in the process' was confirmed by the two out of three participants.

The challenges 'Lack of iterative work and feedback', 'Realise very late that additional boundaries has to be included in ODD', 'Assumption made that the system behave linearly between things', 'Tricky to have autonomous vehicles together with manual driven vehicles', and 'The process is difficult' were only considered as challenges by one of the three participants. It was commented that the challenge 'Realise very late that additional boundaries have to be included in ODD' is a very good point to consider.

All the participants agreed that four of the challenges were not challenges. These are the following: 'Process is blurry and unsharp and hard to improve', 'The top-down and bottom-up approach', 'People within company uses their own tools', and 'Short deadlines'.

4.5.2 Validation of challenges with deriving requirements from use cases

Input regarding the challenges identified for 'Deriving requirements from use cases' was collected from the focus group. The focus group was asked the question 'What are the challenges with deriving requirements from use cases?'. Furthermore, they were asked to confirm the presented challenges as being relevant challenges. The result is shown in Figure 4.16 and Figure 4.17 and further explanation of the challenges is found in Section 4.4.2.

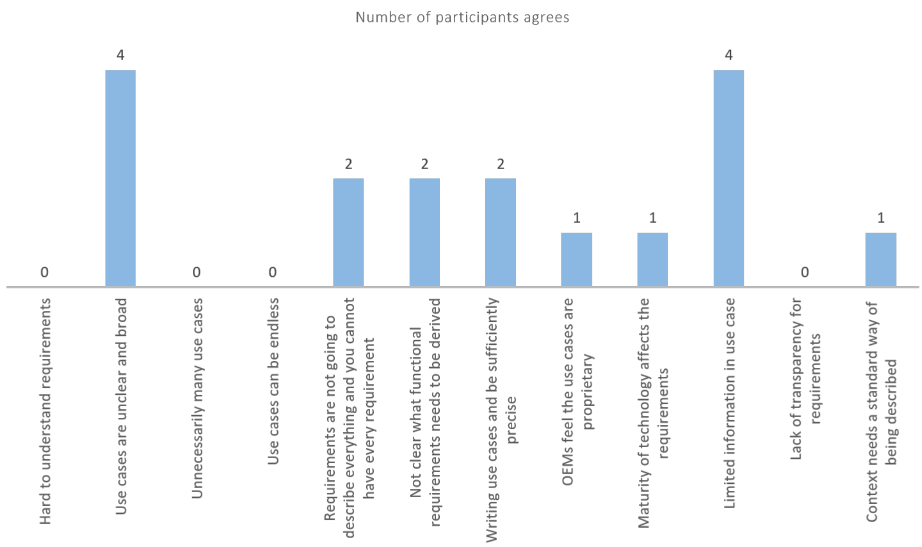

Figure 4.16: Validation result diagram of the challenges for deriving requirements from use cases with four participants

All participants agreed that the challenges 'Use cases are unclear and broad', 'Understanding use case', and 'Limited information in use case' were relevant. In regards to the challenge 'Use cases are unclear and broad' it was mentioned that use case is a good methodology to understand many requirements.

Half of the participants agreed that the challenges 'Requirements are not going to describe everything and you cannot have every requirement', 'Not clear what functional requirements needs to be derived', and 'Writing use cases and be sufficiently precise' were relevant. The motivation for 'Writing use cases and be sufficiently precise' being a challenge is that use cases are hard to write with good language. It is also hard to know what to include.

There were three challenges that only one participant agreed was a challenge. These are 'OEMs feel the use cases are proprietary', 'Maturity of technology affects the requirements', and 'Context needs a standard way of being described'.

All of the participants disagreed that four of the challenges were relevant challenges. These are 'Hard to understand requirements', 'Unnecessarily many use cases', 'Use cases can be endless', and 'Lack of transparency for requirements'. The motivation behind 'Unnecessarily many use cases' was that if you do not have many use cases, then you will not know what is required for the system.

The validation result in Figure 4.17 is separated from Figure 4.16, because there are only three participants validating those results. This due to time restrictions and participants availability.

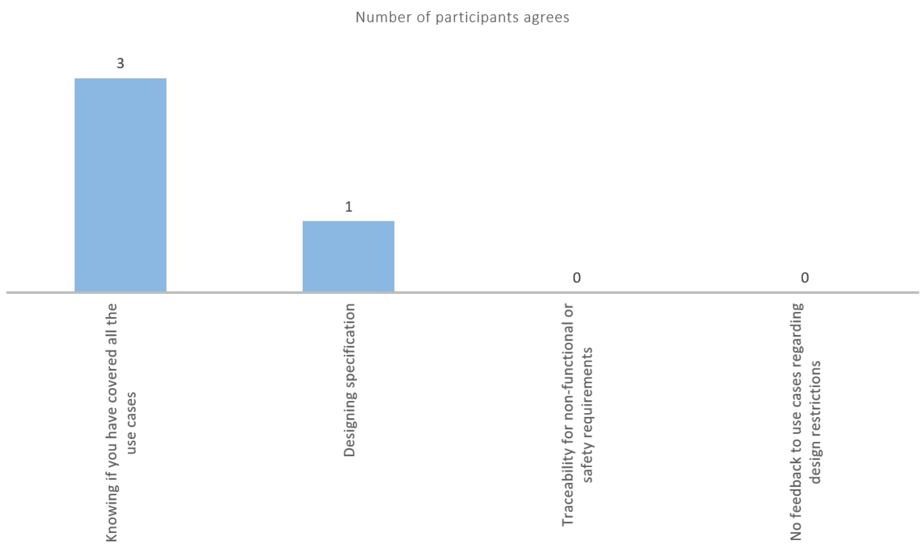

Figure 4.17: Validation result diagram of the challenges for deriving requirements from use cases with three participants

The challenge 'Knowing if you have covered all the use cases' was fully confirmed by the three participants. It was argued that the biggest challenge is to argue completeness. It was also stated that one problem is to know which use cases have a huge impact on the ODD. For example, by missing one use case, one might get a much wider ODD.

'Designing specification' was only considered by one participant to be a challenge for deriving requirements from use cases.

All of the participants disagreed that two of the challenges were relevant challenges. These are ‘Traceability for non-functional or safety requirements’ and ‘No feedback to use cases regarding design restrictions’.

4.5.3 Validation of the challenges with defining ODD

Input regarding the challenges identified for ‘Defining ODD’ was collected from the focus group. The focus group was asked the question ‘What are the challenges with defining ODD?’. Furthermore, they were asked to confirm the presented challenges as being relevant challenges. The result is shown in Figure 4.18 and Figure 4.19 and further explanation of the challenges is found in Section 4.4.3.

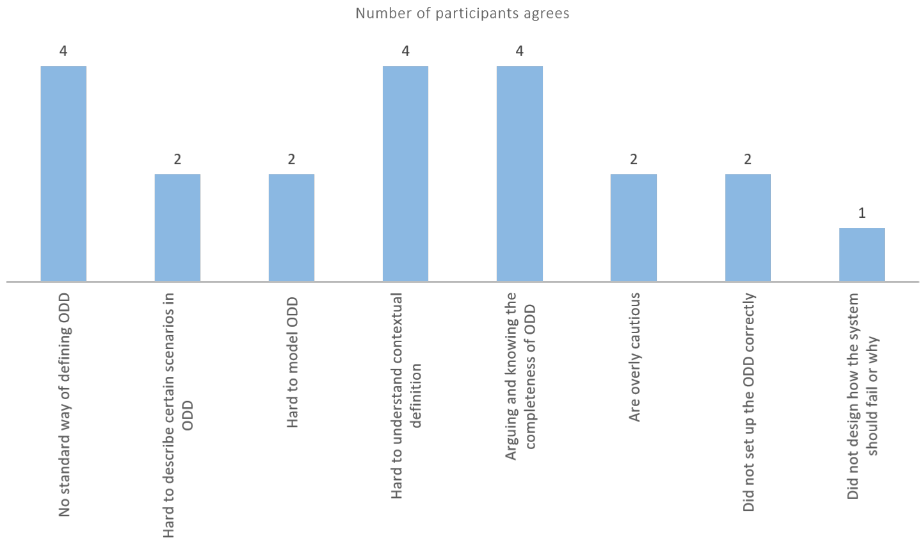

Figure 4.18: Validation result diagram of the challenges for defining ODD with four participants

Three challenges were fully confirmed, and these are ‘No standard way of defining ODD’, ‘Hard to understand contextual definition’, and ‘Arguing and knowing the completeness of ODD’.

The validation result is split about if ‘Hard to describe certain scenarios in ODD’, ‘Hard to model ODD’, ‘Are overly cautious’, and ‘Did not set up the ODD correctly’ are challenges for defining ODD.

One participant considered ‘Did not design how the system should fail or why’ as a challenge for defining ODD. It was discussed that this challenge depends on how well the process is set up and that it is known where it expects to fail and what part of the system should fail, but assumptions are made in regards to this. It is considered highly important to stay within the ODD.

The validation result in Figure 4.19 is separated from Figure 4.18, because there are only three participants validating those results. This due to time restrictions and participants availability.

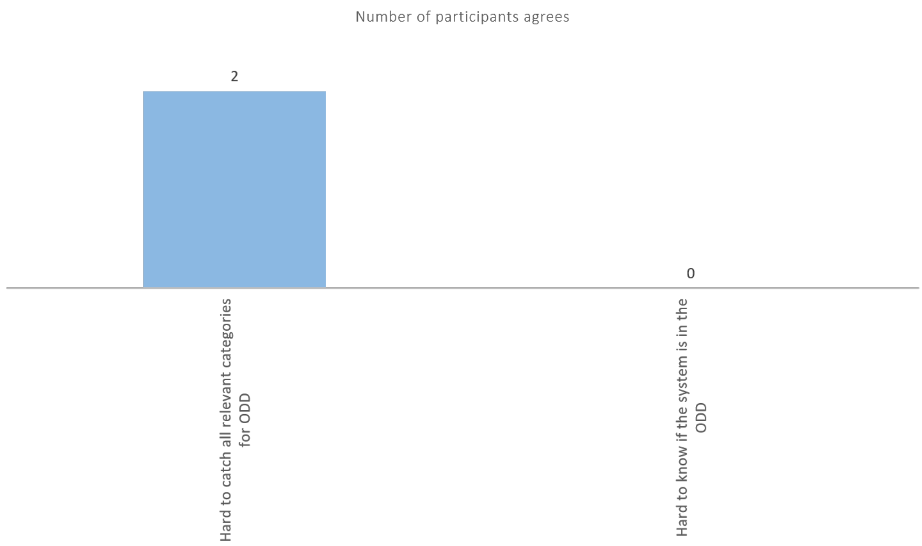

Figure 4.19: Validation result diagram of the challenges for defining ODD with three participants

Two out of three confirmed the challenge 'Hard to catch all relevant categories for ODD'. The challenge 'Hard to know if the system is in the ODD' was not confirmed to be a challenge for defining ODD.

4.5.4 Validation of the challenges with deriving context

Input regarding the challenges identified for 'Deriving context' was collected from the focus group. The question that the focus group was faced with was 'What are the challenges with deriving context?'. Furthermore, they were asked to confirm the presented challenges as being relevant challenges. The result is shown in Figure 4.20 and Figure 4.21 and further explanation of the challenges is found in Section 4.4.4.

All participants agreed that the challenges 'Arguing and knowing completeness of ODD' and 'Some parameters are hard to describe in context' were relevant. Three out of four people agreed that 'Understand what is important and what is necessary for the components success' was a relevant challenge. The focus group was split in their opinion regarding if the challenge 'Environment is dynamic and hard to predict' was relevant. It should be noted that one of the participants reported as disagreeing was swaying towards maybe agreeing with it being a challenge.

The validation result in Figure 4.21 is separated from Figure 4.20, because there are only three participants validating those results. This due to time restrictions and participants availability.

