Thesis for The Degree of Licentiate of Engineering

Accelerating the Design Phase: Towards DevSafeOps for Autonomous Driving Software

Ali Nouri

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg
Gothenburg, Sweden, 2024

**Accelerating the Design Phase:
Towards DevSafeOps for Autonomous Driving Software**

Ali Nouri

ISSN 1652-876X

Department of Computer Science and Engineering
Division of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg
SE-412 96 Göteborg,
Sweden
Phone: +46(0)31 772 1000

Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2024.

"Law 1: A robot may not injure a human being or, through inaction, allow a human being to come to harm."

- Handbook of Robotics, 56th Edition, 2058 A.D.

Abstract

Background: The safety of Autonomous Driving (AD) remains a barrier to its widespread adoption, as evidenced by recent incidents. Factors such as the complex environment, evolving technologies, and shifting regulatory and customer requirements necessitate continuous monitoring and improvement of AD software. This is a process that may favor software and system engineering supported by DevOps. The iterative DevOps process is crucial, serving two purposes: satisfying customer demands through continuous improvement of the function and providing a framework for timely responses to unknown bugs or incidents. However, any update to the software must follow rigorous safety processes prescribed by standards, regulations, or the state of the art in industry. Incorporating these safety activities into the DevOps forms an iterative process called DevSafeOps. These necessary activities, although vital for safety assurance, inherently lead to a compromise in rapidity.
Research Goal: In this work, we initially identify the challenges in the rapid DevSafeOps in AD development and then explore existing solutions. Subsequently, we propose two approaches for accelerating the primary activities in the AD development, which are requirements engineering and safety analysis.
Methods: To address each research objective, diverse research methods are utilized. Interview studies and a systematic literature review are conducted to identify the challenges and research gaps. Then, design science, interview study, and a case study are employed for the proposed approaches.
Results: Initially, the challenges and research gaps related to each essential activity for the safety of AD are identified (Papers A and B). The proposed solutions in literature are identified and mapped to the challenges (Paper B). Then, two approaches are proposed for the rapidity of safety analysis, which is the initial step in the development. We adapt System Theoretic Process Analysis (STPA) for distributed development within automotive system engineering, which is our suggestion to approach the first challenge (Paper C). As an alternative approach, a Large Language Model (LLM)-based hazard analysis risk assessment prototype is developed and evaluated to enable automation (Papers D and E).
Conclusions: There are multiple challenges in achieving rapid DevSafeOps in AD development. The design phase, as a stepping stone of development, was underexplored with respect to methods for rapid updates in its artifacts. In one approach, we propose adapting STPA for multiparty distributed development to increase the speed of DevSafeOps. Subsequently, we explore the possibility of using LLMs to perform design phase activities with reduced engineers' involvement. These two proposed approaches have the potential to contribute to an increase in speed in the design phase, one by enabling distributed development, and the other by automation.

Keywords

Autonomous Vehicles, DevOps, DevSafeOps, Safety, Requirements Engineering, Hazard Analysis Risk Assessment, Large Language Model, STPA

Acknowledgment

First and foremost, I am sincerely grateful to my supervisor, Prof. Christian Berger, for his continuous support, guidance, and insightful feedback in shaping my research. Your patience, encouragement, and availability during times of stress and difficulty have been invaluable to me. I also thank Dr. Beatriz Cabrero-Daniel, whose kind support and feedback have greatly contributed to this work.

I extend my heartfelt thanks to my industrial supervisors, Dr. Fredrik Törner and Dr. Håkan Sivencrona (formerly at Zenseact, now at Volvo Cars). Fredrik, thank you for trusting me and guiding me from the very start; without your support, effort, and guidance, this project would not have been initiated or sustained. Håkan, you have shown me the bigger picture and have been a supportive mentor, alongside sharing so many brilliant ideas. Your expertise and experience have provided me with a unique perspective.

I would like to thank Dr. Reza Khanzadi (formerly at Volvo Cars, now at Spotify) for his guidance during the initial steps of this journey and for the inspiring discussions that have significantly shaped this work. Reza, you not only led me technically but also mentored me to tackle challenges effectively. I am also grateful to Behnam Safakhah for his excellent support throughout my research. Behnam, you always trusted me and enabled me by providing everything I needed for my work.

I am very grateful to Prof. Philip Koopman for his insightful feedback, and I extend my sincere thanks to my examiner, Prof. Jan Bosch, for his valuable evaluation and support.

My deepest appreciation goes to my parents, Forouzandeh and Masoud, whose selfless support gave me the freedom to explore and pursue my dreams. Your belief in me has been a constant source of motivation. Your love, sacrifices, and encouragement have been the foundation upon which I stand today.

Most importantly, I would like to express my deepest gratitude to my beloved wife, Ghazal. Your patience, understanding, and unwavering support have been my greatest source of strength throughout this journey. Thank you for being by my side every step of the way.

This research has been partially funded by Sweden's Innovation Agency (Vinnova, diarienummer: 2021-02585) and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

List of Papers

Appended Papers

This thesis is based on the following papers:

[**Paper A**] **A. Nouri**, C. Berger, F. Törner, *An Industrial Experience Report about Challenges from Continuous Monitoring, Improvement, and Deployment for Autonomous Driving Features*
Proceedings of the 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Gran Canaria, Spain, 2022, pp. 358-365.

[**Paper B**] **A. Nouri**, B. Cabrero-Daniel, F. Törner, C. Berger, *The DevSafeOps Dilemma: A Systematic Literature Review on Rapidity in Safe Autonomous Driving Development and Operation*
Submitted, under review in Journal of Systems and Software.

[**Paper C**] **A. Nouri**, C. Berger, F. Törner, *On STPA for Distributed Development of Safe Autonomous Driving: An Interview Study*
Proceedings of the 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Durres, Albania, 2023, pp. 5-12.

[**Paper D**] **A. Nouri**, B. Cabrero-Daniel, F. Törner, H. Sivencrona, C. Berger, *Welcome Your New AI Teammate: On Safety Analysis by Leashing Large Language Models*
CAIN '24, Association for Computing Machinery, 2024.

[**Paper E**] **A. Nouri**, B. Cabrero-Daniel, F. Törner, H. Sivencrona, C. Berger, *Engineering Safety Requirements for Autonomous Driving with Large Language Models*
2024 IEEE 32nd International Requirements Engineering Conference (RE).

Other Papers

The following papers and patent were published during my PhD studies, or are currently in submission/under revision. However, they are not appended to this thesis, due to contents not related to the thesis, or I was not the main contributor.

[**a**] B. Cabrero-Daniel, Y. Fazelidehkordi, **A. Nouri**, *How Can Generative AI Enhance Software Management? Is It Better Done than Perfect?*
In: Nguyen-Duc, A., Abrahamsson, P., Khomh, F. (eds) Generative AI for Effective Software Development. Springer, Cham..

[**b**] T. Bouraffa, E. Kjellberg Carlson, E. Wessman, **A. Nouri**, P. Lamart, C. Berger, *Comparing Optical Flow and Deep Learning to Enable Computationally Efficient Traffic Event Detection with Space-Filling Curves*
27th IEEE International Conference on Intelligent Transportation Systems.

[**c**] **A. Nouri**, Z. Fei, *Training and operating an object detecting system for a vehicle*
EP4407482A1, European Patent Office, Volvo Car Corporation, published on 31 July 2024, pending.

Research Contribution

I was the main contributor in conceiving the idea, designing and planning the experiments/research methodology, data collection, data analysis, and writing of all appended papers (Papers A to E) and this thesis.

In paper a, I was consulted on the idea, and then contributed to writing, and reviewing. In paper b, I was the industrial supervisor of master thesis students, and contributed in writing, and reviewing the paper. In patent c, I was the main contributor in conceiving the idea, and design.

List of Abbreviations

AI	Artificial Intelligence
ADAS	Advanced Driver Assistance Systems
ADS	Autonomous Driving Systems
ASIL	Automotive Safety Integrity Level
AUR	Absence of Unreasonable Risk
CA	Control Action
DevOps	Development and Operations
DDT	Dynamic Driving Task
DMS	Driver Monitoring Systems
DMV	Department of Motor Vehicles
ECU	Electronic Control Unit
FMEA	Failure Mode and Effect Analysis
FTA	Fault Tree Analysis
HMI	Human-Machine Interface
KPI	Key Performance Indicator
LLM	Large Language Models
NHTSA	National Highway Traffic Safety Administration
ODD	Operational Design Domain
OEM	Original Equipment Manufacturer
OTA	Over-the-Air (updates)
PRB	Positive Risk Balance
SiL	Software-in-the-Loop
SLR	Systematic Literature Review
STPA	System Theoretic Process Analysis

TIM	Traceability Information Model
UNECE	United Nations Economic Commission for Europe

Contents

Part I

Summary

Chapter 1

Introduction

1.1 Background

1.1.1 Autonomous Driving System (ADS)

SAE J3016 [1] defines six levels of automation, ranging from “no driving automation (Level 0)” to unconditional (i.e., not ODD-specific) “full driving automation (Level 5)”. In such a technology-driven approach [2], these levels are determined by the degree of automation and the user’s role and responsibility during ADS [1] activation and in fallback scenarios. The human-centric approach is an alternative that focuses on human needs and abilities and then designs the system based on them [2]. Given the varying readiness and capability for supervision among users, these levels can be categorized into supervised and unsupervised [2], as shown in Fig. 1.1.

In Supervised modes, the function performs interventions to support the driver (i.e., Advanced Driver Assistance Systems or ADAS) or controls both lateral and longitudinal motion while the driver is responsible for the Dynamic Driving Task (DDT). Driver Monitoring Systems (DMS) monitor the driver to ensure their attentiveness and supervision (e.g., hands-on or eyes-on). ADAS functions offer warnings, steering assistance, or braking interventions to help the driver mitigate or prevent incidents.

These systems play a crucial role in progressing toward Vision Zero [3], which aims for no fatalities or severe injuries within the road traffic system. For instance, driver distraction is the main cause of rear-end collisions occurring at speeds up to 30 km/h [4]. This led the industry to introduce active safety functions, and later some Original Equipment Manufacturers (OEMs) offered them as a standard feature in all passenger car models (e.g., Volvo Cars’ City Safety [4]). A subsequent study, analyzing data from 2010 to 2014 from Sweden, showed a 28% reduction in the rate of rear-end frontal collisions compared with cars not equipped with this system [5]. Similar conclusions were drawn for the

[1] J3016 [1] uses the term ADS for Automated Driving System, covering Levels 3 to 5. However, this study focuses on Level 4/5 (i.e., unsupervised capability), often referred to as an Autonomous Driving System.