The challenge 'Input about context can be hard to handle or understand how it impacts ODD' was confirmed as being relevant. Two out of three agreed the challenges 'No good way to define the input about context' and 'Defining the context without assuming sensors' to be relevant. Only one participant confirmed 'Certain

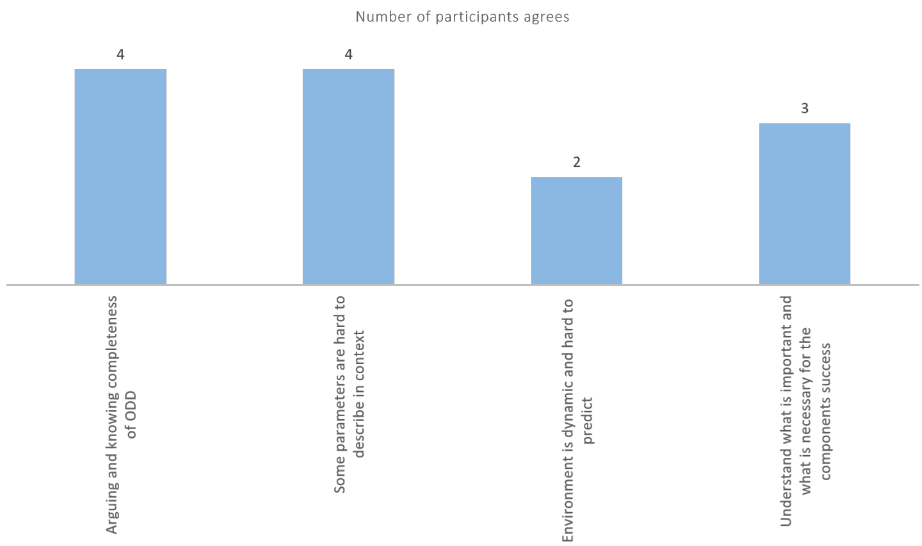

Figure 4.20: Validation result diagram of the challenges for deriving context with four participants

parameters, like weather, is hard to compare' as a relevant challenge. It was a complete disagreement that 'No view inside of the vehicle' and 'Communicate certain information about context to the manufacturer and driver' were challenges.

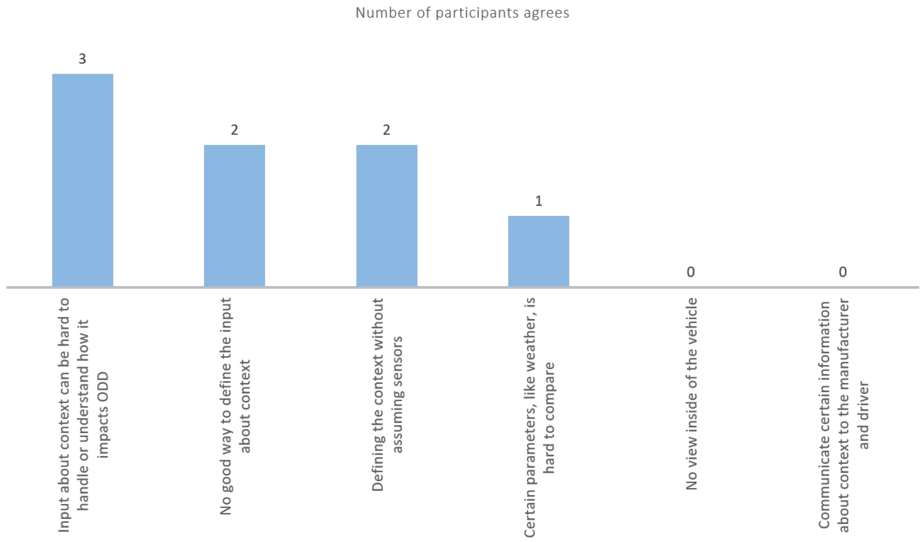

Figure 4.21: Validation result diagram of the challenges for deriving context with three participants

Area of Challenges	Solutions and Improvements
Process, Responsibility, and Communication	* A standard way of describing the context * Involve function developers more in defining the requirements and contextual definition * Communication * Continuous improvement * Different standards mindset can be useful in automotive * Structured and standardised approach * Create faster loops and derive requirements faster * Consider if the interpretation is the same between writer and receiver * Setting up Cross functionality * Prioritisation of requirements * Improve the leveraging from lessons learned or anticipating the issues * Early on agree on context and communicate it in a standardised way * Requirements engineering needs to be included into this new safe setup of developing functions * Continuous deliveries of new or updated set of requirements
Deriving Requirements from Use Cases	* Understand the implications of safety before designing system * Criteria to use to know when ODD and Requirements are complete * Make sure to cover all aspects of functionality, risk, and use case
Defining ODD	* Criteria to use to know when ODD and Requirements are complete * Structured and standardised approach * Automatic tool for deriving ODD * Information should be described clearly in the ODD * Need to define the non-operative domain * Measuring the exposure to a safety event
Deriving Context	* Knowing prerequisites * Collaboration between state/city/community and car/vehicle manufacturer * Good to have all sorts of sensors data * Broaden outside automotive industry
Validation	* More effective testing * Simulations
Standards and Regulations	* Consider aircrafts and ships approach * Different methods like risk analysis tools, fault tree analysis, FMEA

Table 4.4: Solutions and Improvements for Challenge Areas

4.6 RQ3: Which support would be appropriate for deriving contextual definition and requirements from use cases?

This section presents the results for the third research question (RQ3). The sections below are suggested improvements to address the challenges described in the previous Section 4.4. The improvements presented in Table 4.4 are to address the problem areas comprising the challenges, but are not limited to a single theme or challenge.

4.6.1 Improvements for process, responsibility, and communication

This section describes the themes related to process, responsibility, and communication, that were identified from the interviews. The solutions and improvements identified are presented in Figure 4.4.

A standard way of describing the context

An improvement for the process would be to have a standard way of describing context, specifically having some terminology for different parameters. This is a direct improvement to the challenge 'Context needs a standard way of being described' described in the Section 4.4.2.

Involve function developers more in defining the requirements and contextual definition

The function developers should be more involved in the requirements engineering process. To involve them more is a direct improvement to the challenge 'Function developers not enough involved in defining the requirements and contextual definition' stated in Section 4.4.1. The function developers bring a lot of useful experience in designing functions and working with the software. The process would not only benefit from utilising this knowledge, the function developers would also get a more complete picture on why requirements are created. This in turn would help when they start implementing the system.

Communication

An improvement requested from the interviewees was to improve and make sure the communication is good within the process. It is a cornerstone for success when developing a system or function. It is both about improving the communication between people and departments, but also about having good documentation regarding decisions made, adjustments made, and other noteworthy things that occur in the process. By having good traceability between requirements and use cases, it can be easier to make sure nothing is misaligned between artifacts that were developed during different stages in the process.

Continuous improvement

The field of autonomous drive is a pretty new field, and what is important is to continuously improve. The field and technology is evolving constantly, therefore it is important that the industries and processes adapt along with it. An interviewee mentioned that it will never be fixed, it is just about continuously improving and utilising the available resources in an effective way.

Different standards mindset can be useful in automotive

Current available standards and requirements are not directly applicable to the automotive industry. However, it was suggested that the mindset of the standards could still be helpful to apply towards the way of thinking and working within the automotive industry.

Structured and standardised approach

The need of a structured and standardised approach for defining context, ODD, and for the process of deriving ODD and requirements from use cases was clear across the interviews. Further suggestions for how to better define the context, ODD, and how to improve the process are presented in the Section 4.6.

Create faster loops and derive requirements faster

An improvement presented was to be able to derive good requirements faster, without losing quality. It was suggested that this can be done, if made assumptions can be quicker converted into real quantifiable measures. By having quantifiable measures, it will be possible to derive faster.

Consider if the interpretation is the same between writer and receiver

An interviewee brought up that a main challenge is to improve and consider if the interpretation is the same between the writer and receiver. Making sure that it is synchronised will further ensure that the development process artifacts are aligned with the previous stages and documents.

Setting up Cross functionality

A developer mentioned that setting up cross functionality would improve breaking down or transferring use cases to something real. It was suggested that it might help catch more of the issues early, and by involving some more roles they could together find the best way to break down requirements and make them realisable.

Prioritisation of requirements

A developer mentioned two main improvements: One of them was to make it easier and faster to prioritise requirements. This was described as something they go back and forth quite a bit about. The second improvement is to have transparency for requirements, which could help improving the prioritisation. By knowing the reasons behind the requirements, there could be less discussions and confusion, but they could still make a proper and informed prioritisation of the requirements.

Improve the leveraging from lessons learned or anticipating the issues

There is a need to continuously learn from the experiences and make sure not to lose the lessons learned. By improving on passing knowledge along and having proper documentation, it can help avoid repeating mistakes or oversee information. It is especially important to leverage the lessons learned from the first experiences with developing new autonomous systems.

Early on agree on context and communicate it in a standardised way

It was expressed that it would be really helpful to early on agree on the context, so this could be properly communicated out to everyone. By using a standardised vocabulary where the meaning of the words is clearly understood, everyone derives the same understanding of the parameters.

Requirements engineering needs to be included into this new safe setup of developing functions

In combination with the challenge ‘Function developers not enough involved in defining the requirements and contextual definition’ and solution ‘Involve the function developer more in defining the requirements and contextual definition’, in Section 4.4.1 respectively Section 4.6.1, it was brought up that requirements engineering needs to be included into this new safe setup of developing functions.

Continuous deliveries of new or updated set of requirements

An interviewee stated that requirements need to be continuously updated, and have continuous deliveries of updated requirements or new sets of requirements. The requirements also need to be updated together with the software development.

4.6.2 Improvements for deriving requirements from use cases

This section presents themes related to the improvements for deriving requirements from use cases.

Understand the implications of safety before designing system

An interviewee explained that it is important to understand the implications of safety from the first experiences with the development process by knowing the safety records’ current status. After that, one needs to start establishing priorities and finally design and define the system. This process, according to the interviewee, is still adapting to technology advancements, however the organisation needs to think about implementing and regulating that in an effective way.

Criteria to use to know when ODD and Requirements are complete

An interviewee suggested introducing an explicit method, and specific criteria to know when the requirements and ODD are complete and include all the sufficient information.

Make sure to cover all aspects of functionality, risk, and use case

It was described that it is important having structured and standardised methods to be applied and used, as well as consider trying to cover all the aspects of the functionality and the risk and the use case.

4.6.3 Improvements for defining ODD

This section describes the themes that were identified in relation to the improvements for addressing the challenges with defining ODD.

Criteria to use to know when ODD and Requirements are complete

One interviewee suggest as an improvement to have a more explicit method and using criteria to know when the ODD is complete and correct.

Structured and standardised approach

According to some interviewees, an improvement of the process would be to have a standardised and structured approach for defining ODDs. This will help to avoid misunderstandings and misinterpretations of what an ODD is. It would allow for a generalised process which is suitable for different use cases and contexts.

Automatic tool for deriving ODD

The improvement described is to have a tool that can take as input the context and automatically derive ODD. The context should have enough details so that the appropriate ODD can be derived. It was further argued that this is an efficient way because they have detailed context descriptions to derive appropriate ODD.

Information should be described clearly in the ODD

An improvement proposed by an interviewee is that the ODD should be clearly described with the information about context. This improvement helps in identifying the issues early on and communicating with the customer to decide whether it is acceptable or not.

Need to define the non-operative domain

It was stated that the organisations need to define the non-operative domain, so that they know where the system should fail and where it will not work. This will help to identify the issues early on and to develop safe systems in the end.

Measuring the exposure to a safety event

One interviewee explained that they are doing hazard and risk analysis (HARA) to figure out how the ODD should be defined from a safety viewpoint. The assumptions are made about the context and this information effects the HARA. For a safety relevant event, the exposure to a certain context can then be measured to decide on what should be defined in the ODD. This will help in the refinement of ODD and can reduce the amount of functionality needed for the system.

4.6.4 Improvements for deriving context

This section elaborates on themes related to the improvements identified for addressing the challenges with deriving context.

Knowing prerequisites

According to an interviewee, an improvement would be to know about what prerequisites are required in order for the system to function in certain environments. For example, in road construction scenarios, the prerequisites such as the need for specific signs or signals should be considered.

Collaboration between state/city/community and car/vehicle manufacturer

An interviewee suggested that the collaboration between state, city, or commune and the car or vehicle manufacturers, can in some way, improve quite a lot the development of autonomous vehicles.

Good to have all sorts of sensors data

It was argued that it is good to have all sorts of sensor data to figure out the system's exposure to various scenarios. The data can be very useful for the HARA. In order for it to be useful or quantifiable for HARA, one must be able to extract measurable things from the huge data sets.

Broaden outside automotive industry

An interviewee argued that it would be good if the way of describing context can be broadened to other applications outside of the automotive industry, because it will be much easier to get people to adapt to a standard way of describing the context.

4.6.5 Improvements for validation

This section describes the themes related to the improvements that were identified from the interviews for addressing the challenges with validation.

More effective testing

Instead of testing too many use cases, the interviewee stated that they want to be more effective. It was emphasized that an improvement would be the quicker identifying the key dimensional use cases, the more effective the process will be. This will save time and cost of testing while also allowing the team to focus on the system's most critical features.

Simulations

It was expressed that doing system level simulations to ensure fulfillment of use cases and without assuming any sensor processing performance, would be an improvement. It was further stated that simulation is needed a lot more for verification and validation. Because with real world testing, it is difficult to get statistical significance for the claims since the amount of data to prove that the ADS is better than human drivers is so large. The simulations also help to figure out a range of assumptions which must be tested and confirmed. In addition, it was stated that simulations help in measuring the severity of an impact on certain circumstances and also quantify the severity aspect of the HARA.