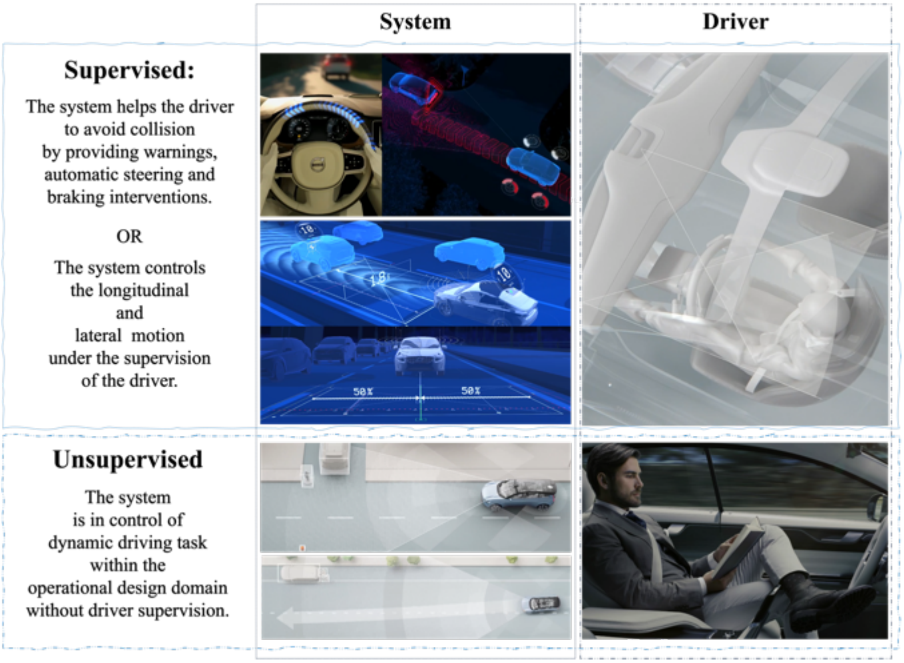

System
Driver
Supervised:
The system helps the driver to avoid collision by providing warnings, automatic steering and braking interventions.
OR
The system controls the longitudinal and lateral motion under the supervision of the driver.
Unsupervised
The system is in control of dynamic driving task within the operational design domain without driver supervision.

Figure 1.1: Human-centric automation levels.

reduction of car-to-pedestrian collisions by using other ADAS functions [6].

In Unsupervised mode, the vehicle fully controls the driving task. If a fallback is needed, the system will request a takeover but will not rely solely on the driver, as they might not be available. Thus, ADS is fully responsible for maintaining the safety of the driving task. While ADAS functions such as Automatic Emergency Braking (AEB) have demonstrated improvements in road safety, supported by statistics [2], the overall benefit of ADS on traffic safety is still theoretical. This highlights research gaps in both ensuring the safety of ADS and understanding its effect on traffic safety.

1.1.2 Safety of ADS

Despite the fact that the development of ADS started back in the 1980s [7], its safety remains a challenge, as there are still loss events. For instance, a pedestrian was severely injured in a recent mishap involving a robotaxi [8]. The pedestrian was pushed into the robotaxi's lane as a result of an initial hit-and-run collision with another vehicle. The ADS ran over the pedestrian, decelerated initially, and then moved forward while pulling the pedestrian.

The investigation [8] showed that the cause was neither hardware nor software failure. As mentioned in the investigation report [8], the ADS could initially detect both the pedestrian and the adjacent vehicle and even predict a

[2] Real-world benefits of crash avoidance technologies: summary of IIHS-HLDI findings, accessed Aug. 02, 2024, https://www.iihs.org/topics/advanced-driver-assistance

potential collision between them, but it could not predict either of them entering its lane. After the collision between the other vehicle and the pedestrian, the ADS classified the pedestrian as an unknown object and started deceleration. The collision detection system incorrectly detected the pedestrian's position as being on the side. As a consequence, the vehicle made the wrong maneuver instead of stopping in the lane.

Weak recognition and response to a nearby incident, as well as an inaccurate post-crash world model, are some of the technical issues and challenges as highlighted by Koopman [9]. There are multiple hints for a human driver in this situation, such as reducing speed when predicting a nearby incident even if both remain in the adjacent lane. However, as the system was following the design, it failed to adapt the trajectory based on this unforeseen event. Investigating these incidents sheds light on the need for novel or improved approaches in the design of ADS that enable the system to adapt to these unforeseen, complex, and numerous events.

Introducing the system without sufficient confidence or necessary fallback strategies can lead not only to safety risks but also to delays in deployment or even termination of the project. For instance, failing to assure public safety during the operation of ADS (without a human operator in the vehicle) led to the immediate revocation of ADS deployment and testing permits by the California Department of Motor Vehicles (DMV).[3]

1.1.3 Defining Safety

To assure and subsequently argue the safety of ADS, the term 'safety' must initially be defined both qualitatively and quantitatively. This definition is then broken down into more granular requirements for each building block of the ADS, which are used as criteria for pass or fail in the verification and validation phases.

Several stakeholders are involved in defining the safety criteria, such as legislators and involved parties in the industry. For instance, the United Nations Economic Commission for Europe (UNECE) published UN Regulation No. 157 [10], which defines qualitative and quantitative criteria to ensure the safety of Automated Lane Keeping Systems (ALKS). Additionally, UN Regulation No. 157 indirectly recommends compliance with ISO 26262 [11], ISO 21448 [12], and ISO 21434 [13] by requiring a competent auditor and assessor in these standards. Each standard defines safety criteria in a specific way, depending on its scope. For instance, ISO/TR 4804 [14] defines it as the "Absence of Unreasonable Risk (AUR)." ISO 26262, ISO 21448, and ISO 8800 refine the definition by adding a second part, making it more specific based on the root causes they target:

ISO 26262: "Absence of unreasonable risk due to hazards caused by **malfunctioning behaviour of E/E systems**"

[3] DMV STATEMENT ON CRUISE LLC SUSPENSION, accessed May 20, 2024, https://www.dmv.ca.gov/portal/news-and-media/dmv-statement-on-cruise-llc-suspension/

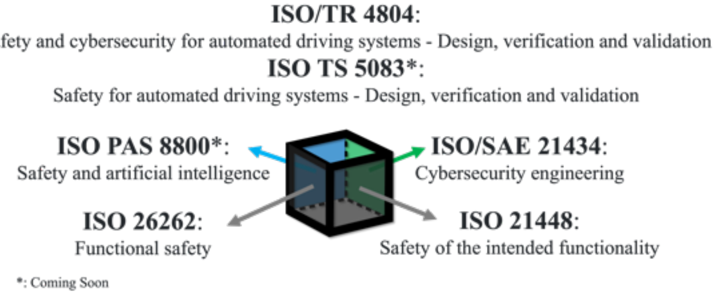

ISO/TR 4804:
Safety and cybersecurity for automated driving systems - Design, verification and validation
ISO TS 5083*:
Safety for automated driving systems - Design, verification and validation
ISO PAS 8800*:
Safety and artificial intelligence
ISO/SAE 21434:
Cybersecurity engineering
ISO 26262:
Functional safety
ISO 21448:
Safety of the intended functionality
*: Coming Soon

Figure 1.2: Depicting ISO standards for road vehicles relevant to the safety of ADS. Both ISO 4804 and ISO 5083 are specific to ADS applications and address gaps in relation to other standards or refer to them when applicable. Other standards mentioned are not specific to ADS applications, although ISO 21448 and ISO PAS 8800 primarily address challenges related to ADS.

ISO 21448: "Absence of unreasonable risk due to hazards resulting from **functional insufficiencies of the intended functionality or its implementation**."

ISO 8800: "Absence of unreasonable risk due to **errors of the AI system**."

Positive Risk Balance (PRB), as one of the concepts, can be used for the main quantitative safety criterion [14], which can then be broken down and used as acceptance criteria in ISO 21448 [12]. PRB expects the ADS to "cause fewer crashes on average compared to those made by drivers" [14]. However, questions such as "How much less?" (e.g., X% less) and "which driver?" (e.g., an attentive driver) need to be answered to calculate the quantitative target [15].

As highlighted in the paper "Redefining Safety for Autonomous Vehicles" by Koopman and Widen [16], both concepts of AUR and PRB fall short when applied to ADS. The authors point out that AUR fails to account for scenarios requiring moral and contextual understanding and might lead to risk redistribution to other road users, including emergency vehicles and vulnerable communities. For PRB, on the other hand, it is difficult to define a comparable baseline between human drivers and ADS. These issues underscore the need for updated safety definitions [16].

ISO 26262 [11] (functional safety, or so-called FuSa) prescribes remedies to avoid or mitigate systematic failures in hardware and software, as well as random hardware faults, in the design, implementation, verification, validation, and field monitoring phase. These activities are then used in a safety case to argue the achievement of functional safety of the item.

ISO 21448 (Safety of Intended Functionality or SOTIF), is a complementary standard to ISO 26262 [12], which contains recommendations for avoiding hazardous events caused by functional insufficiencies, incorrect/inadequate Human-Machine Interface (HMI), and insufficiencies of Artificial Intelligence (AI)-based algorithms. SOTIF starts by requiring the definition of acceptance criteria, which can be both qualitative and quantitative (e.g., one event per

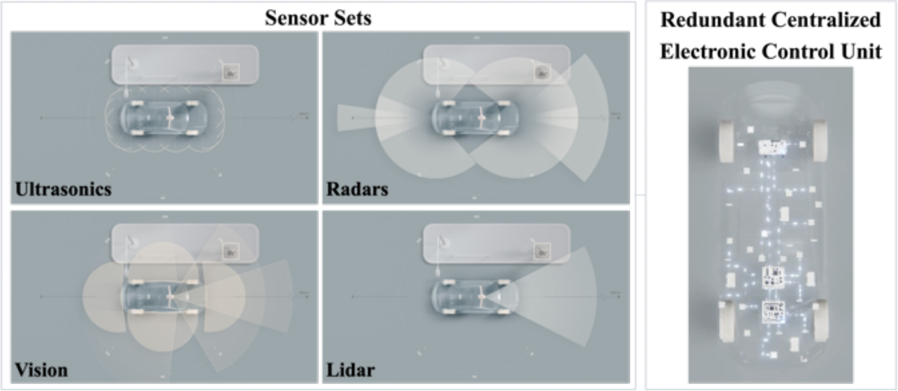


Figure 1.3: Presenting some of the diverse sensor technologies, their field of view, and redundant Electronic Control Units (ECUs). This includes the diversity of sensor technologies and redundant ECUs employed in Automotive Safety Integrity Level (ASIL) decomposition and arguing for sufficient independence between decomposed channels.

X km). The fulfillment of each acceptance criterion is then argued based on evidence from analysis, tests, and examinations for each validation target (e.g., No hazardous behaviour during X hours of testing) [12]. In case any of the acceptance criteria are not being met, then a functional modification is needed, which leads to changes in design and specification.

ISO 4804 [14] and upcoming ISO 5083 are standards that result from 'Safety First for Autonomous Driving' white paper [17]. These ADS-specific standards discuss on additional process aspects of ADS in relation to ISO 26262 and ISO 21448. Moreover, they contain technical and architectural recommendations for ADS. UL 4600 is another ADS-specific standard that addresses system-level aspects [18].

Using AI technology in automotive applications is not specific to ADS. However, relying on it for safety-critical tasks in a complex environment without human supervision is a new challenge for the automotive industry, which ISO 8800[4] aims to address. It will propose a framework covering all phases from development to operation, such as data completeness and quality.

1.1.4 Autonomous Driving System Technology

The ultimate goal in the development of ADS is unconditioned autonomous driving capability. However, due to known safety challenges, immature technologies in ADS such as AI, and the unbounded complexity of the environment, there is a need to divide the scenario space and start with one that is seen as less risky. When enough confidence is gained, the Operational Design Domain (ODD) would be expanded until all limitations are removed. For instance, highways with structured separations, where the exposure to Vulnerable Road

[4] ISO/CD PAS 8800, accessed May 20, 2024, https://www.iso.org/standard/83303.html

Users (VRU) is lower [14] can be seen as one. Others, however, decide to start in an urban area such as complex city streets [19].

Similar to all other robotic systems, ADS also utilize the sense-plan-act paradigm [14]. Perceiving relevant static and dynamic objects, localization, and trajectory prediction of relevant surrounding objects are the main capabilities expected from sensing. [14].

Since perception relies solely on the sensors in the system, it must reliably detect dynamic objects (e.g., other road users) and static objects (e.g., road boundaries), especially those with a risk of collision [14]. Moreover, the dynamic and complex nature of the environment makes it more challenging to create the world model. For instance, false positives or negatives in object detection might lead to a collision. As shown in Fig. 1.3, diverse sensing technologies such as radar, lidar, camera, and ultrasonic, each with different detection capabilities, lead to a more reliable perception system due to the complementary and redundant information received by each [20]. Diversity can be used in arguing for independence from systematic failures or technical limitations between two sensor clusters when a safety requirement is decomposed between them as two parallel channels to satisfy the same capability [11]. Diversity and redundancy requirements are not limited to the sensor technology but include the entire chain of components involved, such as the power supply and computation unit. The sensor fusion component analyzes, evaluates, and arbitrates between diverse received information from sensor clusters, and creates a relevant representation of the world around the ego vehicle [20].

Planning is responsible for creating a “collision-free and lawful driving plan” [14]. Moreover, it shall avoid being exposed to unsafe situations, for instance, through precautionary behaviors. Correctly executing the trajectory decided by the plan and communicating with other road users, including the driver, are the main capabilities of the actuators [14]. The trajectory execution is carried out through lateral and longitudinal motions using actuation systems such as steering, braking, and propulsion.

1.1.5 Automotive Safety Development Process

Safety Engineering Activities

As presented in Fig. 1.4, the development starts with specification and design. In this phase, the function is defined for a specific ODD. Data collection is also conducted in this phase to use the data for training the ML-based software and for extracting relevant scenarios. Fig. 1.5 shows one example of these vehicles, which, in addition to production-intent sensor sets, contain high-precision sensors on the rooftop that can be used as a reference sensors. Next, the software and hardware components are implemented and tested. They are then integrated to form the system, which is tested in different environments. At the end, assessment, audit, and certifications (depending on the region) are conducted. The item is monitored in the field, and modifications might be required to improve the performance of the function or expand the ODD.

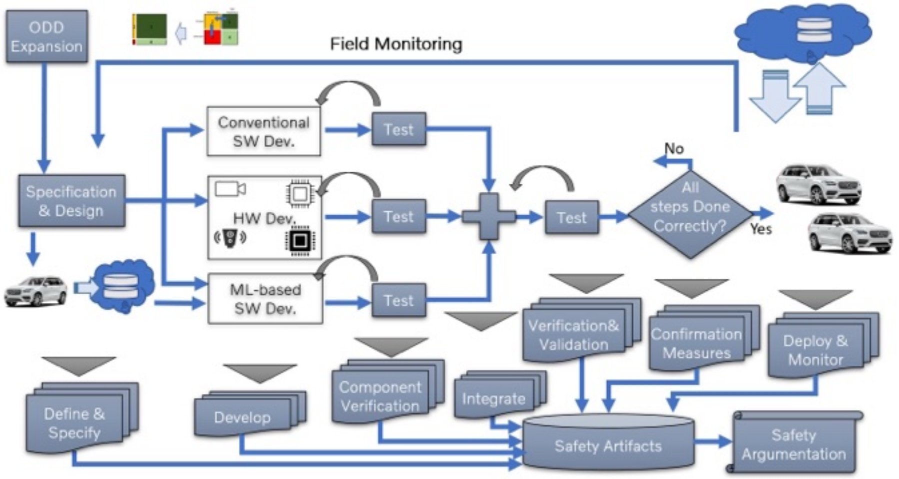


Figure 1.4: Presenting the development process for autonomous driving and its safety argumentation. It starts with specification and design, followed by implementation and testing. Subsequently, after assessment, the software will be updated in the vehicle. This process will be continuously repeated as part of ODD expansion. The elements in the figure are abstracted, with each representing an encapsulation of several activities. For instance, both clouds (top right and middle left) represent not only data storage but also activities such as data cleaning, preparation, and feature extraction, as prescribed by relevant guidelines and standards (e.g., Annex D, ISO 21448 [12]).

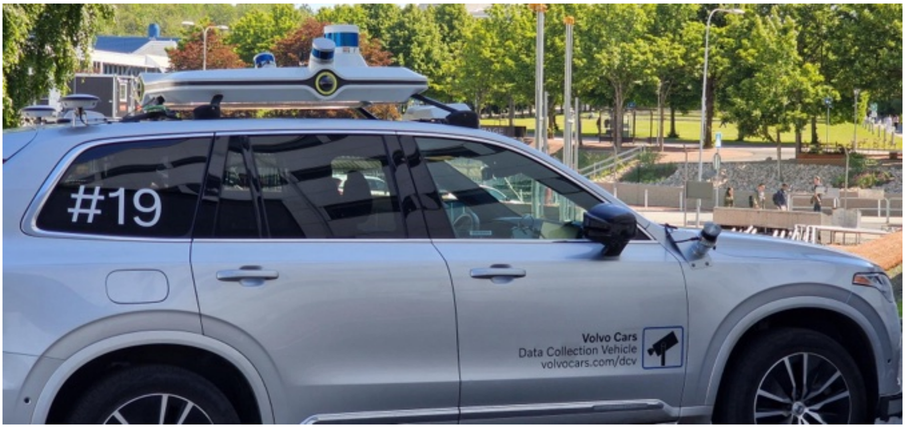


Figure 1.5: Presenting a data collection vehicle. The collected data is used for identifying relevant scenarios, utilized in the development of rule-based or machine learning software, and ultimately used for verification and validation.

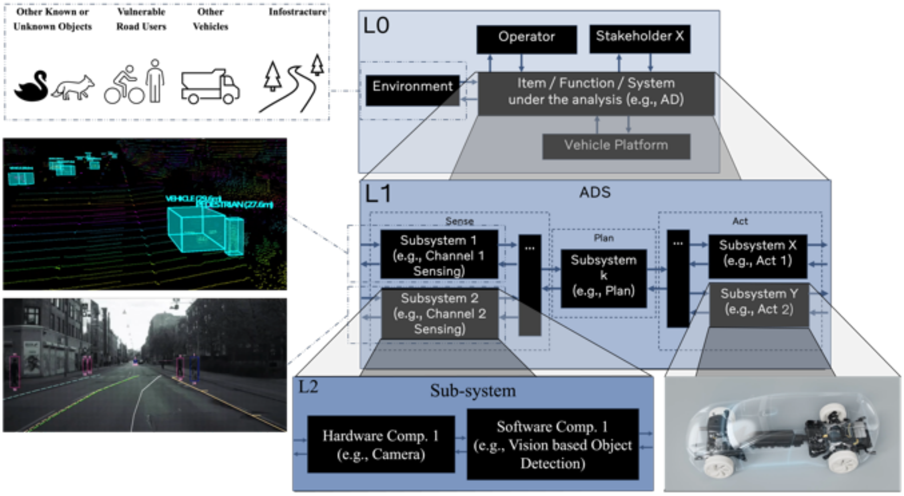

Other Known or Unknown Objects
Vulnerable Road Users
Other Vehicles
Infostructure
L0
Operator
Stakeholder X
Environment
Item / Function / System under the analysis (e.g., AD)
Vehicle Platform
L1
ADS
Sense
Subsystem 1 (e.g., Channel 1 Sensing)
Subsystem 2 (e.g., Channel 2 Sensing)
...
Plan
Subsystem k (e.g., Plan)
...
Act
Subsystem X (e.g., Act 1)
Subsystem Y (e.g., Act 2)
L2
Sub-system
Hardware Comp. 1 (e.g., Camera)
Software Comp. 1 (e.g., Vision based Object Detection)

Figure 1.6: Illustrating multi-abstraction levels in a system such as AD. In Level 0, the interaction between the system, users, and environment is depicted. In Level 1, the system itself is depicted. It is important to employ abstraction to reduce the complexity of the system. As the system's complexity increases, the number of abstraction levels also increases. The abstraction levels and their modules in the figure are for illustration purposes and do not include all modules, such as cloud-based components, other sensor technologies, or actuators.

Abstraction Levels

Safety assurance of such a software-intensive system in an environment with unbounded complexity requires a modularization and abstraction approach [21]. Automotive system engineering follows a well-established modular and multi-abstraction architecture, as illustrated in Fig. 1.6. The modules in each abstraction level are abstracted by removing or merging unnecessary characteristics of that module [22]. Each module shall be traceable to the lower or higher abstraction levels to enable impact analysis and maintainability of the system after modifications during the life cycle of the system. Fig. A.2 presents a bird's-eye view of the abstraction levels in the development process of the V model in the context of both ISO 26262 and ISO 21448. Each abstraction level and its relevant activities follow the flow of the V model as outlined below. The abstraction levels and activities on the left side of the V model are as follows, and the right side of V model adheres the same:

"Concept" is the initial abstraction level in design phase.

> **"Item Definition"** is a short description of the item, describing its functionality and ODD. It also includes the boundary diagram, which represents the item's interaction with the environment and stakeholders such as the driver.