4.6.6 Improvements for standards and regulations

This section presents suggested improvements to address the challenges with standards and regulations.

Consider aircrafts and ships approach

Some interviewees expressed that approaches used by aircrafts and ship industries should be considered. It is considered that these industries are much more mature and experienced in the development of automated systems. For example, in their regulation the probability of catastrophic events is explicitly considered. It was further mentioned that they have a lot of commonly used techniques on how to ensure that you have the sufficient requirements and how to do the analysis. Those techniques can even be valid to use in the automotive industry and should be used in order to make a standardised and structured approach on how to break the use cases down, how to do the analysis, and how to extract the context and requirements from the use case. In addition, the aircrafts and ship industries are more experienced in considering what the prerequisites are from the surrounding and what is good enough for ensuring protective measures.

Different methods like risk analysis tools, fault tree analysis, FMEA

An interviewee thought that there are many different methods like different risk analysis tools, fault tree analysis and Failure mode and effect analysis (FMEA) that can be applied for breaking down the use cases into actual requirements and ODD requirements for prerequisites. It was mentioned that all of these methods and tools are used in the automotive industry. However, they are described as being generic because they are used in different fields of engineering.

4.7 Validation Result for Improvements

In this section, validation result will be presented for which improvements would be helpful or appropriate for a particular challenge area. The chapter is divided into four areas of improvements: 'Process, Responsibility, and Communication', 'Deriving requirements from use cases', 'Defining ODD', and 'Deriving context'. Also, for the area 'Process, Responsibility, and Communication', it will be presented which improvements the focus group thought would be the most helpful or be the most important solution for the challenge area.

There is no validation result for the improvements in the challenge areas 'Validation' and 'Standards and Regulations', because from the workshop these challenges were deemed less interesting and important for the result of the study.

4.7.1 Validation of the improvements for process, responsibility, and communication

Input regarding the improvements identified for 'Process, Responsibility, and Communication' was gathered from the focus group with four participants. They were asked to confirm which of the improvements presented are helpful improvements for 'Process, Responsibility, and Communication'. The result is shown in Figure 4.22 and further explanation of the improvements is found in Section 4.6.1.

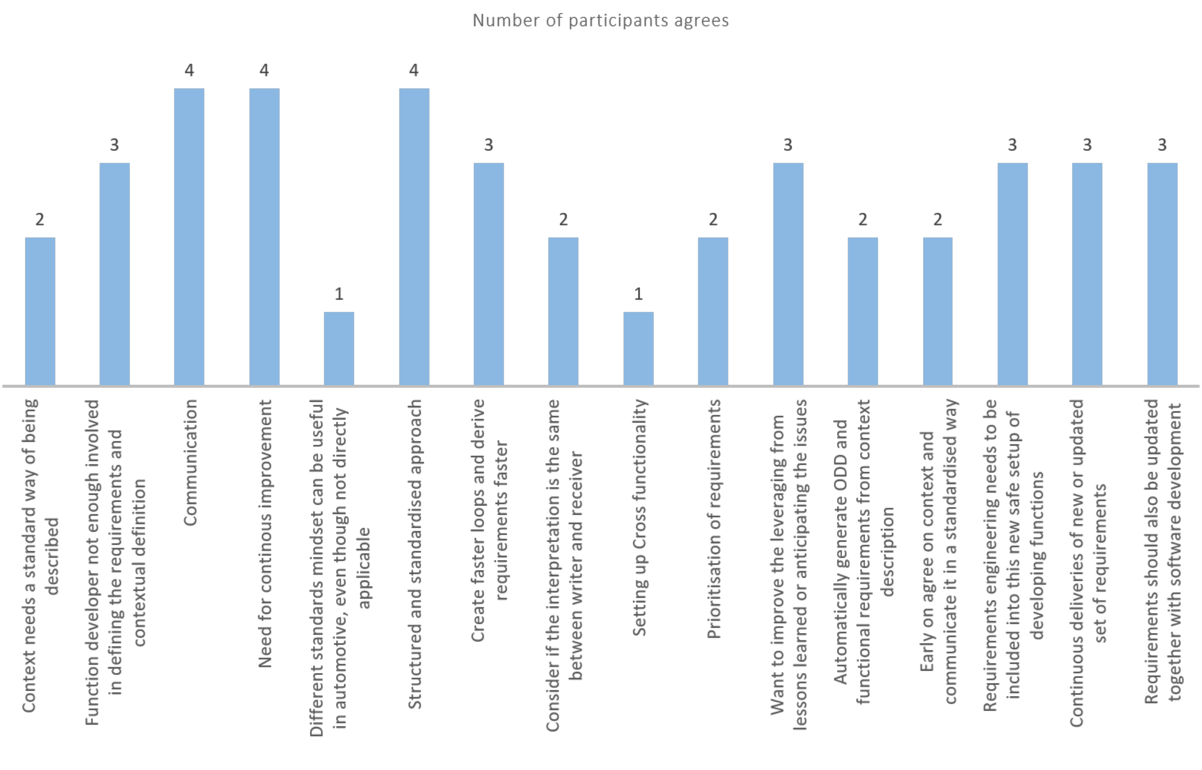

Figure 4.22: Validation result diagram of the improvements for Process, Responsibility, and Communication

Three of the improvements were fully confirmed. The improvements are ‘Communication’, ‘Need for continuous improvement’, and ‘Structured and standardised approach’. These are also all ranked in the top six most prominent improvements, see Appendix A.4 for the ranking result.

Three out of four found six improvements to be helpful for the area. These are ‘Involve function developers in defining the requirements and contextual definition‘, ‘Create faster loops and derive requirements faster‘, ‘Want to improve the leveraging from lessons learned or anticipating the issues‘, ‘Requirements engineering needs to be included into this new safe setup of developing functions‘, ‘Continuous deliveries of new or updated set of requirements‘, and ‘Requirements should also be updated together with software development‘.

There are five improvements that the focus group was divided about. These are ‘Context needs a standard way of being described‘, ‘Consider if the interpretation is the same between writer and receiver‘, ‘Prioritisation of requirements‘, ‘Automatically generate ODD and functional requirements from context description‘, and ‘Early on agree on context and communicate it in a standardised way‘.

There were two improvements that only one participant agreed would be helpful for the area and these are ‘Different standards mindset can be useful in automotive, even though not directly applicable’ and ‘Setting up Cross functional’.

Further the participants of the focus group were asked to rank the improvements, according to which was found most important. The five highest ranked improvements are shown in the list below, whilst the full ranking can be found in the Appendix A.4.

Top five ranked improvements:

1. Involve function developers in defining the requirements and contextual definition

2. Structured and standardised approach [1]

3. Continuous deliveries of new or updated set of requirements

4. Need for continuous improvement

5. Context needs a standard way of being described

[1]The improvement ’Structured and standardised approach’ has the same ranking score as the number one ranked improvement ’Involve function developers in defining the requirements and contextual definition’

4.7.2 Validation of the improvements for deriving requirements from use cases

Input regarding the improvements identified for 'Deriving requirements from use cases' was gathered from the focus group with four participants. They were asked to confirm which of the improvements presented are helpful improvements for 'Deriving requirements from use cases'. The result is shown in Figure 4.23 and further explanation of the improvements is found in Section 4.6.2.

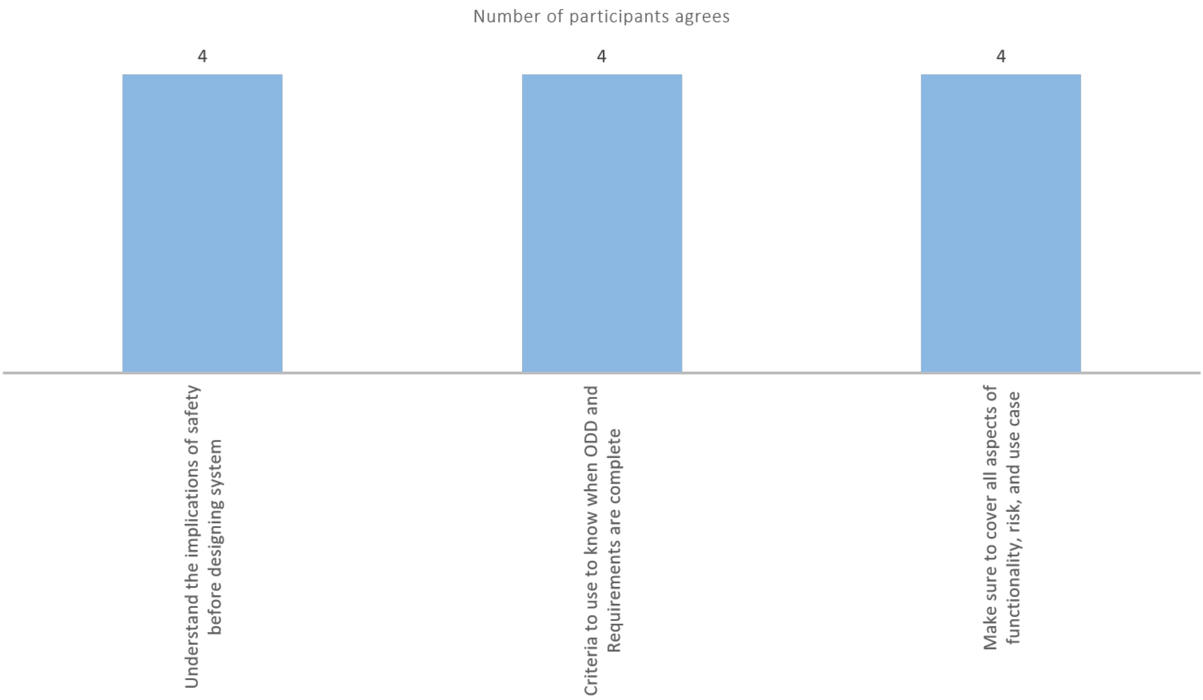

Figure 4.23: Validation result diagram of the improvements for deriving requirements from use cases

For 'Deriving requirements from use cases' all the improvements presented were confirmed. The improvements are 'Understand the implications of safety before designing system', 'Criteria to use to know when ODD and Requirements are complete', and 'Make sure to cover all aspects of functionality, risk, and use case'. The participants considered the improvements stated as goals to strive for.

4.7.3 Validation of the improvements for defining ODD

Input regarding the improvements identified for 'Defining ODD' was gathered from the focus group. They were asked to confirm which of the improvements presented are helpful improvements for 'Defining ODD'. The result is shown in Figure 4.24 and further explanation of the improvements is found in Section 4.6.3.

The improvements 'Criteria to use to know when ODD and Requirements are complete' and 'Automatic tool for deriving ODD' were confirmed by all the participants

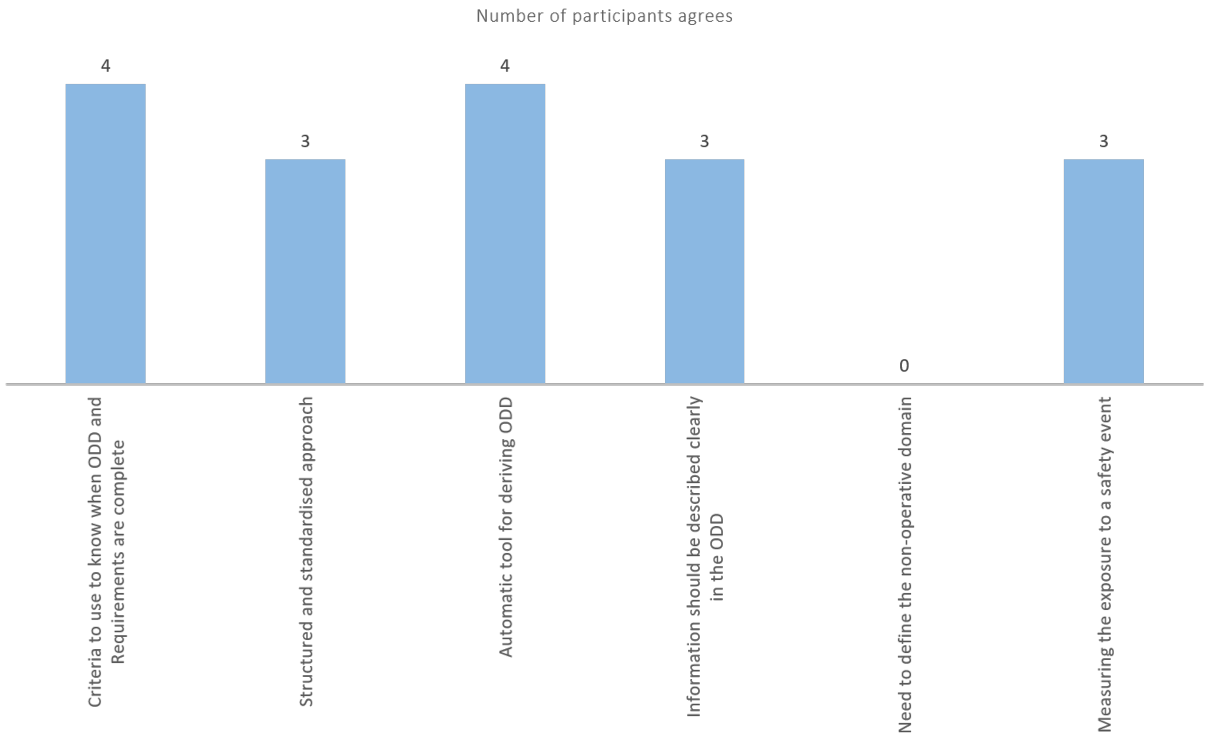

Figure 4.24: Validation result diagram of the improvements for defining ODD

of the focus group.