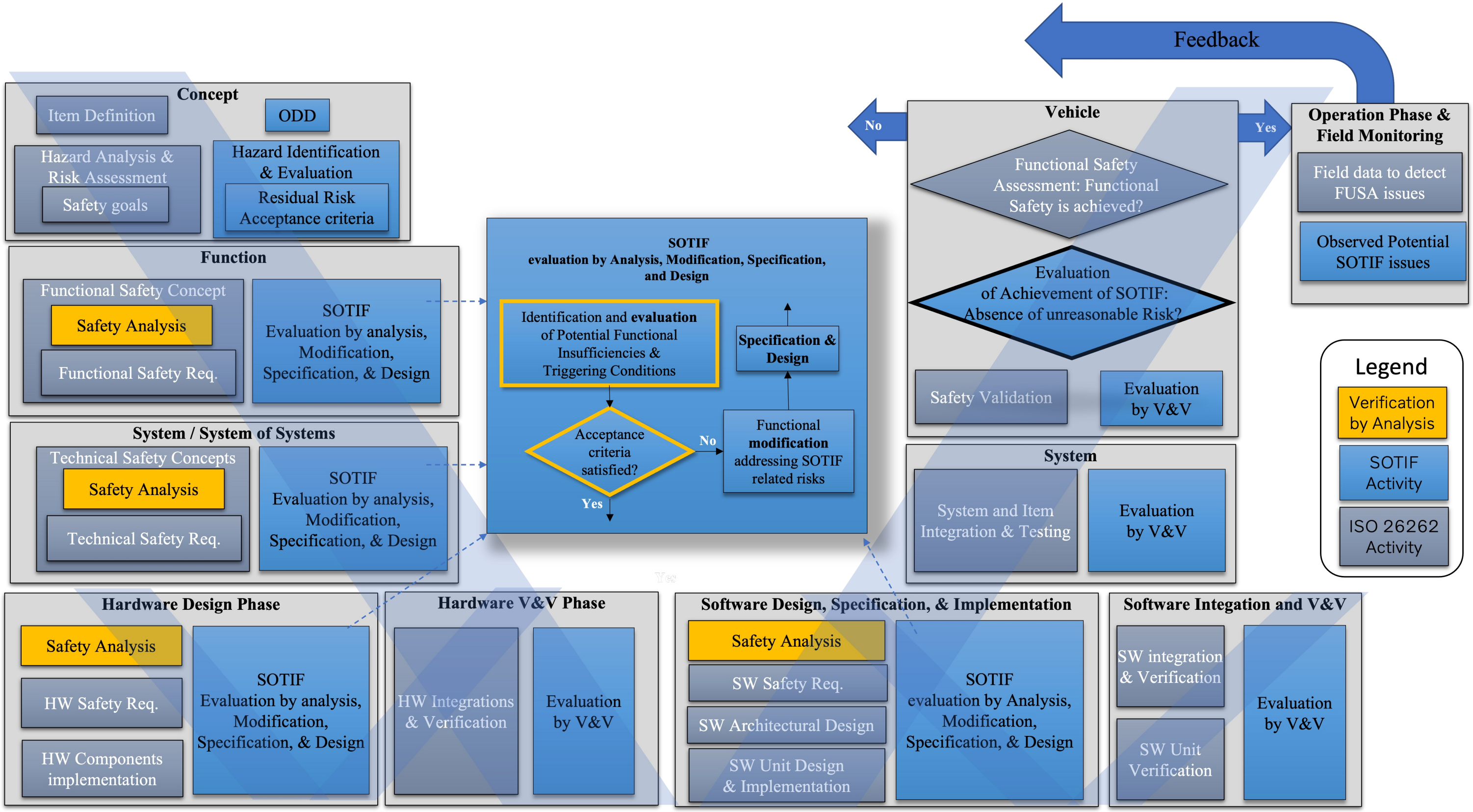

Feedback
Concept
Item Definition
ODD
Hazard Analysis & Risk Assessment
Safety goals
Hazard Identification & Evaluation
Residual Risk Acceptance criteria
Function
Functional Safety Concept
Safety Analysis
Functional Safety Req.
SOTIF Evaluation by analysis, Modification, Specification, & Design
System / System of Systems
Technical Safety Concepts
Safety Analysis
Technical Safety Req.
SOTIF Evaluation by analysis, Modification, Specification, & Design
SOTIF evaluation by Analysis, Modification, Specification, and Design
Identification and evaluation of Potential Functional Insufficiencies & Triggering Conditions
Specification & Design
Acceptance criteria satisfied?
No
Yes
Functional modification addressing SOTIF related risks
No
Vehicle
Yes
Functional Safety Assessment: Functional Safety is achieved?
Evaluation of Achievement of SOTIF: Absence of unreasonable Risk?
Safety Validation
Evaluation by V&V
Operation Phase & Field Monitoring
Field data to detect FUSA issues
Observed Potential SOTIF issues
System
System and Item Integration & Testing
Evaluation by V&V
Legend
Verification by Analysis
SOTIF Activity
ISO 26262 Activity
Hardware Design Phase
Safety Analysis
HW Safety Req.
HW Components implementation
SOTIF Evaluation by analysis, Modification, Specification, & Design
Hardware V&V Phase
HW Integrations & Verification
Evaluation by V&V
Software Design, Specification, & Implementation
Safety Analysis
SW Safety Req.
SW Architectural Design
SW Unit Design & Implementation
SOTIF evaluation by Analysis, Modification, Specification, & Design
Software Integration and V&V
SW integration & Verification
SW Unit Verification
Evaluation by V&V

Figure 1.7: Presenting mapped ISO 26262 and ISO 21448 activities to each abstraction level. Unlike ISO 26262, the activities in ISO 21448 are not clustered based on abstraction levels. However, as shown, they can be mapped to each abstraction level. For instance, the "SOTIF evaluation by Analysis, Modification, Specification, and Design" process is repeated for each abstraction level, although the scope and level of granularity are adapted to the element under development.

“Hazard Analysis & Risk Assessment” (HARA) and Hazard Identification & Evaluation are then enhanced to identify all hazardous events that result from the occurrence of malfunctions or functional insufficiencies in a specific scenario. The risk for each hazardous event is then assessed. If a hazardous event is safety-critical, a safety goal (ISO 26262) or an acceptance criterion (ISO 21448) is specified and allocated to the hazardous event to avoid or mitigate it.

“Function” level is the next abstraction level, which contains the functional architecture, safety analysis, and functional safety requirements gathered in functional safety concept and SOTIF safety concept.

“Deductive Safety Analysis” (Top-Down) methods are brainstorming tools used to identify the potential causes of the violation of a safety requirement under analysis. Fault Tree Analysis (FTA) is a deductive method that can be used to analyze the parent requirement and identify the potential faults (causes) in the modules at the current abstraction level.

“Inductive safety Analysis” (bottom-up) methods can be used to identify the effects of a specific failure mode on higher levels, which might lead to violation of safety requirements. Failure Mode and Effect Analysis (FMEA) is an inductive approach that starts from the failure modes of each module at the current level and identifies the effects at higher levels, which can violate the safety goal.

“Functional Safety Requirements” and SOTIF safety requirements are then designed to avoid or mitigate the unsafe failure modes of the functional blocks. This is achieved through modifications to the original functional block and the specification of functional safety requirements, which are derived from safety goals and acceptance criteria. Ideally, the functional safety requirements are technology-agnostic.

“System” level is the first level where technical aspects are designed. A function can be realized by multiple systems, and each system consists of at least a controller, and internal or external sensor(s) and actuator(s). A technical safety concept is the container for system-level safety requirements, including the safety analysis.

“Safety Analysis” are performed similarly to those at the function level.

Technical Safety Requirements are specified for each component in the system and derived from the functional safety requirements. Decomposition is one of the techniques that can be used to design safety solutions. Decomposition is not specific to the system level and can be applied at higher or lower abstraction levels.

“Software” design and specification are the last major abstraction level in the design phase, although both software and hardware might contain multiple abstraction levels. The system level technical safety requirements are broken down into hardware and software safety requirements. After performing safety analysis, relevant software safety requirements are specified, and the safety solutions are designed in the software architectural design.

1.1.6 System Theoretic Process Analysis (STPA)

Verification by Analysis can be employed to identify functional inefficiencies, technical limitations [12] and design mistakes [11] that can lead to safety-critical hazardous events during an early development phase. Analysis is one of the key activities in the safe-by-design approach. STPA provides a systematic approach to identify and analyze hazardous events and corresponding causes in complex systems such as ADS. STPA is one of the methods referenced for AD development by regulators such as UN Regulation No. 157 [10]. The following summary of each step in STPA is gathered by aggregating and adapting the proposed steps from the STPA Handbook [23], SAE J3187 [22], Annex B.4 in ISO 21448[12], and Appendix B in “Safety of the Intended Functionality of Lane Centering and Lane Changing Maneuvers of a Generic Level 3 Highway Chauffeur System” by NHTSA [24].

Step 1 - Define the Purpose of Analyses: The goal of this step is to define the function and then to identify hazards and relevant safety constraints to avoid them. This step consists of several sub-steps:

- System Scope: Here, system boundaries are defined and shall include information such as purpose, constraints (e.g., ODD), and actors [22]. This step is similar to “Item definition” in the context of ISO 26262.

- Losses: The losses are the overall goal of the system to fulfill or topics of concern that affect the stakeholders of the system such as “financial loss” and “Loss of life or injury”. Based on the instructions and examples in the STPA Handbook [23], they should be quite generic for each context (e.g., financial, safety, privacy) and they are not allowed to be more granular. For example, regarding safety-related aspects, the only loss is “Loss of life or injury”. Unlike ISO 26262, which assesses the hazard based on four levels of severity from no injuries (S0) to fatality level (S3) [11], STPA only *identifies* the hazard and does not intend to *assess* it.

  Most of the time, such losses are connected such as “Loss of life or injury”, which leads to “financial loss”. But in some cases, satisfying one can also violate another one. This is why there is a need to prioritize the losses. Then, the priority would be inherited by traceability between the results of each step and the relevant loss [22].

- Hazards: A hazard is caused by malfunctioning behavior [11] (Functional Safety causes) or insufficiencies [12] (SOTIF causes) of the function, which in a specific scenario would lead to a loss.

- System Level Constraints: Based on the example mentioned in SOTIF [12], system level constraints are in the same abstraction level as safety goals in the context of FuSa or same as validation targets in SOTIF. Since in automotive abstraction levels, a vehicle function consists of several systems and thus, “Vehicle-level safety constraints” is a more suitable name for this sub-step as also proposed in SOTIF [12].

Step 2 - Model Control Structure: A control structure is a model representing the interaction between elements (i.e., control action and feedback) and the hierarchy of control [23] at each abstraction level. The relevant properties of elements and their interactions need to be chosen for each designated abstraction level.

Although the main part of the analysis will be done in the next steps, drawing the control structure itself may highlight design weaknesses, as raised by the STPA Handbook [23]. Missing a block, control action, feedback, or having an incorrect hierarchy of control in the control structures, such as the one presented by Xing et al. [25], can result in an incomplete analysis in steps 3 and 4, potentially leading to missing or incorrectly defined safety requirements. The control structure in [25] is presented as a physical model rather than a functional model, which can lead to the omission of functional elements mapping to the main blocks in a control structure (e.g., controlled process). Additionally, the hierarchy of elements does not follow the order proposed by the Handbook [23]. For instance, the driver, as the operator, should be positioned at the top of the control structure, but in this case [25], it is placed at the bottom.

Step 3 - Identify Unsafe Control Actions: Based on the STPA Handbook, an Unsafe Control Action (UCA) is a state or a behaviour of control action that, in a specific scenario, would lead to a hazard [23]. In other words, each UCA leads to a violation of the vehicle-level safety constraint. The UCAs shall be traceable to at least one hazard; otherwise, it is an indication for a missing hazard and the relevant hazard, and relevant hazard shall be introduced.

Using a set of guidewords combined with control actions can help to find UCAs. The STPA Handbook proposes “not providing,” “providing,” “too early or late,” “out of order,” “stopped too soon,” and “applied too long.” Since these guidewords will be used in combination with control actions, they shall be adapted to the control action and the context. In Fig. 1.8, a set of suitable guidelines for AD are shown with respect to the required and safe control action. The list is not limited to these and based on the nature of the Control Action (CA), more guidewords can be used, or some might not be applicable. In addition, enhancing the pre-existing knowledge about the undesired or unsafe control actions from other projects might be helpful.

Safety Requirements: In ISO 21448 [12] and STPA Handbook [23], the next sub-step is to specify safety constraints to prevent UCAs. SAE J3187 [22] introduced this sub-step only in step 4, which leads to an incomplete analysis in step 3, as there would not be any safety requirement to avoid UCAs.

Step 4 - Identify Loss Scenario: Identifying the causal factors is the last step, after which relevant safety constraints shall be specified. There are four main types of causes that shall be analyzed in this step [23]:

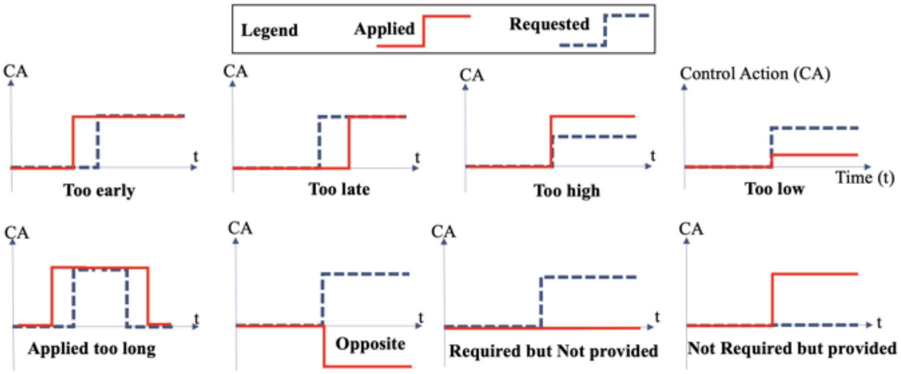


Figure 1.8: Illustrating the suitable undesired or unsafe manners, which lead to UCAs in AD, extracted from the STPA Handbook [23].