Three out of four agreed that the improvements 'Structured and standardised approach', 'Information should be described clearly in the ODD', and 'Measuring the exposure to a safety event' are helpful for defining ODD.

There was one improvement that everyone agreed would not help improve defining ODD, which was 'Need to define the non-operative domain'.

4.7.4 Validation of the improvements for deriving context

Input regarding the improvements identified for 'Deriving Context' was gathered from the focus group with three participants. They were asked to confirm which of the improvements presented are helpful improvements for 'Deriving Context'. The result is shown in Figure 4.25 and further explanation of the improvements is found in Section 4.6.4.

The participants confirmed that 'Good to have all sorts of sensors data' is an improvement that would be helpful for deriving context. Two out of three confirmed the improvement 'Knowing prerequisites'. There are two improvements that only one participant agreed would be helpful for the area and these are 'Collaboration between state/city/community and car/vehicle manufacturer' and 'Broaden outside automotive industry'.

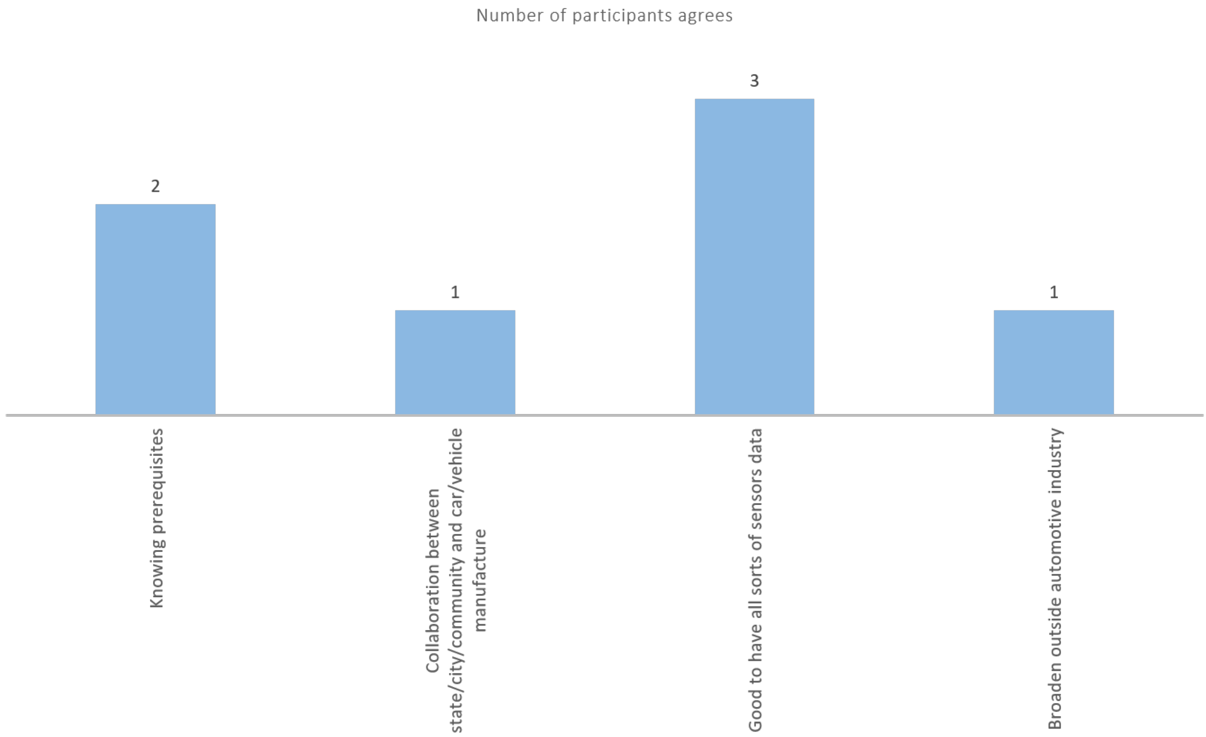

Figure 4.25: Validation result diagram of the improvements for deriving context

5 Discussion

The chapter begins with the discussions about the results of this study, which then is followed by discussions about implications of the results. The chapter ends with a section discussing threats to validity and ethical concerns.

5.1 RQ1: Which current solutions work well for deriving contextual definition and requirements from use cases?

For RQ1, nine different areas were identified to work well. These are described in Section 4.3. Next we will discuss a few of the interesting things that were brought up as working well.

Use cases are confirmed to work well as a communication tool. However, it should be noted that how good the use cases act as a communication tool is affected by how well defined the use cases are.

The iterative and feedback loop was confirmed as being a good approach, but a challenge was identified that there is a lack of iterative and feedback loop, see 'Lack of iterative work and feedback loop' in Section 4.4.1. It needs to be taken into account that not all companies have a good iterative process or feedback loop in place.

Half of the focus group confirmed that 'Breaking down into requirements' worked well. No challenges were identified that specifically opposed this. It should be considered that the opinion can be influenced by what tool and method are used to break down requirements, and also the complexity of the system that is being developed.

The results in this study oppose that the process of deriving contextual definition and requirements from use cases works well. Khastgir et al. [40] and Burton [36] clearly state that there are several challenges with the process. It should be considered that it also depends a lot on what processes and practices are in place. We concur with the literature and results.

The focus group does not agree that it is too early to tell what is working well in the process. We concur, based on literature and the study, and believe that the industry has already started to learn from the first experiences about what works and does not. Level 2 and level 3 systems are already on the market, which lessons can be learned from.

Identifying distinct properties in ODD is by majority considered as not working well, which correlates to the challenges. There are however suggestions given for categories of ODD by e.g., Thorn [2] and Czarnecki [27].

The validation is considered to not work well. Both Khastgir et.al. [40] and Jackson [39] support, in their papers, that this is hard because of the lack of a standardised approach. Specifically, Khastgir et al. mentions that it is difficult to define the pass and fail criteria, and also ensure completeness and consistency of the requirements [40].

5.2 RQ2: What are the challenges with deriving contextual definition and requirements from use cases?

This next part is going to discuss the results for RQ2, see Figure A.5 in the Appendix for all the challenges identified. We are going through four of the challenge areas identified: ‘Process, Responsibility, and Communication’, ’Deriving requirements from use cases’, ‘Defining ODD’, and ‘Deriving context’. In each of these areas we are going to present and discuss the challenges that we find most prominent to consider and solve. This will be based on the validation result, literature, and our gathered knowledge and understanding throughout this study. The prominent challenges are also divided into generic challenges and specific challenges i.e., that are closely related to the topic of the study. We are going to focus the discussion on the specific challenges, since we think these would be most relevant and promising for the field to address.

Challenges with process, responsibility, and communication

For the area ‘Process, Responsibility, and Communication’ we see nine challenges that are identified as main challenges to address. Three out of these we consider to be generic challenges, whilst the rest are specific to the topic of the study.

The three generic ones are: ‘Misunderstanding of requirements because interpreted by different people and roles ‘, ‘Communication ‘, and ‘Need good communication about requirements ‘.

The six specific ones are: ‘No common structured process’, ‘Needs input about context, but do not have, this goes to the ODD’, ‘Missing documenting assumptions ‘, ‘Assumptions are made in the process ‘, ‘Function developer not enough involved in defining the requirements and contextual definition ', and 'Unclear who is responsible for deriving context & requirements '.

The generic challenge 'Misunderstanding of requirements because interpreted by different people and roles ' is gathered as a main concern from the interviews, and the validation confirms. This can probably be an effect of not having a common terminology or definition of terms used. If proper communication or good practices are not in place to reduce misunderstandings, it can cause negative effects throughout the development process and even lead to failures in the system. These issues might not be noticed until late in the process.

The challenge 'No common structured process' is considered as a main challenge, since the validation results confirms that. From the interviews, it is evident that there exists different processes for different companies and teams, see Section 4.1.5. The lack of a common structured process leads to misinterpretations and confusion. It makes the development process less effective and requires more resources to be spent on verification and validation. Furthermore, the companies are not keen to collaborate and share their processes with each other, due to the competitive nature of the field, which makes it difficult to learn from one another.

The challenge 'Needs input about context, but do not have, this goes to the ODD' was reported as a main concern in the interviews and the validation group agrees with that. The input about context is needed in order to define the limitations in the ODD. When this input is not available, the ODD should define the missing input and state the limitations on where the system will work and where it will not. The challenge is to effectively derive the proper limitations, to put in the ODD, from the use case and context. This is considered a main challenge because if left out, the ODD will not be properly defined and can lead to unsafe operation.

According to the results, a lot of assumptions about the system's context and behavior were made during the development process. There might be a risk of misassumptions, if the information about the context is not clearly stated in the use cases. These kinds of assumptions can cause issues along the development and even result in an unsafe system. In addition, if the assumptions are missed and not documented properly, then it results in misalignment between documents and important parameters can be missed. The undocumented or missed assumptions can be challenging to handle later in the development process and we believe this is one of the main challenge to properly handle to get one step closer to a more effective process.

The specific challenge 'Function developers not enough involved in defining the requirements and contextual definition' half of the validation group agrees is a challenge. This might be due to the viewpoint of the participants, but the consequences and potential advantages are clear. The lack of involvement causes the process to be less effective because the function developers must investigate the reasoning behind requirements or make assumptions before implementation. According to other challenges identified, assumptions can be problematic and are already a challenge we want to address.

The specific challenge ‘Unclear who is responsible for deriving context & requirements’ was not considered predominantly by the focus group to be a challenge, but we want to highlight it as one of the main challenges to address. It was clear from the interviews that there is an uncertainty about who is responsible for deriving and maintaining context and requirements. Various suggestions of who is responsible was also put forth, which can be an effect of the interviewees having experience with different processes or that the process is unclear. This is likely, because we interviewed people from different companies, and we found that the processes varied.

Challenges with deriving requirements from use cases

We consider six challenges in the area ‘Deriving requirements from use cases’ to be more important to address. The generic challenges include ‘Limited information in use case’, ‘Writing use cases and be sufficiently precise’, and ‘Requirements are not going to describe everything and you cannot have every requirement’. The challenges that are considered to be specific to this study are ‘Knowing if you have covered all the use cases’, ‘Use cases are unclear and broad’, and ‘Not clear what functional requirements needs to be derived‘.

The challenge ‘Limited information in use case’ is tied in to the previously described challenge ‘Use cases are unclear and broad’. When the use cases are not having enough information, it leaves room for assumptions when making design decisions, which can cause problems in the later stages of the development process. The use cases should include necessary information for the system engineers to derive the requirements and context of the system. Furthermore, the use cases should be written in a language so that everyone can understand and it should not be too specific or vague. The use cases need to be encompassing enough for the developers [19].

The challenge ‘Knowing if you have covered all the use cases’ was mentioned as a main concern in the interviews and the validation. There is a risk of missing some use cases when the development team is unsure whether or not all use cases are covered. If the use cases are missed, then certain requirements will be missed, necessitating the refinement of the ODD as well as the inclusion of additional ODD. It will cause issues later on in the process and be more expensive to refine and adjust, if missed use cases are realised very late in the process.

The specific challenge ‘Use cases are unclear and broad’ was confirmed as a challenge. It is considered important because of what ripple effect it can have throughout the process, as stated in the result. The customer should be considered responsible for making clear in the use case what they want. It should be clear what use cases should describe and entail, so the people deriving requirements get the necessary information that they need. It was, also, further expressed that the use cases are not written properly i.e., the language and text are not sufficiently clear. All of this might be due to a lack of established and sufficiently good templates and practices for the use cases. The lack of practices and templates might cause the use case to be unclear and broad because the customers do not have the same expectations on what and how things should be written in the use case. The context that should be described in the use case is also quite complex and is found to be quite hard to write down in a clear and understandable way with enough details. This further can cause the customer to be unsure what to include in the use case. We believe sufficient and good templates, practices, and feedback would promote a more efficient process.

The specific challenge 'Not clear what functional requirements needs to be derived' was identified in a process where a trial-error approach was used. The focus of this approach is to get the minimal viable product out and then add upon it, this might explain why the challenge was mentioned. The nature of the systems developed are also new, with lack of previous experience. We believe also that this challenge can be an effect of the challenge 'Use cases are unclear and broad'. It makes it harder to know if the necessary functional requirements are derived and what exactly is needed from the use cases. It is also described from the interviews that the more use cases you have the more functional requirements need to be derived, since also stated that use cases can theoretically be endless, this becomes problematic for deriving functional requirements.

Challenges with defining ODD

As seen from the literature and result there are different ideas about the definition of ODD. We will present and discuss here how we view the definition of ODD. In our opinion the ODD is meant to clearly state the limitations of the operational environment. We agree with the J3016 definition[1], but since it is a recommended practice we believe a proper standard for the definition should be put in place.

We would like to also address the three additions to the definition that were requested by the interviewees. We believe road conditions are already included in the environmental or roadway characteristics. We consider the addition of the internal state of the vehicle to be a proper and good addition to the definition. It is not clear in the definition that internal state could be included. Knowing the internal state feels essential for systems which depend on conditions within the vehicle e.g., if handing over control of the vehicle to the user. The requirements should describe what happens when you are outside of the ODD. We believe this information does not belong in the ODD. Finally, it should be noted that the J3016 does not really exclude any of these additions, because it does not limit what the operational conditions could be. The term ODC is described by Khastgir [25] and we think this could potentially replace ODD further down the line, because it is an extended version of ODD.

Eight challenges in the area 'Defining ODD' are identified to be important to address. All of the challenges are considered to be specific to the study. The prominent challenges are 'No standard way of defining ODD', 'Hard to describe scenarios in ODD', 'Hard to catch all relevant categories for ODD', 'Did not set up the ODD correctly', 'Hard to understand contextual definition', 'Hard to model ODD', 'Arguing and knowing the completeness of ODD', and 'Are overly cautious'.

The study's result shows that there exists no standard way of defining ODD, which is in line with the literature [2]. We consider this also as a main underlying challenge, since a lot of other challenges found are tied to this.

The literature [41] confirms that it is challenging to identify certain scenarios and describe them in the ODD. In addition, the result and literature [2] show that it is hard to catch all the relevant categories and elements in the ODD. We believe that categories in ODD need to be identified as part of standardising the ODD definition, which also helps to make the process more generalisable and applicable. Additionally, a standard approach would ease the main concern 'Did not set up the ODD correctly' and the challenge 'Hard to understand contextual definition'.

The need for the ODD to be defined properly was supported by Khastgir [25]. Khastgir mentions that the ODD needs to be defined to determine the system's capabilities and limitations. We further believe a standard way of defining ODD could make it easier to model scenarios within the ODD, which is valuable for validation of the system and to ensure safety.

'Arguing and knowing the completeness of ODD' will also be eased by having a proper process for defining ODD. However, this challenge is also affected by how well you know and can argue that you have described the whole context in a proper way and all the necessary use cases are included. As discussed, the lack of certain use cases will have consequences throughout the process. Knowing and arguing the completeness of ODD does not go unaffected.

The focus group had divided opinions about if the 'Are overly cautious' was considered a challenge, however it was referred to as a main concern by an interviewee. Two viewpoints exist here. The first one being about being overly cautious when defining the ODD. Creating an overly strict ODD, will in turn restrict the capabilities of the system further than maybe necessary. The other viewpoint is that the overly cautious approach is based on the concern for proving safety, which might be a necessary concern in these beginning stages of development.

Challenges with deriving context

In the interviews, we found that some participants were unsure about the term 'Context' and some believed that context and ODD are the same. The participants' responses also reflect that the context is a broad term with different perspectives and understandings on what it describes and includes. Furthermore, in the literature [28, 29, 30], we found that there exists different definitions for context. We believe that context describes the whole environment or the surrounding where the system is operating. In our opinion, the terms 'Context' and 'ODD' are not the same. The term 'Context' describes the environment for a function or a behaviour, whilst the ODD defines the limitations that must be fulfilled by the environment and the system to achieve the desired behaviour. For example, the context for the ACC feature is that the ACC shall function on motorways, with good visibility, and speeds above 50 kilometers per hour, but not more than 130 kilometers per hour. The ODD is defined as the speed must be above 50, but below 130, camera must report clear visibility, brake actuators must function, and hands are on steering wheel. We believe that the results from this study about the definition for context and what it should include should be taken into consideration, when creating a specific definition for context.

In the area 'Deriving context', five challenges are considered as important to discuss and these challenges are more related to this study. The five challenges are 'Some parameters are hard to describe in context', 'No good way to define the input about context', 'Input about context can be hard to handle or understand how it impacts ODD', 'Environment is dynamic and hard to predict', and 'Defining the context without assuming sensors'.

The specific challenge 'Some parameters are hard to describe in context' is considered to be one of the important challenges. We believe this challenge is caused by the complexity of context and the difficulties to know what should be included in the context. In addition, the context is dynamic and it is hard to foresee all the conditions. The standardisation organisations and the industries are currently working on methods and approaches for identifying and describing various scenarios where the AV can operate.

The specific challenge 'No good way to define the input about context' impacts the challenge 'Input about context can be hard to handle or understand how it impacts ODD'. If it is hard or there is no good way to define the input about context, especially more complicated input, it will be harder to handle and know how the input affects the ODD i.e., how should the limitations be stated/written. A big concern is also how to handle input that cannot be retrieved and especially how that should be handled in the ODD.

The specific challenge 'Environment is dynamic and hard to predict' makes a lot of other challenges in the result harder e.g., the challenges 'No good way to define the input about context' and 'Arguing and knowing the completeness of ODD'. As an effect, if this challenge is not eased it will be harder to define use cases, which have consequences for how well we can derive requirements, define the ODD, and, as discussed, argue completeness and safety of the system.

The specific challenge 'Defining the context without assuming sensors' is important to address, because you want to set up a context that is not restricted by sensors' capabilities. Specifically, when doing simulations early on, we want to be sensor agnostic, so we are not already restricting the context. There are also other disadvantages to assuming sensors early on, like you might opt for sensors that are not an optimal fit for the system. This can also restrict the customer needs or requirements early on.

5.3 RQ3: Which support would be appropriate for deriving contextual definition and requirements from use case

This next part is going to discuss the results for RQ3, see Figure 4.4 for all the improvements identified. We will suggest and discuss the most prominent improvements from our result. This will be based on the validation result, literature, and our gathered knowledge and understanding throughout this study.

Improvements for process, responsibility, and communication

We consider four improvements as prominent improvements to implement for the area of 'Process, Responsibility, and Communication'. The improvements are 'Structured and standardised approach', 'Involve the function developer more in defining the requirements and contextual definition', 'Continuous deliveries of new or updated set of requirements', and 'Continuous improvement'.

It is clear from literature [2, 40, 26] and the result that the improvement 'Structured and standardised approach' is needed for both the overall process, but also for individual parts. There are several identified main challenges that concern the lack of standard approaches e.g., for deriving ODD, defining context et cetera. The result in this study can aid in creating a standard approach. The main challenges should be considered, and the overall result in regards to the view of the concepts and processes. Standardised approaches could reduce the time to market for developed systems and help with the assessment of safety for the automated vehicles. However, we believe it might be hard to implement a standard in an efficient way across organisations, since the current structures within organisations can look vastly different.

The improvement 'Involve function developers more in defining the requirements and contextual definition' is a direct solution to the challenge 'Function developers not enough involved in defining the requirements and contextual definition', which was considered one of the main challenges to address. The improvement 'Continuous deliveries of new or updated set of requirements' will be helpful for getting feedback, having a discussion early on about requirements, and being made aware of issues and misinterpretations earlier on. Also, it will give an opportunity to continuously refine, for example the ODD, according to the feedback. The need for 'Continuous improvement' is something that we believe needs to be considered and is a necessity to uphold.

Improvements for deriving requirements from use cases

We consider three improvements as prominent goals to strive for in the area of 'Derive requirements from use cases'. The goals are 'Understand the implications of safety before designing system', 'Criteria to use to know when ODD and Requirements are complete', and 'Make sure to cover all aspects of functionality, risk, and use case'.

The goal ‘Understand the implications of safety before designing system’ should be strived for. Lessons learned should be utilised and considered for future development. The implications of safety can be learned throughout development, but mostly obtained from simulation and testing.

The goal ‘Criteria to use to know when ODD and Requirements are complete’ should be strived for. The sufficient and complete information in the ODD and requirements will ease the work of implementation of the system for the function developers. It will further ease testing of the system. We recommend that a standard template or framework could be implemented to clarify when the ODD and requirements are considered complete.

By striving for the goal ‘Make sure to cover all aspects of functionality, risk, and use case’ it will help to describe the needed use cases, which will have positive effects throughout the process. Ensuring that all use cases are covered will ease validation. The use case template suggested by Supriya [21] can be utilised to ensure that necessary information is covered. We further suggest investing in simulations to make sure that you have covered all, or a sufficient amount of, the use cases.

Improvements for defining ODD

We consider five improvements to be important to improve the area ‘Defining ODD’. The five improvements are ‘Criteria to use to know when ODD and Requirements are complete’, ‘Structured and standardised approach’, ‘Automatic tool for deriving ODD’, ‘Information should be described clearly in the ODD’, and ‘Measuring the exposure to a safety event’.

The improvement ‘Criteria to use to know when ODD and Requirements are complete’, has been previously discussed. We consider that the improvement ‘Structured and standardised approach’ is the most vital one to define the ODD. The standardised approach for defining ODD is essential to identify and organise the various ODD elements. We believe that the goal is to rely on the established taxonomies and ontologies for developing a structured and standardised approach for defining ODD. For example, Thorn [2] provides a taxonomy and a checklist for ODD.

We think that the improvement ’Automatic tool for deriving ODD’ will be challenging to achieve, but it will be useful and will make the process of deriving ODD more efficient. Another improvement which was discussed in [41], is that scenarios can be described using a two level scenario description language. The SDL serves as a common language to define the scenarios in a good way so that stakeholders can understand. This can be utilised to further ease the challenge ‘Hard to describe scenarios in the ODD’.

Furthermore, we suggest that a role or team should be designated for deriving and maintaining ODD. This can further ensure that the appropriate ODD is derived and changes to the ODD are effectively communicated to other concerned teams.

In the result, see Table 4.2, you can see that different roles were suggested as being responsible for deriving and maintaining ODD.

Improvements for deriving context

We think that the improvement 'Good to have all sort of sensors data' is important for the area 'Deriving Context' since the validation result confirms that. In the study by Damak [34], the proposed operational context ontology can be considered when defining context, since it helps in identifying various context elements for the system. We suggest further that a role should be designated for deriving context. In the result for context, see Table 4.3, you can see that different roles were suggested as being responsible for defining the context.

5.4 Implications to Researchers

We believe the result of this study provides further information that fills in some of the knowledge gaps within the research field, both specifically for the automotive but also for other domains. It provides knowledge and identifies gaps about the cornerstones of the process of deriving contextual definition and requirements from use cases.

The result of the concepts e.g., ODD and context, shines light on the believed definition and the confusions or challenges with these concepts. The relationships identified between the concepts e.g., the relationship between context and ODD, provides helpful insight. This is not specifically useful for research within automotive, but also for other domains that deal with context and limiting the operational domain. We further believe specifically that this study puts the definition of context more clearly on the research map for requirements engineers. The results about the different processes could be an interesting basis for the researchers to investigate further about which ones could be beneficial for developing context-aware systems.

Further, we have identified challenges that could be of interest to solve for four different areas: process, responsibility, and communication; deriving requirements from use cases; defining ODD; and deriving context. The discussion and the improvements provided, can serve as a helpful start to develop concrete solutions for the challenges identified.

5.5 Implications to Practitioners

We believe that the results of this study will serve as a conceptual ground for discussing requirements in relation to context. Also, the results serve as a base for defining ODD in a systematic and integrated way, and will be instrumental to build the automotive systems of tomorrow.

The identified challenges are viewed as roadblocks to more efficient development; if these could be overcome, progress could be made and we would be one step closer to a proper process for deriving contextual definition and requirements from use cases.

The industry can use the results of this study as a base for consideration when creating processes, practices, or templates for a common definition of ODD and context. The results can be utilised to define the concepts like ODD and context, in a common way within the company. It would promote efficient communication, because everyone has a clear and common definition of the concepts. Also, the results can be considered when creating a standard process of deriving context, ODD and requirements from use cases and for ensuring the completeness of ODD and requirements. The suggested improvements in this study should be taken into account when determining what practices should be implemented to address the identified challenges.