- Controller-related such as "inadequate control algorithm"
- Feedback or Sensing-related such as "unsafe control inputs" or "incorrect feedback"
- Execution or Actuation-related such as "the command is not received" or "improperly executed"
- Other factors related to controlled processes such as "the actuation is not effective"

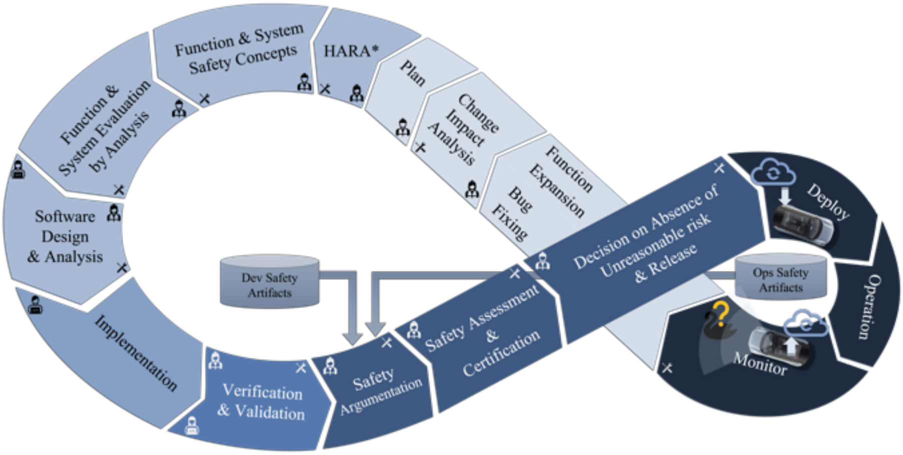
The first two types in the list are the causes for the unsafe control actions in step 3, and the last two are the causes for the improper execution of a correct control action.

Safety Requirements: Then, similar to step 3, the safety requirements are specified to mitigate or prevent the identified causal factors [22]. Safety requirements, which are defined in steps 3 and 4, are communicated to the next abstraction level to be used as a starting point of the analysis.

1.1.7 Software Update and DevOps

Similar to other domains, the role of software in automotive systems is growing, and a significant part of the consumer's experience is driven by the vehicle's software, known as software-defined vehicle [26]. Centralized computing units, over-specified hardware components (e.g., additional sensors for future functions), and over-the-air updates (OTA) are enablers that facilitate software updates.

Consumers' expectations, legislation, and technological aspects are evolving continuously, which require continuous function improvements to stay competitive. Moreover, field monitoring and fast reactions to unknown, rare, but still possible incidents are crucial to mitigate and avoid them before they lead to loss events. Iterative and continuous software development, deployment,


Function & System Safety Concepts
HARA*
Plan
Function & System Evaluation by Analysis
Change Impact Analysis
Function Expansion
Bug Fixing
Software Design & Analysis
Deploy
Decision on Absence of Unreasonable risk & Release
Dev Safety Artifacts
Ops Safety Artifacts
Operation
Implementation
Safety Assessment & Certification
Monitor
Verification & Validation
Safety Argumentation

HARA*: Hazard Analysis Risk Assessment (ISO 26262), Identification and Evaluation of Hazard (ISO 21448)

Figure 1.9: Presenting the integration of required safety activities in DevOps cycles. The significant number of activities required on the development side, compared to the operation phase, is illustrated by the asymmetrical shape of the loop.

and monitoring are widely used for non-safety-critical software applications to enable rapid and innovative feature improvements and bug fixes. This approach, known as DevOps, couples continuous Development (Dev) and Operations (Ops) [26].

1.1.8 Safety of DevOps

Software update and system modification in iterative loops are not a new challenge for automotive systems. Change management [11] (ISO 26262, Part 8, clause 8) is a systematic process for maintaining the safety of the item while implementing changes to the item or its elements during an item's lifecycle. For instance, as part of the safety planning at the start of each loop, performing a "Change Impact Analysis" is required to identify the affected work products and elements in the item, as well as the potential impact on them. Fig. 1.9 presents the integration of safety activities in DevOps infinity loop, which, unlike non-safety-critical applications, does not rely solely on final stage testing in the field. Instead, the process heavily requires "safe-by-design" principles.

Munk and Schweizer [26] proposed the SafeOps approach, while Siddique [27] introduced the SafetyOps framework, which first focuses on ensuring safety in iterative development and second on integrating safety and DevOps by emphasizing the need for adaptation to unique needs of safety. We propose DevSafeOps as a term to emphasize Safe by Design in the development phase, which the earlier terms miss.

1.2 Motivation and Problem Domain

Integrating automotive system engineering, with all its complexities and numerous activities as explained in Sec. 1.1.5, with the rapid and frequent iterative DevOps process is challenging. In other applications, the risk of prioritizing speed over quality might be seen as an option; however, this is not acceptable in safety-critical applications, and the OEM will not release the product without performing all required activities to ensure the product is safe. Hence, the automotive industry is lacking comprehensive methodological and technical approaches to enhance the speed and efficiency of DevSafeOps iterative loops.

1.3 Potential Approaches

Distributed development and automation using software tools are already part of the process in automotive system engineering, which also improves speed. Distributed development as one approach represents the category of development models which enhance the efficiency and effectiveness of teams of engineers during system design. "Generative AI for Systems Engineering," on the other hand, represents technologies that can enable automation in activities involved in the automotive process.

1.3.1 Distributed Development

Distributed development is essential due to the complexity of such systems and the short time to market. It also enables the OEM to leverage the expertise from specialized suppliers in each technology, ensuring the system is developed to the highest performance and quality levels. Thus, modularization needs to be employed to clearly define the perimeter of the responsibility of each internal or external team involved in development. Moreover, each team (especially if external) is responsible only for their own module, which also protects the intellectual property and business factors of each party.

1.3.2 Generative AI for Systems Engineering

Software-in-the-Loop (SiL) test environments and pipelines leverage predefined test cases in a simulation environment to automate some aspects of verification when new software is introduced [28]. However, most automation pipelines are applied to repetitive tasks that do not require creativity. Moreover, some activities in the safety process, such as analysis methods and requirements engineering, are based on natural language, making their automation challenging.

Generative Artificial Intelligence (AI)-based tools capable of text processing and generation can be seen as a potential approach for these tasks. One of the promising options are Large Language Models (LLM). Unlike domain-specific language models, LLMs are trained on a vast amount of text across various domains, and their outputs are perceived as relevant for diverse tasks [29]. As the model generates text based on the input, crafting the input, known as prompt engineering, is crucial for obtaining relevant output. Some of these prompt patterns are suggested by studies such as [30].

However, due to the stochastic nature of these models, multiple weaknesses are reported, such as generating text which is not correct with respect to reality, commonly referred to as hallucination [31].

1.4 Research Goal and Questions

The goal of this study is accelerating safety argumentation of ADS in DevOps. This research goal is multidisciplinary, encompassing software engineering, automotive system safety engineering, and DevOps. Moreover, the complexity of ADS and its development process justify breaking down the research goal into more granular objectives. In this manner, it becomes feasible to tackle each challenge individually. At the time of this research, there was no systematic study on the existing challenges and corresponding solutions, making it crucial to first identify the gaps and then propose approaches for them (see Paper B for a detailed discussion B). The following list presents the two research goals and their corresponding research questions:

G1: Identify Challenges, existing solutions, and Gaps.

> **RQ1:** What are the challenges to rapid DevSafeOps of ADS?
>
> **RQ2:** What solutions are proposed in the literature to address the challenges, and what are the gaps?

G2: Identify the root causes for each challenge and addressing them by proposing relevant technical and methodological approaches.

> **RQ3:** What are the root causes of each challenge?
>
> **RQ4:** What technical or methodological approaches can be proposed to tackle the challenges in RQ3?

For G2, we focused on design phase activities, as they are the initial steps in the development of ADS and serve as the foundation for activities in the implementation, verification, and monitoring phases. The research questions in this thesis are addressed in the appended papers. Table 1.1 presents the traceability between the research questions in this thesis and the ones in the appended papers.

1.5 Scope

This thesis centers on the safe design and development process of ADS, specifically identifying the challenges of DevSafeOps and collecting the proposed approaches to address them. The rest of the study focuses on safety analysis methods and requirements engineering as the foundation for a safe-by-design approach. The study focuses on software development, while hardware aspects of ADS are out of scope. Additionally, proposing approaches and methodologies for implementation, verification by testing, validation, and safety argumentation are out of the scope of this study.

The technical aspects of ADS, such as social or legal requirements (e.g., defining safety), sensor technologies (e.g., Lidar vs. Vision, HD Map), fallback strategies, and system design, are out of the scope of this thesis. However, for illustration purposes, some of these aspects are used to clarify the proposed approaches or serve as examples for applying the methods or using the tool.

The study uses LLMs as a tool for safety analysis without delving into their underlying development processes, as the internal development and technical aspects behind LLMs are out of scope. However, relevant aspects such as LLMs' weaknesses, concerns, and limitations are collected to understand their effects on the output and to design appropriate mitigation mechanisms. Legal and ethical aspects of LLMs are also out of the scope of this study, although we reported on the identified concerns.

1.6 Structure of the Thesis

In this thesis, the structure of the remaining chapters is as follows: In Chapter 2, we describe our research design, including the research paradigms and methods employed in this study. In Chapter 3, we summarize the key findings of the included papers and highlight our contributions in the context of the respective research area. Chapter 4 presents a discussion of the research contributions and potential threats to validity. Finally, we conclude our findings and suggest directions for future research in Chapter 5.

The kappa is intended to introduce the PhD research and highlight the key contributions of each paper, offering a snapshot of their significance without duplicating the in-depth findings, analyses, and discussions present in the papers. For further details, including comprehensive evidence and findings, readers are encouraged to refer to the individual papers.

Table 1.1: Presenting the mapping between thesis research questions and appended papers' research questions. An ID similar to RQX.Y.Z is allocated to each research question, where X denotes the sub-goal ID, Y refers to the paper's name, and Z represents the ID of the research question in that paper.

RQs	Contributing Research Questions in Appended papers
RQ1	RQ1.A.1: What are safety-related challenges during field monitoring of AD?
	RQ1.A.2: What are safety-related challenges for continuous development and deployment?
	RQ1.B.1: What challenges are identified in literature when applying DevOps to safety-critical AD functions?
RQ2	RQ2.B.2: What solutions are proposed in literature for these challenges while fostering rapidity and safety?
	RQ2.B.3: What challenges for DevOps in safety-critical AD applications still remain open in literature?
RQ3	RQ3.C.1: What are the challenges of applying STPA in an automotive context with multi-abstraction levels in distributed development for OEM and suppliers?
	RQ3.C.2: What do existing guidelines and papers propose for handling the multi-abstraction level designs in a distributed development environment for each involved party?
	RQ3.E.1: What are the limitations of using LLMs for specifying safety requirements for AD functions?
RQ4	RQ4.C.3: How could STPA be modified to overcome the identified and confirmed shortcomings in RQ2 for each involved party?
	RQ4.D.1: What are the steps in a prompting pipeline to propagate the context needed to generate a HARA?
	RQ4.E.2: What is the task breakdown to enhance the LLMs' performance in specifying safety requirements using HARA?
	RQ4.E.3: How can prompt engineering enhance the LLMs' performance in specifying safety requirements for AD functions?