5.6 Threats to Validity

This section discusses the possible validity threats in this study. The discussion will be based on the four aspects of validity, described by Runeson et.al. [46]. The four aspects of validity described are: Internal validity, Construct validity, External validity, and Reliability. This section also discusses some of the ethical concerns and how this study strives to comply with ethics.

Internal validity

Internal validity is concerned about if a third factor is affecting the factor that is being investigated. In our case, the concern regards the validity of the result and if any third factor has affected the collected data from the interviews. There is a risk, when collecting and analysing data, for researcher's bias. In this study, it was mitigated by having discussions and a workshop with the supervisors about the findings.

Another risk in this study for sampling the interviewees, when the case company is providing most of the participants, is two-fold. One is that the study is being presented only with participants that are foremost available, but are not the optimal to interview for the study. The second risk is that the ones providing the participants are providing those that correlate with their interests in the study i.e., what they are interested in finding. This was mitigated in this study by having a proper sampling strategy created by the researchers that the case company followed. Also, all the participants were not picked or provided by the case company. A couple of the participants were contacted through the academic supervisors' connections. There was also an attempt to contact other various companies by the researchers, but no responses were received.

There are some potential threats in regards to performing remote interviews, like interview environment, lack of face-to-face interaction, and bad internet connection.

The interview environment threat refers to that the researchers have no control over what environment the interviewee is within while performing the interview. A noisy, hectic, or interruptive environment could potentially affect the interviewee's answers. Bad internet connection might cause interruptions in the interview and make it harder to understand each other. This was mitigated as much as possible by the researcher's by testing the internet connection on their end before the interview started, and informing the interviewee about actions that should be taken if interruptions occurred. The interviews were otherwise kept to a maximum of one hour in order to avoid the interviewee feeling tired.

Construct validity

Construct validity refers to the extent to which the study measures represent the researcher's intentions and what it claims to address for the research questions. This was addressed by sending the purpose of the study and the goal of the session to the participants prior to each interview and focus group. It was also explained to the participants at the start of each interview to avoid any confusion. During the interviews, the participants were encouraged to ask questions that might come to mind. Also when needed, the questions were clarified and examples were given, to give the participants a better idea of the intent of the questions. The interview guide was designed in such a way that the first section of questions establishes the participant's understanding of the ground concepts of the process. This avoids misinterpretations and ensures non-ambiguity. Furthermore, the researchers of this study have a lack of working experience with systems like ADAS and ADS. This was compensated by utilising both the academic and industry supervisors' experience and knowledge, as well as a bit of literature review when designing the interview guide.

External validity

External validity is concerned about how generalisable the result is. For this study, we are specifically discussing if the result is applicable and useful to other fields outside of the automotive domain. The study is, of course, interesting for development of ADS and ADAS in the automotive industry, but we argue that the result is helpful for development of other context-aware systems and functions, even outside of the automotive domain. The sampling helps with making the result further generalisable within the automotive field by gathering results from different companies, different parts of the industry, and different roles on different levels of the process. The interviewees are from the automotive industry, but the interview questions asked are created with the intent of being non-specific to the automotive field. Even though the term ODD is specific to the automotive industry, the concept of limitations on the context exists outside of the automotive domain. This makes us further believe that the results are applicable and helpful for other fields that deal with development of context-aware systems and functions.

Reliability

Reliability is concerned with replicability of the study and consistency of the results across different researchers and different projects. The reliability was ensured by the detailed description presented for the sampling strategy and the research method. The interview participants' experience and roles were also presented in the preparation for data collection section, see Section 3.2. In addition, the interview guide was presented in the Appendix A.1. The reliability was further improved by the validation of the results with the participants during the focus group, although not all of the results were validated.

Informed consent

A consent form was sent out to the interview participants before the interview. The interview participants were, through the consent form, informed about the purpose of the study, how data collected would be handled, when and what would be deleted after completion of the work, and promised confidentiality and anonymity. The consent form informed and asked for consent regarding recording the interview. They were also asked again for consent verbally right before the interview started. The focus group participants were informed about the purpose of the session, how their feedback would be handled, and that only notes would be taken during the session.

Confidentiality and Anonymity

The researchers, before the study started, signed a confidentiality form with the company. The researchers agreed to not disclose any company related information. The interview participants received a consent form that was signed, which informed the participants about the study, how the data would be handled, and promised confidentiality and anonymity.

6
Conclusion

Based on the qualitative research, this study is conducted in the context of the automotive field by collecting data from the automotive experts through interviews and then the thematic analysis is carried out and finally the results are validated.

In this study, the current working process of deriving contextual definition and requirements from use cases is investigated to learn how the different concepts and terms within the process have been interpreted and used in the industry and research field. Also, the study investigated what is currently working well within the process. The study clarifies the definition of different terms like use case, context, and ODD and how these terms are related. The challenges with deriving contextual definitions and requirements from use cases are identified. The challenges were further classified based on the areas of the process and also the causes for those challenges were discussed. In addition, potential suggestions of improvements to solve the challenges for each area are provided.

The study shows that there is a lack of a common process for deriving contextual definition and requirements from use cases, and a lack of standard definition for concepts. The challenges mapped out in this study can be utilised to improve the process, to further ensure more efficient development.

This study, therefore, recommended future work on an improved practice for deriving contextual definition and requirements from use cases by considering the challenges and the suggested improvements. Also, the usefulness or how to best utilise the improvements could be further investigated. The future work could also be focused on developing a common or standard process for deriving contextual definition and requirements from use cases. The study can serve as a baseline for the research in establishing standard definitions of context and ODD. The relationship between context, use cases and requirements can be interesting to consider for research and development of context-based approaches in relation to requirements engineering. The study is only performed with a limited amount of participants, therefore further research could also be performed with other companies that this study did not have the opportunity to interview. Furthermore, the requirements engineering community could consider the result of this study to investigate the importance and need of context for AI systems.

Bibliography

[1] SAE, "J3016B Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," Tech. Rep., 2018. [Online]. Available: https://www.sae.org/standards/content/j3016_201806/

[2] E. Thorn, S. Kimmel, and M. Chaka, "A Framework for Automated Driving System Testable Cases and Scenarios," *Dot Hs 812 623*, no. September, p. 180, 2018. [Online]. Available: https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/13882-automateddrivingsystems_092618_v1a_tag.pdf

[3] A. Shaout, D. Colella, and S. Awad, "Advanced driver assistance systems - Past, present and future," *ICENCO'2011 - 2011 7th International Computer Engineering Conference: Today Information Society What's Next?*, pp. 72–82, 2011.

[4] Veoneer, "Advanced Driver Assistance Systems." [Online]. Available: https://www.veoneer.com/en/advanced-driver-assistance-systems

[5] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a Formal Model of Safe and Scalable Self-driving Cars," *arXiv*, pp. 1–37, 2017.

[6] A. Reschka, J. R. Böhmer, T. Nothdurft, P. Hecker, B. Lichte, and M. Maurer, "A surveillance and safety system based on performance criteria and functional degradation for an autonomous vehicle," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 237–242, 2012.

[7] A. Soultana, F. Benabbou, and N. Sael, "Context-awareness in the smart car," in *Proceedings of the 4th International Conference on Smart City Applications - SCA '19*. New York, New York, USA: ACM Press, 2019, pp. 1–8. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3368756.3369019

[8] I. Colwell, B. Phan, S. Saleem, R. Salay, and K. Czarnecki, "An Automated Vehicle Safety Concept Based on Runtime Restriction of the Operational Design Domain," *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2018-June, no. Iv, pp. 1910–1917, 2018.

[9] A. Knauss, D. Damian, and K. Schneider, “Eliciting contextual requirements at design time: A case study,” in *2014 IEEE 4th International Workshop on Empirical Requirements Engineering (EmpiRE).* IEEE, aug 2014, pp. 56–63. [Online]. Available: http://ieeexplore.ieee.org/document/6890117/

[10] M. Gyllenhammar, R. Johansson, F. Warg, D. Chen, H.-M. Heyn, M. Sanfridson, J. Söderberg, A. Thorsén, S. Ursing, Z. Ab, and M. G. Com, “Towards an Operational Design Domain That Supports the Safety Argumentation of an Automated Driving System,” *10th European Congress on Embedded Real Time Systems*, pp. 1–10, 2020. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1390550/FULLTEXT01.pdf

[11] NHTSA, “Automated Driving Systems: a vision for safety,” 2017.

[12] P. Koopman and F. Fratrik, “How many operational design domains, objects, and events?” in *CEUR Workshop Proceedings*, vol. 2301, 2019.

[13] T. R. Austin, “Letter to Dr. Steven E.Shladover.” [Online]. Available: www.nspe.org/sites/default/files/resources/pdfs/ShladoverLtr-2016-Feb5-FINAL.pdf

[14] ADASME, “Use Cases.” [Online]. Available: https://www.adasandme.com/about-adasme/use-cases/

[15] A. Knauss, D. Damian, X. Franch, A. Rook, H. A. Múller, and A. Thomo, “Acon: A learning-based approach to deal with uncertainty in contextual requirements at runtime,” *Information and Software Technology*, vol. 70, pp. 85–99, 2016.

[16] A. Knauss, “The Capture and Evolution of Contextual Requirements: The Case of Adaptive Systems,” Ph.D. dissertation, University of Victoria, Canada, 2015.

[17] K. Kelly and B. Bowe, “Qualitative Research Methods in Engineering,” pp. 22.1207.1—-22.1207.10, 2020.

[18] C. N. Creswell, John W.; Poth, *Qualitative Inquiry and Research Design: Choosing Among Five Approaches*, 4th ed. SAGE Publishing, 2017.

[19] A. Cockburn, “Witing effective use cases,” Tech. Rep., 2000.

[20] W. Wachenfeld, H. Winner, J. C. Gerdes, B. Lenz, M. Maurer, S. Beiker, E. Fraedrich, and T. Winkle, “Use Cases for Autonomous Driving,” in *Autonomous Driving.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 9–37. [Online]. Available: http://link.springer.com/10.1007/978-3-662-48847-8_2

[21] S. Supriya, "Engineering Requirements for the Development of Cloud Based Support for Autonomous Trucks," Ph.D. dissertation.

[22] P. Inverardi and M. Mori, "Requirements models at run-time to support consistent system evolutions," in *2011 2nd International Workshop on Requirements@Run.Time*. IEEE, aug 2011, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/document/6046241/

[23] R. Ali, F. Dalpiaz, and P. Giorgini, "A goal-based framework for contextual requirements modeling and analysis," *Requirements Engineering*, vol. 15, no. 4, pp. 439–458, nov 2010. [Online]. Available: http://link.springer.com/10.1007/s00766-010-0110-z

[24] A. J. Ramirez, A. C. Jensen, and B. H. C. Cheng, "A taxonomy of uncertainty for dynamically adaptive systems," in *2012 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE, jun 2012, pp. 99–108. [Online]. Available: http://ieeexplore.ieee.org/document/6224396/

[25] Siddartha Khastgir, "The Curious Case of Operational Design Domain: What it is and is not?" 2020. [Online]. Available: https://medium.com/@siddkhastgir/the-curious-case-of-operational-design-domain-what-it-is-and-is-not-e0180b92a3ae

[26] "Best Practice for Describing an Operational Design Domain: Conceptual Framework and Lexicon." [Online]. Available: https://avsc.sae-itc.org/principles-02-5471WV-44074RU.html?respondentID=25427355#Read-More

[27] K. Czarnecki, "Operational Design Domain for Automated Driving Systems - Taxonomy of Basic Terms," 2018.

[28] A. Zimmermann, A. Lorenz, and R. Oppermann, "An Operational Definition of Context," in *Modeling and Using Context*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4635 LNAI, pp. 558–571. [Online]. Available: http://link.springer.com/10.1007/978-3-540-74255-5_42

[29] Y. Nemoto, K. Uei, K. Sato, and Y. Shimomura, "A Context-based Requirements Analysis Method for PSS Design," *Procedia CIRP*, vol. 30, pp. 42–47, 2015. [Online]. Available: http://doi.org/10.1016/j.procir.2015.02.095

[30] A. K. Dey, "Understanding and using context," *Personal and ubiquitous computing*, pp. 4–7, 2001. [Online]. Available: http://dl.acm.org/citation.cfm?id=593572

[31] P. J. Brown, "The stick-e document : a framework for creating context-aware applications," Tech. Rep. June, 1996.

[32] K. Henricksen and J. Indulska, "A software engineering framework for context-aware pervasive computing," in *Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the.* IEEE, 2004, pp. 77–86. [Online]. Available: http://ieeexplore.ieee.org/document/1276847/

[33] H. CHEN, T. FININ, and A. JOSHI, "An ontology for context-aware pervasive computing environments," *The Knowledge Engineering Review*, vol. 18, no. 3, pp. 197–207, sep 2003. [Online]. Available: https://www.cambridge.org/core/product/identifier/S0269888904000025/type/journal_article

[34] Y. Damak, "Operational Context-Based Design and Architecting of Autonomous Vehicles," Ph.D. dissertation, Université Paris-Saclay, 2020.