Chapter 2

Research Design

The methodology defines the process of conducting research, starting from problem investigation, data collection, analysis, and interpretation. Choosing the appropriate research methodology and tailoring it to suit the research question forms the backbone of any research, ensuring reliability, validity, and generalizability of the results.

2.1 Research Paradigms

Knowledge or Design Questions:

One way to classify research questions is into "Knowledge questions" and "Design questions". Knowledge questions investigate the current state of software engineering in a particular sub-area, while the latter focuses on designing better methods, tools, or processes [32].

Academia (contrived) vs. Industry (natural) Setting:

A natural setting studies a phenomenon without altering the real-world environment, whereas in contrived settings, researchers study the subject within an artificial environment (e.g., simulation) [33]. Studies in natural settings contribute to relevance, realism, and richer qualitative data but might lack precision, repeatability, and isolation of variables since the environment is not designed and controlled by the researcher.

The multi-disciplinary nature of software engineering does not solely depend on technological aspects but also includes numerous sociological aspects. Understanding the development and maintenance of such complex and evolving systems requires more than just examining processes or tools; it also involves considering the social and cognitive aspects of the process [32]. To properly understand the complexity of such a subject, the study must be conducted in industrial settings [32]. Otherwise, the proposed approaches may not be suitable for real-world usage. Continuous involvement of industrial practitioners in such research is an essential factor, ensuring the relevance of collected data, applicability of findings, and confirming the benefits for the industry.

Positivism, Constructivism, and Pragmatism:

Positivism, Constructivism, and Pragmatism are the philosophical stances related to this study. Positivism seeks a logical relationship between knowledge and facts, while Constructivism focuses on the human context of knowledge. Meanwhile, Pragmatism evaluates the value of knowledge based on its usefulness for solving real-world problems, introducing a subjective element into the evaluation of results [32].

Exploratory, Descriptive, Explanatory, and Improving:

Descriptive methods aim to depict a situation or phenomenon and are suitable for answering research questions that start with ‘What,’ ‘When,’ or ‘Where.’ Then, explanatory methods are used to address research questions that start with ’How’ or ’Why’ and are aimed to discover causal relationships. When a specific aspect of the subject under study presents challenges, improving methods are employed to improve the relevant elements in that phenomenon [34].

Inductive vs. Deductive:

In the inductive approach, the researcher first gathers data and then constructs a theory based on that data. In the deductive approach, they start with a theory and then validate it using data [34].

Qualitative, Quantitative, or Mixed methods:

The collected data, consisting of words, diagrams, pictures, and descriptions, is categorized as qualitative data, while quantitative data consists of numbers [34]. Quantitative research typically employs statistical techniques on numerical data to examine the relationship between variables, while qualitative research explores human behavior and experience, usually using non-numerical data.

2.2 Prescribed Methods

The research questions are classified based on the categories previously explained and are illustrated in Fig. 2.1. Understanding these categories can better guide us in selecting methods and designing a study. The necessity for a natural setting and the categorization of the main study goal primarily as Pragmatism in this study narrows down the methods to the following:

RQ1, RQ2 & RQ3: The exploratory and qualitative nature of these knowledge questions makes the following two methods suitable candidates:

Expert Interviews: To explore and discover the unknown limitations and challenges in an industrial setting (For RQ1.A.1, RQ1.A.2, and RQ3.C.1).

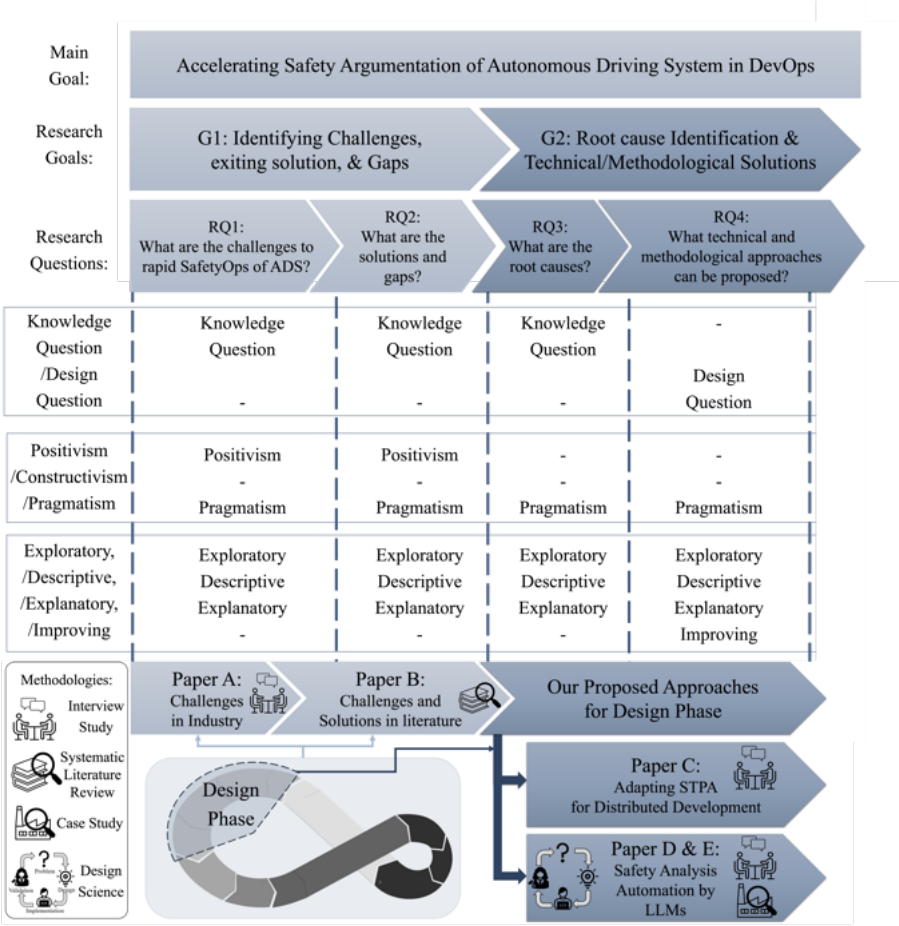

Main Goal:
Accelerating Safety Argumentation of Autonomous Driving System in DevOps
Research Goals:
G1: Identifying Challenges, exiting solution, & Gaps
G2: Root cause Identification & Technical/Methodological Solutions
Research Questions:
RQ1: What are the challenges to rapid SafetyOps of ADS?
RQ2: What are the solutions and gaps?
RQ3: What are the root causes?
RQ4: What technical and methodological approaches can be proposed?
Knowledge Question /Design Question
Knowledge Question
-
Knowledge Question
-
Knowledge Question
-
-
Design Question
Positivism /Constructivism /Pragmatism
Positivism
-
Pragmatism
Positivism
-
Pragmatism
-
-
Pragmatism
-
-
Pragmatism
Exploratory, /Descriptive, /Explanatory, /Improving
Exploratory
Descriptive
Explanatory
-
Exploratory
Descriptive
Explanatory
-
Exploratory
Descriptive
Explanatory
-
Exploratory
Descriptive
Explanatory
Improving
Methodologies:
Interview Study
Systematic Literature Review
Case Study
Design Science
Problem
Design
Validation
Implementation
Paper A: Challenges in Industry
Paper B: Challenges and Solutions in literature
Our Proposed Approaches for Design Phase
Design Phase
Paper C: Adapting STPA for Distributed Development
Paper D & E: Safety Analysis Automation by LLMs

Figure 2.1: Presenting the traceability between research goals, research questions, and the studies performed in the appended papers. The top part presents the research goal and the breakdown into research questions. The middle part presents the categorization of the research questions. All research questions require an industrial setting, are inductive, and qualitative. Suitable methods are employed based on the nature of each research question. The lower part presents the scope and methods of the papers included in this thesis. Papers A and B contribute to identifying the challenges and solutions of all phases in the DevSafeOps infinity loop. Papers C, D, and E specifically focus on the design phase and propose approaches for the challenges in the design phase.

Literature review: To synthesize the literature to collect the challenges identified by other researchers or industrial practitioners, as well as suitable approaches to address them. (For RQ2.B.2, RQ2.B.3, RQ3.C.2, and RQ3.E.1)

RQ4: This research question is categorized as 'improving' and is connected to design questions. Given the nature of the topics and their qualitative aspects, the following methods have been selected as suitable:

Design Science: To facilitate the design of treatments for identified problems in a systematic and iterative manner. (For RQ4.C.3, RQ4.D.1, RQ4.E.2, and RQ4.E.3)

Expert Interviews: To validate the effectiveness of the proposed approaches, experts are asked to review the design and provide comments. (For RQ4.C.3, RQ4.E.2, and RQ4.E.3)

Case Study: To validate the effectiveness of the proposed approach in an industrial setting, the method can be applied in an actual industrial environment. (For RQ4.E.2, and RQ4.E.3)

2.2.1 Systematic Literature Review

A systematic literature review (SLR) is a method for identifying, evaluating, and synthesizing the scientific literature relevant to a research question. Kitchenham et al. [35] provide recommendations for designing the SLR, including the definition of the review protocol, search strategy, inclusion, and exclusion criteria.

In Paper B, SLR is employed to identify the state of the art for our main research goal and to capture existing gaps. Its primary goal is to capture the challenges identified within these papers and then map them to existing solutions proposed in the literature. Paper B serves as a foundation for other parts of our research, where we focus on addressing each identified challenge. Literature review is used as a primary step in other papers as well, although some aspects, such as the documentation of the protocol and review strategy, are simplified. For example, in Papers C and E, queries are employed to capture relevant literature, aiming to identify the state of the art as well as the limitations of the subject under study.

2.2.2 Interview Study

In an industrial setting, especially for novel products and their development aspects, much of the knowledge resides in the minds of experts, as the documentation may not be mature. Therefore, there is a need for a systematic research method to extract this information. During interviews, the researcher engages with the participants and asks questions to collect data on the subject under study. These interviews can be conducted for primary data collection or for the validation of data from other sources [34]. During the design of the interview study, the researcher must make various design choices, including, but not limited to:

a. Type of questions (e.g., open- or closed-ended)
b. Length of the interview, and number of sections
c. Expert's profile (e.g., experts with more than X years of experience in ...)
d. Method of participant selection (e.g., convenience sampling)
e. Data collection tools and environment (e.g., physical vs. online)
f. Number of participants and saturation point

When the interview is designed, a pilot interview can be conducted to assess the effectiveness of the interview design, as well as the relevance, clarity, and structure of the questions. It also helps the researcher to practice managing the session and improve the flow of the interview for the main interview sections. Expert interviews are used in Paper A for exploring, and in Papers C and E for both exploring and validating the designed improvements.

In Paper A, the purpose is to collect insights from industrial practitioners working in various aspects of autonomous vehicle development. These insights are then analyzed and clustered into relevant challenges in DevSafeOps cycles. The study employs open-ended questions to first inquire about the challenges they are facing and, secondly, to ascertain if they agree with the challenges identified by the researchers.

In Paper C, the limitations of STPA are identified, and the method is adapted to better suit automotive system engineering. Expert interviews are used to validate the limitations experienced by the researcher, as well as the effectiveness of the proposed solutions. The study is designed to maintain exploratory aspects, allowing experts the freedom to express their opinions freely and provide answers beyond given options or select hybrid solutions (i.e., open-ended questions).

In Paper E, the effectiveness of the prototype is evaluated by reviewing the outputs, such as the HARA table and safety requirements. These results are forwarded to experts for a detailed analysis and review. After receiving their comments, interviews are conducted to first clarify these remarks and then to ask predefined questions aimed at assessing the quality of the outputs. Finally, the experts are informed about how the outputs were generated, allowing the collection of their suggestions to improve the prototype, thereby adding an exploratory aspect to the study.

2.2.3 Case Studies

Another effective method for studying a phenomenon in its natural setting (such as an industrial environment in this research) is the case study. This approach is particularly valuable when the distinction between the phenomenon and its environment is not clear, leading to a lack of control [34]. The case definition, data collection/analysis methods, data selection strategy, and validity assurance are some of the key elements in designing a case study.

In Paper E, a prototype is designed and implemented for performing HARA using a pipeline of LLMs. The prototype is implemented within an IP-protected service, allowing for fine-tuning of the process for the case company and utilizing real input in an industrial setting. The vehicle function used as input is a novel feature, not previously accessible for training the LLM. Initially, the prototype

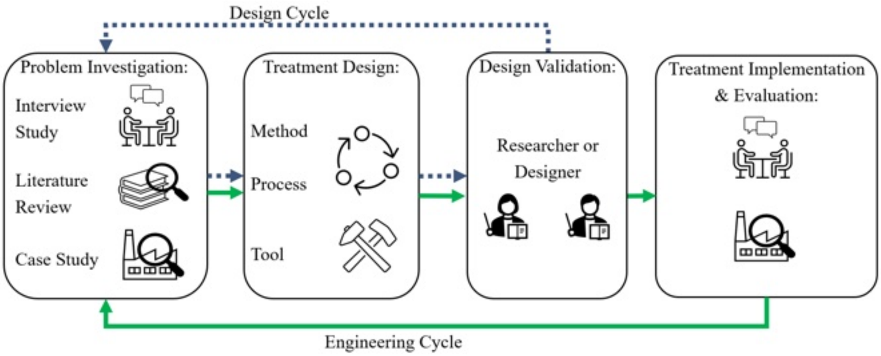


Figure 2.2: Illustrating the Design Science Research Methodology process. The design undergoes iterations within an internal cycle called the Design Cycle. It is then implemented and evaluated as part of a larger cycle known as the Engineering Cycle.

is presented to the team responsible for HARA. Subsequently, four experts are asked to use the prototype and provide feedback on their experiences. The case study serves as a form of triangulation alongside the interview study, as it involves a real function that already has an associated HARA within the company, enabling participants to compare the prototype's output quality against the existing company standard.

2.2.4 Design Science

Design science paradigm enables systematic problem investigation, treatment design and design validation/evaluation to answer design questions. Hevner et al. [36] recommended seven guidelines to be followed in each phase of design science, including problem relevance, design evaluation, research rigor, and design as a search process. Design science is an iterative process comprising two cycles: one internal (the Design cycle) and one external (the Engineering cycle) [37], as shown in Fig. 2.2. The Design cycle is the internal cycle, beginning with problem investigation, followed by treatment design, and concluding with treatment validation. Once validation is satisfactory, the validated treatment is applied in a real-world environment, and its effectiveness is evaluated. The engineering cycle iterates until the evaluation of the treatment in the real world is deemed satisfactory.

In Paper E, Design Science serves as the foundational methodology framework, targeting the iterative design and evaluation of a prototype through interviews and case studies. This iterative process facilitates the continuous improvement of the pipeline, addressing identified weaknesses in the treatment or new LLM limitations discovered through validation and real-world evaluation.

2.3 Methodological Considerations

There are multiple considerations to account for before, during, and after designing a study, depending on the subject under study, the chosen method, and the data collection strategies.

2.3.1 Level of Researcher's Involvement:

The level of involvement from the researcher and participants should be decided in the planning phase of the study, and the study should be designed to adhere to that decision. For instance, in an exploratory case study, the researcher does not have control over the setting, unlike in an experimental study, which requires full control [33].

2.3.2 Generalizability - Universal vs. Particular:

The degree to which the conclusions of a study can be extended or applied to different contexts, such as time or populations, is known as generalizability. Lack of generalizability is one of the threats to external validity which shall be considered during the design. For instance it is a known weakness for case study that they typically report one single case [33]. It is important for the researcher to carefully analyze the external validity of the results and clearly report on the scope and limitations to avoid overgeneralizing their findings. The focus of this study is on safety processes aligned with international standards, and regulations which companies are required to follow to ensure compliance. By aligning the study with these globally recognized frameworks and involving industry experts, the generalizability of the findings can be enhanced. In addition, each study employs further measures to enhance generalizability, such as selecting experts from various companies from both OEMs and suppliers.

2.3.3 Ethical Considerations:

The researcher shall analyze the study with respect to various ethical considerations, and appropriate measures should be implemented to ensure ethical integrity in the research. The lessons learned from more mature fields such as medicine can be adapted to the specific field and used as guidelines, similar to what Strandberg [38] did. He prescribed ethical recommendations and considerations for each step of interview study in software engineering. For instance, consent, confidentiality, and scientific value are among the ethical principles that should be considered. Considering these principles would lead to designing limitations and mitigation strategies, such as anonymization. However, there are also novel and immature aspects, such as the usage of generative AI, that lack consensus on the ethical and legal guidelines to be followed. Transparent discussions among researchers and reporting on these can serve as initial steps to mitigate risks. Employing critical thinking about what could potentially go wrong is another vital mechanism.

Chapter 3

Summary of Included Papers

This chapter is a summary of all included papers. First, the problems they aimed to address are described. Then, the supporting methodology is explained, and finally, the results and contributions are summarized.

3.1 Paper A - An Industrial Experience Report about Challenges from Continuous Monitoring, Improvement, and Deployment for Autonomous Driving Features

Problem: The paper addressed the challenges faced by an OEM in rapid continuous development, deployment, and monitoring in the context of complex and safety-critical applications such as autonomous driving features.
Methodology: The findings were aggregated from existing relevant safety standards, reported experiences, and interviews with eight experts in this field.
Contribution: The paper summarizes the iterative DevOps process for ADS, highlighting its critical role in ensuring safety of ADS, beyond the known advantages observed in other industries. Then, through an in-depth analysis based on industry standards and expert interviews, paper A identifies key challenges associated with the rapid continuous development, deployment, and monitoring of ADS, presenting them in nine clusters. Paper A provides an overview of ADS development, compiling a detailed list of challenges and identifying research gaps, compared to literature [21], [39], [40], which addresses specific challenges.

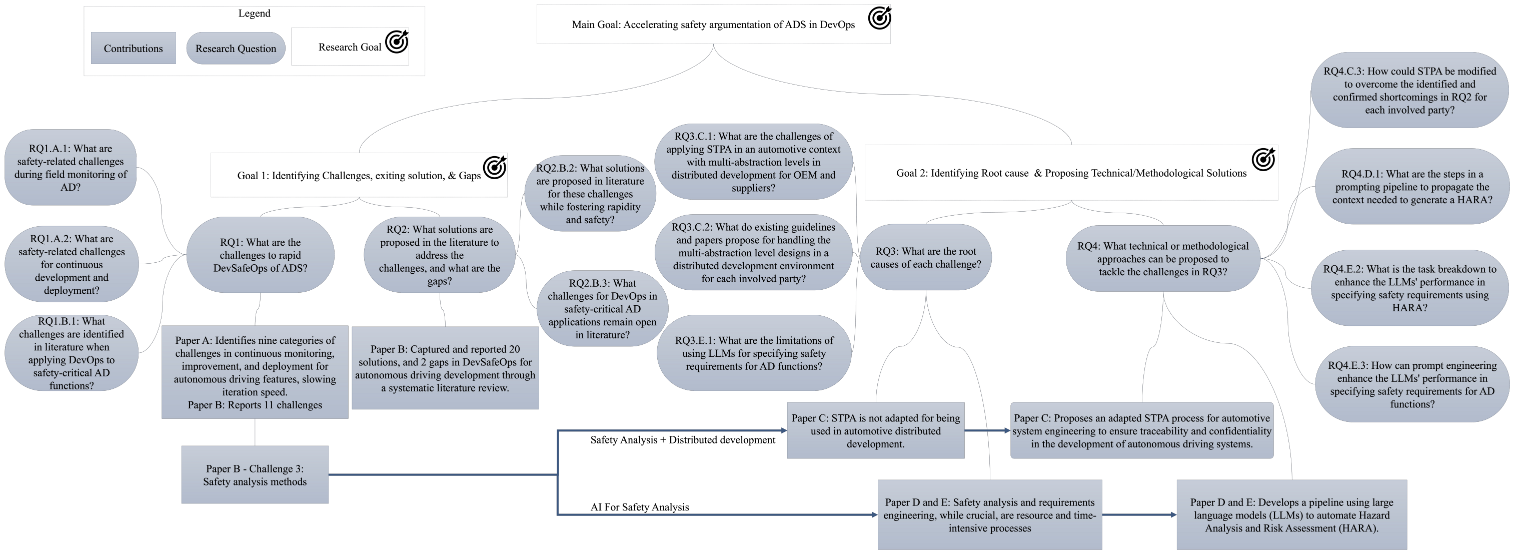

Legend
Contributions
Research Question
Research Goal
Main Goal: Accelerating safety argumentation of ADS in DevOps
Goal 1: Identifying Challenges, exiting solution, & Gaps
Goal 2: Identifying Root cause & Proposing Technical/Methodological Solutions
RQ1.A.1: What are safety-related challenges during field monitoring of AD?
RQ1.A.2: What are safety-related challenges for continuous development and deployment?
RQ1.B.1: What challenges are identified in literature when applying DevOps to safety-critical AD functions?
RQ1: What are the challenges to rapid DevSafeOps of ADS?
RQ2: What solutions are proposed in the literature to address the challenges, and what are the gaps?
RQ2.B.2: What solutions are proposed in literature for these challenges while fostering rapidity and safety?
RQ2.B.3: What challenges for DevOps in safety-critical AD applications remain open in literature?
RQ3.C.1: What are the challenges of applying STPA in an automotive context with multi-abstraction levels in distributed development for OEM and suppliers?
RQ3.C.2: What do existing guidelines and papers propose for handling the multi-abstraction level designs in a distributed development environment for each involved party?
RQ3.E.1: What are the limitations of using LLMs for specifying safety requirements for AD functions?
RQ3: What are the root causes of each challenge?
RQ4: What technical or methodological approaches can be proposed to tackle the challenges in RQ3?
RQ4.C.3: How could STPA be modified to overcome the identified and confirmed shortcomings in RQ2 for each involved party?
RQ4.D.1: What are the steps in a prompting pipeline to propagate the context needed to generate a HARA?
RQ4.E.2: What is the task breakdown to enhance the LLMs' performance in specifying safety requirements using HARA?
RQ4.E.3: How can prompt engineering enhance the LLMs' performance in specifying safety requirements for AD functions?
Paper A: Identifies nine categories of challenges in continuous monitoring, improvement, and deployment for autonomous driving features, slowing iteration speed.
Paper B: Reports 11 challenges
Paper B: Captured and reported 20 solutions, and 2 gaps in DevSafeOps for autonomous driving development through a systematic literature review.
Paper B - Challenge 3: Safety analysis methods
Safety Analysis + Distributed development
AI For Safety Analysis
Paper C: STPA is not adapted for being used in automotive distributed development.
Paper C: Proposes an adapted STPA process for automotive system engineering to ensure traceability and confidentiality in the development of autonomous driving systems.
Paper D and E: Safety analysis and requirements engineering, while crucial, are resource and time-intensive processes
Paper D and E: Develops a pipeline using large language models (LLMs) to automate Hazard Analysis and Risk Assessment (HARA).