[35] PEGASUS, "PEGASUS Method an Overview," Tech. Rep., 2020. [Online]. Available: https://www.pegasusprojekt.de/files/tmpl/pdf/PEGASUS_Abschlussbericht_Gesamtprojekt.PDF

[36] S. Burton, "Bounding the open context. Specifying safety requirements for automated driving," 2020. [Online]. Available: https://www.york.ac.uk/assuring-autonomy/news/blog/safety-highly-automated-driving-open-context/

[37] R. Roosien and T. van Birgelen, "Introducing automation in aviation," Tech. Rep. December, 2017.

[38] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A Survey of Autonomous Driving: <i>Common Practices and Emerging Technologies</i>," *IEEE Access*, vol. 8, pp. 58 443–58 469, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9046805/

[39] D. Jackson, "Autonomous Driving How to overcome the 5 main technology challenges," Tech. Rep., 2020. [Online]. Available: https://capgemini-engineering.com/nl/en/insight/autonomous-driving-overcome-5-main-technology-challenges/

[40] S. Khastgir, S. Birrell, G. Dhadyalla, and P. Jennings, "The Science of Testing: An Automotive Perspective," in *SAE Technical Papers*, vol. 2018-April, apr 2018, pp. 10–12. [Online]. Available: https://www.sae.org/content/2018-01-1070/

[41] X. Zhang, S. Khastgir, and P. Jennings, "Scenario Description Language for Automated Driving Systems: A Two Level Abstraction Approach," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, vol. 2020-Octob. IEEE, oct 2020, pp. 973–980. [Online]. Available: https://ieeexplore.ieee.org/document/9283417/

[42] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 4th ed. CA: SAGE Publications: SAGEPublications, 2014.

[43] B. Hancock, "An Introduction to Qualitative Research Au t hors," *Qualitative Research*, vol. 4th, p. 504, 2006. [Online]. Available: http://books.google.cz/books?id=sFv1oWX2DoEC

[44] J. W. Creswell, *Educational research : planning, conducting, and evaluating quantitative and qualitative research*, 4th ed. Pearson Education, Inc., 501 Boylston Street, Boston, MA 02116., 2012.

[45] C. Marshall and G. B. Rossman, *Designing Qualitative Research*, 6th ed. Thousand Oaks, California: SAGE Publishing, 2015.

[46] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, apr 2009. [Online]. Available: http://link.springer.com/10.1007/s10664-008-9102-8

[47] T. C. Lethbridge, S. E. Sim, and J. Singer, *Studying software engineers: Data collection techniques for software field studies*, 2005, vol. 10, no. 3.

[48] S. Brinkmann and S. Kvale, *InterViews: Learning the Craft of Qualitative Research Interviewing*, 3rd ed. SAGE Publishing, Thousand Oaks, CA, 2015.

[49] K. Robson, Colin ; McCartan, *Real world research*, 4th ed., 2016.

[50] J. Saldaña, *The coding manual for qualitative researchers*, 2nd ed., J. Seaman, Ed. SAGE Publishing, 2013.

[51] Ann Lewins and Christina Silver, *Using Software in Qualitative Research: A Step-by-Step Guide.* London: Sage Publications, 2007.

[52] "Operational Design Domain (ODD) taxonomy for an automated driving system (ADS)-Specification." [Online]. Available: https://www.bsigroup.com/en-GB/CAV/pas-1883/

A

Appendix 1

A.1 Interview Guide

This appendix presents the interview guide used for the interviews in the data collection phase.

Time of interview:
Date:
Place:
Interviewer:
Interviewee:
Role of interviewee:

Who we are
Jennifer Linder & Padmini Subbiah
Studies Software Engineering Master at Chalmers.
Now we are doing our Master Thesis about "Deriving Contextual Definition and Requirements from Use Cases"

Purpose of Study
The purpose of this study is to investigate and explore the current status and uncover challenges for deriving contextual definition and requirements from use cases. The purpose of this interview is to get your personal experience and knowledge regarding the topic and the process of deriving contextual definition and requirements from use cases for autonomous drive.

How to handle certain field issues
If lose internet connection or get kicked out of the meeting. Reconnect as soon as possible. Otherwise we will email you in order to rebook another meeting hopefully soon.

Final Questions before Interview Starts
Do you have any questions before we start the interview?

Tell the interviewee they are able to hide the subtitles (captions)

Are you alright with us recording this meeting?

Start recording *Start transcription*

Research Questions:

RQ1: Which current solutions work well for deriving contextual definition and requirements from use cases?

RQ2: What are the challenges with deriving contextual definition and requirements from use cases?

RQ3: Which support would be appropriate for deriving contextual definition and requirements from use cases?

Interview Questions

General Questions

1. What is your role in the company?
2. Describe shortly your role and what you work with?
3. What is your experience in this field of study?

Establish their understanding of terms and concepts

4. Are you familiar with the concepts of ODD, Context and Use cases? 4.1. Which ones?

- If yes ODD,
  4.1.1 We give a description of ODD.
  4.1.1.1 Do they agree with that description? What additional information do you think that it should include?
  4.1.1.2 If they don't agree with the description, why not?

- If yes Context,
  4.1.2. Can you describe the context?
  4.1.3. Can you give an example?

- If yes Use Cases,
  4.1.4. What do you think is the most relevant information for an use case for autonomous drive?

5. Are there any problems that come to mind when understanding the ODD and requirements?

- If Yes, What are the problems?

6. Is the context limited by the levels of automation?

7. How would you go from context to ODD?
7.1 What additional information do you need in order to go from context to ODD?
7.2 If confused or unclear, what difference do you see between context and ODD?

Process

8. How would you describe the process of deriving ODD and requirements from use cases?
8.1 Is what you described how it is done in the industry today?
8.2 If they don't want to answer or can't, is there a lack of process or can't you disclose this information due to your company?

9. What do you think about the current way of deriving ODD and requirements from use cases?
9.1 What works well in the current process?
9.2 What are the issues in the current process?
9.2.1 Which of these issues is the most critical one? Do you see that issues occur more in certain areas?

10. Do you think that the current way of deriving ODD and requirements from use cases could be refined further?

- If yes, How?

11. Who is responsible for deriving the ODD from use cases?
11.1 Do you think they should be the role/area responsible for this?

12. Who is responsible for defining the context and requirements?
12.1 Do you think they should be the role/area responsible for this?

13. Who maintains the ODD? Who maintains the requirements?
13.1 Do you think they should be the role/area responsible for this?

14. What are the main concerns with deriving ODD and requirements from use cases?
14.1 If concerns are given, why are these the main concerns?

15. What other solutions do you think can be appropriate for deriving ODD and requirements from use cases?

16. Would you like to improve any aspects in deriving ODD and requirements from use cases?
16.1 If yes, What aspects would you like to improve? How and why?

Wrap-up Questions

Summarize a bit what we have discussed before finishing with this question 17. Is there anything else that you would like to add?

! Stop recording. ! Save transcription.

More information to Interviewee

- Follow-up email/meeting in case of any questions
- Focus group - where you will be able to see results from our study
- Consent form - can you send us a signed version? Do you have any questions regarding that?

Finish!

A.2 Broad categories and subcategories from thematic analysis

- USE CASE

  USE CASE-FAMILIARITY

  USE CASE-DEFINITION

  USE CASE-IMPORTANT INFORMATION

  USE CASE-PROBLEMS

  USE CASE-RESPONSIBLE

  USE CASE-OTHER

- ODD

  ODD-FAMILIARITY

  ODD-DEFINITION

  ODD-MISSING INFORMATION

  ODD-PROBLEMS

  ODD-RESPONSIBLE

  ODD-OTHER

- CONTEXT

  CONTEXT-FAMILIARITY

  CONTEXT-DEFINITION

  CONTEXT-PROBLEMS

  CONTEXT-RESPONSIBLE

  CONTEXT-OTHER

- REQUIREMENTS
  - REQUIREMENTS-DEFINITION
  - REQUIREMENTS-PROBLEMS
  - REQUIREMENTS-RESPONSIBLE
  - REQUIREMENTS-OTHER
- CONTEXTUAL DEFINITION
  - CONTEXTUAL DEFINITION-FAMILIARITY
  - CONTEXTUAL DEFINITION-DEFINITION
  - CONTEXTUAL DEFINITION-PROBLEMS
- CONTEXTUAL INFORMATION
  - CONTEXTUAL INFORMATION-DEFINITION
  - CONTEXTUAL INFORMATION-RESPONSIBLE
  - CONTEXTUAL INFORMATION-OTHER
- CONTEXTUAL DESCRIPTION
  - CONTEXTUAL DESCRIPTION-DEFINITION
  - CONTEXTUAL DESCRIPTION-OTHER
- PROCESS
  - PROCESS-CURRENT PROCESS
  - PROCESS-WORKS WELL
  - PROCESS-CHALLENGES
  - PROCESS-REFINEMENT
  - PROCESS-RESPONSIBLE
  - PROCESS-MAIN CONCERN
  - PROCESS-OTHER SOLUTIONS
  - PROCESS-OTHER

- OTHER

  OTHER-CHALLENGES

  OTHER-EXAMPLE

  OTHER-GENERAL

  OTHER-LEVELS OF AUTOMATION

  OTHER-OTHER SOLUTIONS

  OTHER-OUT OF SCOPE

A.3 Workshop

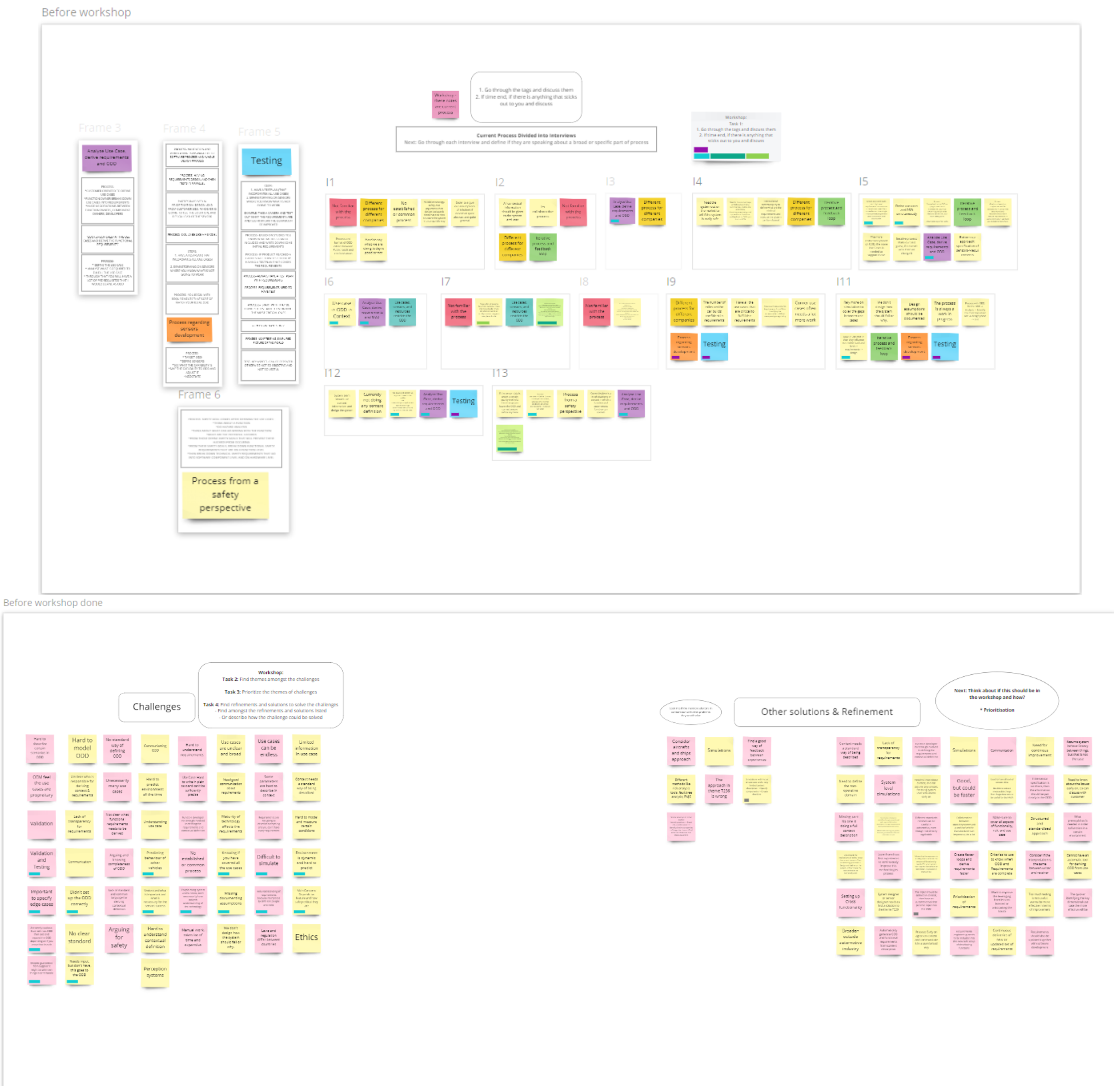

Figure A.1: Themes before workshop

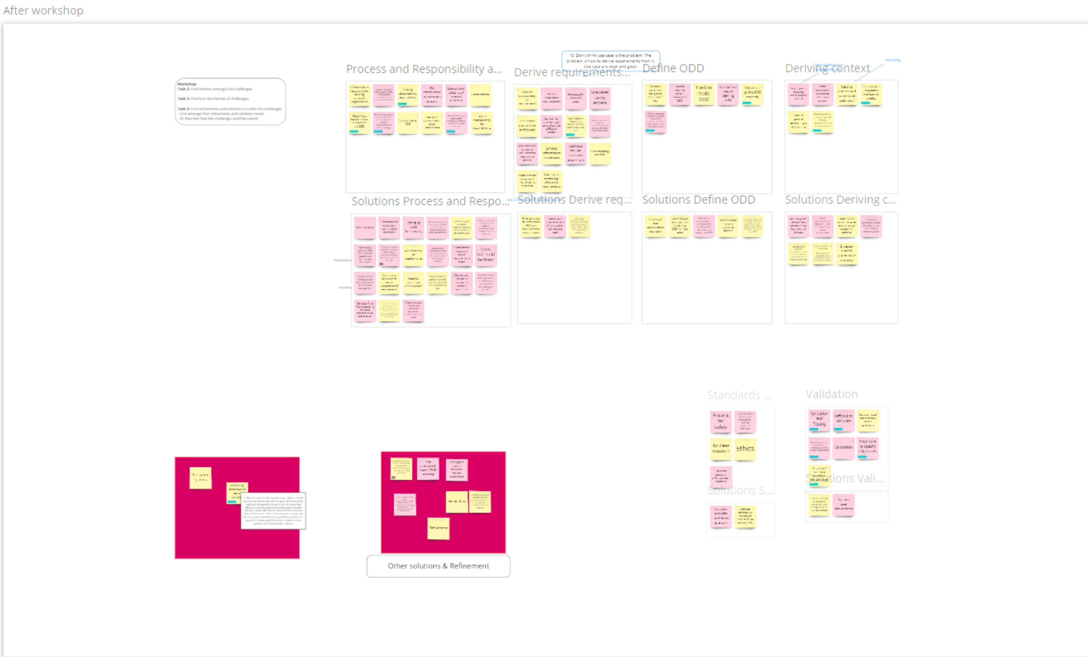

Figure A.2: Themes after workshop

A.4 Complete Mindmap for Context

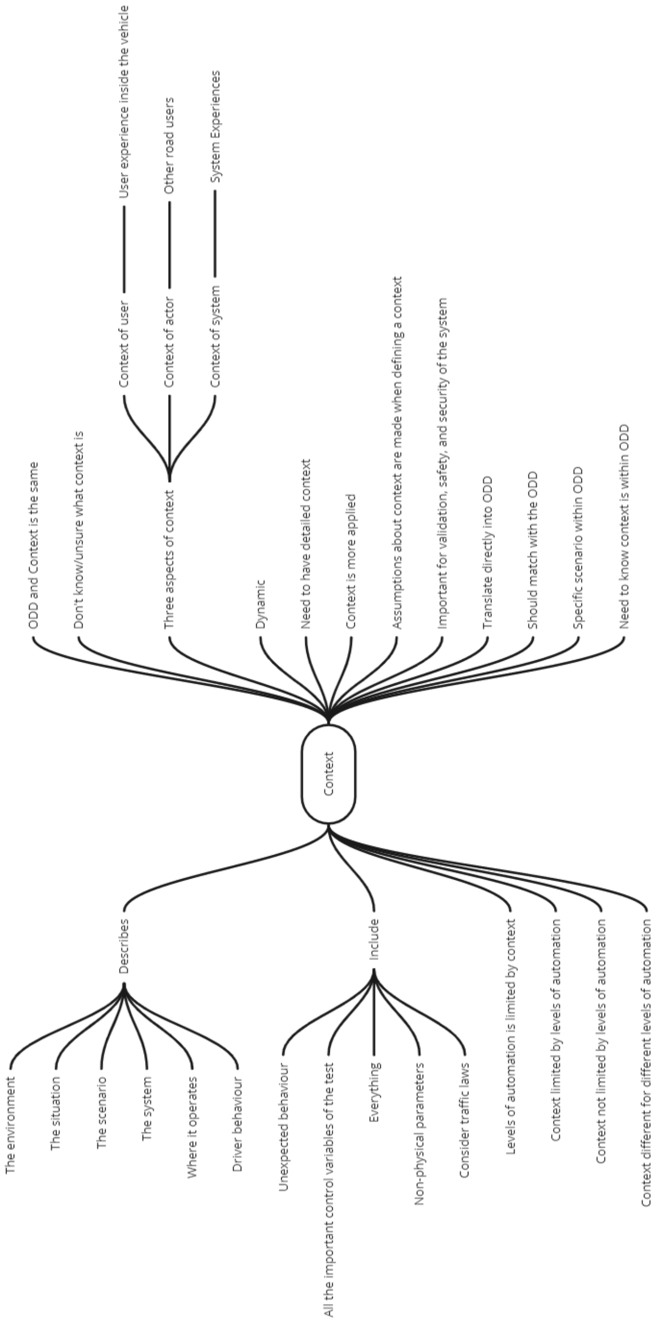

Figure A.3: Themes identified with regards to the definition of context

A.5 Ranking for most promising improvement for Process, Responsibility, and Communication

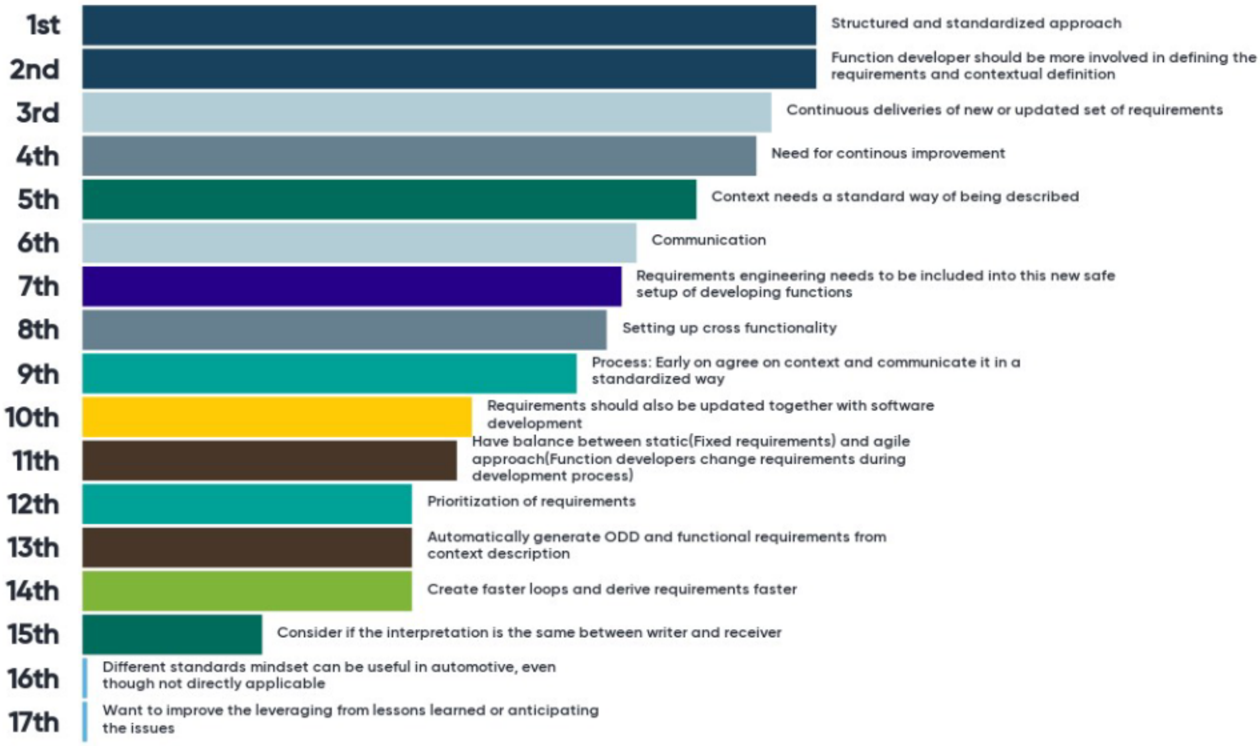

Figure A.4: Ranking of the most promising improvements for Process, Responsibility, and Communication

A.6 Full Ishikawa diagram for challenges

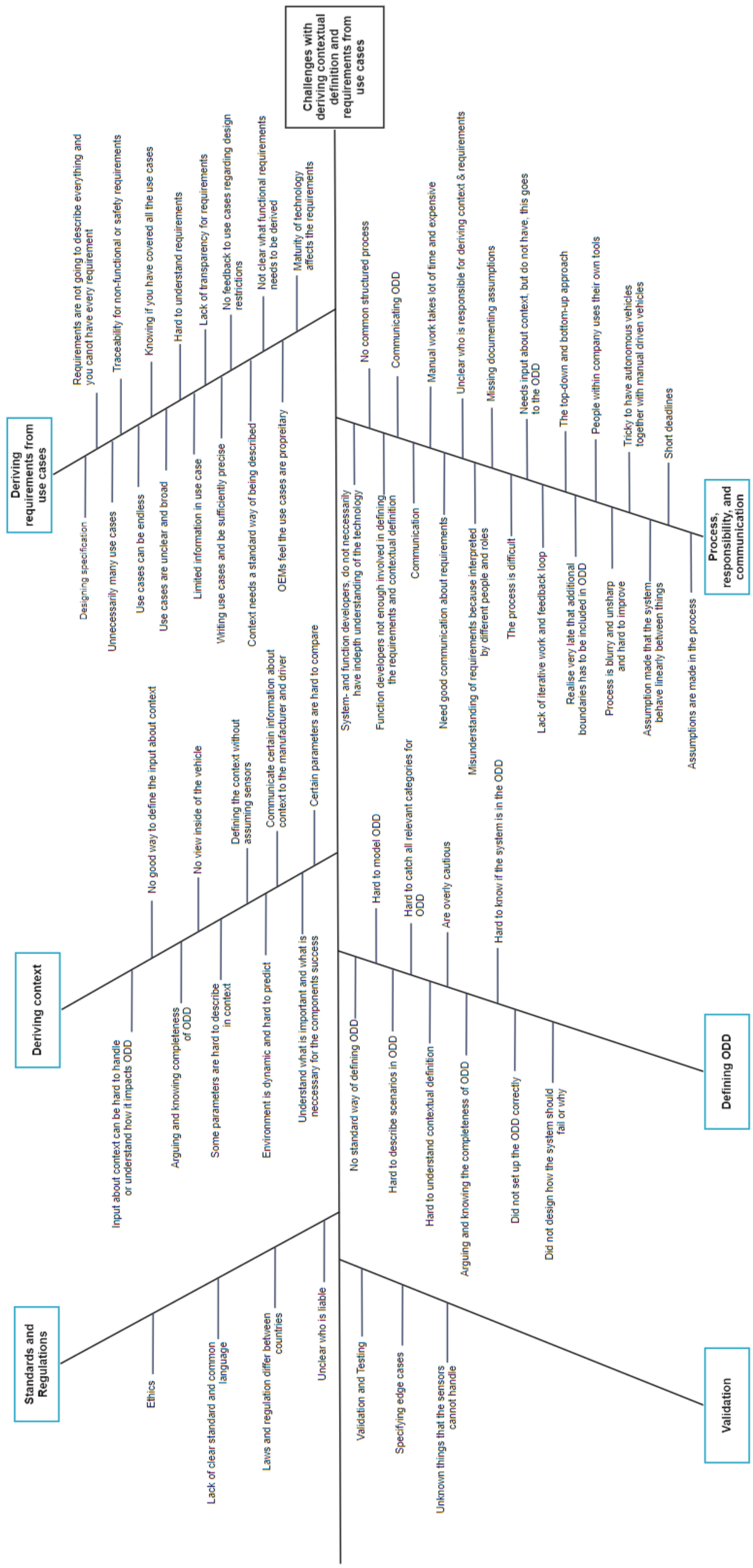

Figure A.5: The complete Ishikawa for all the challenges identified