Figure 3.1: Presenting the traceability from the main research goal to the detailed research questions and the contribution of each paper. The arrows indicate the connection between the publications, from the identification of challenges to the proposed approaches. Publications A and B contribute to RQ1, while Publication B alone contributes to RQ2. Publications C, D, and E contribute to RQ3 and RQ4. Safety analysis and requirements engineering are the main pillars of a safe-by-design approach and the initial steps in the development of ADS in each DevOps cycle. Therefore, this thesis focuses on methodological and technical solutions for accelerating safety analysis as a stepping stone to the rest of the activities. In Publication C, distributed development is enhanced for the acceleration of the process, while Publications D and E explore the use of LLMs as a tool.

3.2 Paper B - The DevSafeOps Dilemma: A Systematic Literature Review on Rapidity in Safe Autonomous Driving Development and Operation

Problem: This study aimed to identify and cluster reported challenges and relevant solutions to the rapidity of DevSafeOps in the literature.
Methodology: A systematic literature review was performed on DevOps in safe autonomous driving development. A total amount of 181 papers was screened, and 19 were selected based on inclusion and exclusion criteria, then included in data extraction and synthesis.
Contribution: 11 challenges, 20 solutions, and 2 gaps were reported. The reported solutions were then mapped to the challenges. Requirement and safety analysis updates on the left side of the V model, verification and validation on the right side of the V model, and then safety argumentation and certification updates are among the identified challenges. Data-driven development [41], shadow mode testing [42], and dynamic safety cases [43] are some of the proposed solutions for improving the speed of safety activities. Paper B introduces a novel and comprehensive systematic literature review of DevOps challenges specific to safety-critical automotive functions such as ADS, uniquely mapping these challenges to state-of-the-art solutions, which has not been fully addressed in prior literature (cf. Munk and Schweizer [26], Siddique [27], Fayollas [42]).

3.3 Paper C - Using STPA for Distributed Development of Safe Autonomous Driving: An Interview Study

Problem: To ensure the completeness and effectiveness of safety analysis methods such as STPA, the entire system, including all detailed software units and hardware components, must be included in the analysis. However, the inherent complexity of the AD system and the need to protect intellectual property necessitate modularization and abstraction levels, as followed by ISO 26262. Unlike FMEA and FTA, STPA is not inherently designed for multiple abstraction levels.
Methodology: A literature review was conducted to capture state-of-the-art processes and proposals. The identified challenges and the effectiveness of the proposed modifications were validated through a semi-structured interview study with 14 participants.
Contribution: The existing STPA process from the main four guidelines [12], [22]–[24] was reviewed, and a literature review [44]–[48] was conducted. The aggregated STPA process was then adapted and mapped to automotive system engineering, specifically for a complex system such as AD. The proposed process was then tailored for subsystem teams to meet both traceability and confidentiality requirements. First, this involved replacing the activities in step 1

(i.e., defining the purpose of analyses) with a collection of received requirements from steps 3 and 4 of the system team. Second, in step 2 (i.e., modeling the control structure), the process involved encapsulating and abstracting the rest of the system (i.e., out of the scope of supply) as a controlled process. Hence, the proposed adaptations address the challenges of using STPA in distributed automotive system engineering, compared to existing literature [44]–[48] and guidelines (cf. ISO 21448 [12], SAE J3187 [22], STPA handbook [23]).

3.4 Paper D - Welcome Your New AI Teammate: On Safety Analysis by Leashing Large Language Models

Problem: The study investigates the feasibility of using a state-of-the-art LLM as a potential solution to accelerate the safety analysis activities in rapid DevOps.
Methodology: The feasibility of the concept was examined by prototyping and applying it to a specific function, and the results were reviewed.
Contribution: In this study, we designed a pipeline for the LLM by breaking the HARA process into sub-tasks, each handled by an individual LLM. Relevant prompt engineering techniques were then identified and used to create a prompt template for each sub-task. The study demonstrated the potential of LLMs to be employed in performing HARA, which was investigated and validated in detail in Paper E. This paper's contributions propose a framework for fully automating HARA using LLMs for the first time, in contrast to existing literature where LLMs are used as supportive text generation tools without automating the safety analysis (cf. Qi et al. [49], Diemert and Weber [50]).

3.5 Paper E - Engineering Safety Requirements for Autonomous Driving with Large Language Models

Problem: This study investigated the limitations of LLMs for specifying safety requirements for AD functions by performing HARA. It then explored effective task breakdown and prompt engineering to improve the accuracy and efficiency of LLMs in specifying safety requirements.
Methodology: Design science was used in designing the prototype. In the first engineering cycle, the results of the system were evaluated by nine experts from three companies. In the second engineering cycle, after making improvements, a case study at an OEM was performed. A cloud-based internal LLM was employed in the case study, and the responsible team for HARA used the prototype for an internal function.
Contribution: This study builds on Paper D, which represented the first engineering cycle. In this paper, further engineering cycles were conducted by refining the pipeline and its prompts. The results generated by the prototype

for a lesser-known function were evaluated and reviewed by domain experts, and their feedback was collected. Additionally, the overall performance of the models was assessed against ten different Key Performance Indicators (KPIs). In the second engineering cycle, the prototype was first presented, and then the experts tested it, with their reflections reported in the paper. LLM's limitations in this specific task, such as misunderstandings of domain-specific terms and insufficient interpretation of non-textual information such as diagrams or figures, were identified and reported.

Moreover, a threat to the validity of a commonly used experiment (e.g., testing the LLM for AEB [49]), which involves testing the LLM's output for a well-known function against publicly available baselines, was investigated, analyzed, and reported. The final prototype, including the pipeline and the prompt templates, is presented in the paper. The weaknesses and limitations, such as irrelevant or incorrectly formulated requirements in the sample output of the prototype, were reported with examples from review reports. On average, the KPIs were fulfilled in most of the HARA, as reported in detail in the paper. The experts found the prototype useful, although they emphasized the absolute necessity of human supervision and review over the output, which aligns with our goal. This paper introduces a novel, structured approach for engineering safety requirements in autonomous driving using LLMs, validated through a systematic process that contrasts with existing literature, which lacks both comprehensive frameworks and rigorous validation methods (cf. Qi et al. [49], Diemert and Weber [50]).

Chapter 4

Discussion

This section will discuss the contribution of each paper to the main goal of this thesis and the connections between them. Then, identified threats to validity and the mitigation strategies will be presented.

4.1 Contributions

DevOps processes, methods, and tools are well-studied and quite mature, continually improving with industry advancements. However, rapid DevSafeOps remains relatively new, as highlighted in Paper A and B, which sheds light on several research opportunities to confirm the identified challenges and better address them with relevant solutions. Several challenges and their corresponding solutions are identified in Papers A and B, aiming to highlight gaps in the scientific field. For instance, any change in the safety or cyber-security domain might lead to a violation of requirements in other domains. Moreover, as the processes and teams responsible for each domain might be different, it is difficult to analyze and identify the impacts of changes in one domain on the others. Change impact analysis across safety and cyber-security domains is one of the challenges without any proposed solution in the literature (i.e., a gap).

Notably, the challenges and proposed solutions are often supported by a limited number of papers, which weakens the argument regarding the applicability of these solutions for the respective challenges, thus necessitating further research to strengthen these claims. For instance, safety analysis updates in each iterative cycle of DevOps are one of the challenges with only one solution proposed by a single reference [27].

In some proposed solutions, such as the one for requirements updates, the proposed approach can be seen as a solution only for one aspect of the challenge. For instance, "data-driven development" [41], [51] is only applicable to the lowest software abstraction level and not the higher levels, such as the function or system level. Hence, there is a need for further studies to identify new methodological or technological solutions suitable for other abstraction levels and compatible with natural language.

One potential reason for the limited number of papers is the necessity of an industrial setting for both the identification and validation of proposed approaches, making it more challenging without collaboration between academia and industry. Moreover, there is a need for analyzing the published white papers by the main industrial players in this field. However, since these white papers are not peer-reviewed, their inclusion in academic papers might face resistance, as observed through feedback received from reviewers. This shed light on the importance of stronger partnerships between academia and industry to enhance both the applicability and validation of research findings.

For instance, as discussed in paper C, STPA is one of the methods that requires adaptation to handle unbounded complexity and frequent changes in a distributed development ecosystem. However, this challenge was not observed in most papers, and guidelines [23], [24], [45], [46] because, in many research projects, the problem domain was simplified and abstracted. Moreover, real industrial setup constraints, such as intellectual property and liability in distributed development were not considered. In all studied papers on STPA, the analysis was performed on a simplified and abstracted architecture. For instance, most of the papers focus on a vision-only sensor set, with a large black box representing its software. This limitation prevented extending the analysis to apply STPA to all aspects of systems of systems as reported in Paper C. Other limitations, such as different terminologies and missing traceability, along with proposed remedies, are reported and validated in Paper C.

Updating requirements and the corresponding safety analysis method is necessary in each iteration. However, due to the system's complexity, multiple abstraction levels, and the involvement of several internal or external teams, analyzing and identifying all affected activities and requirements is a time- and resource-intensive task. So, existing methods and tools in the design phase should be adapted, or new ones proposed. These proposed approaches need to be carefully analyzed and qualified before they can be used in real-world automotive system engineering.

Automation of design phase activities such as requirements engineering and safety analysis was not among the identified solutions, as they require intellectual capabilities that, so far, only humans have mastered. Initially, in paper D, we performed a feasibility study on the capability of LLMs to be used in the automation of some aspects of design phase activities. In the very initial experiments, the model was asked to perform HARA in a single prompt. The quality, relevance, and correctness of the output in this phase of the experiment were not satisfactory. However, as we broke down the tasks into subtasks and crafted better prompts, the performance of the models improved. Although the system is absolutely not sufficient to be relied upon for providing the design, it might be seen as a teammate whose output needs to be verified and confirmed through review. The weaknesses and limitations of LLM-based safety analysis and requirements engineering are identified and reported in paper E, which need to be taken into consideration when using such a tool in similar safety-critical applications.

While speed is a priority in DevOps, which might sacrifice quality in some cases, it is not a priority in safety-critical systems, and all activities outlined in the safety plan shall be satisfied before the product can be released. The mindsets of the safety and software communities are also different, and they do not readily accept concepts from each other. A lack of deep understanding of each other's concepts, processes, and terminologies, or different priorities, can be some of the potential root causes of the observed resistances. For instance, resistance to adding novel tools and methods to the safety tool-chain has led to outdated methods and tools in the safety tool-chain compared to those in software engineering. Bridging the gap between DevOps and safety is a challenging task that requires not only hard skills but also soft skills.

4.2 Threats to Validity

Feldt and Magazinius [52] proposed the following clustering of potential threats.

Conclusion Validity: Statistical assessment of the effectiveness of the proposed approaches in the current study is challenging, as they are based more on qualitative expert opinions obtained from in-depth interviews with a manageable sample size than on large and objective empirical evidence from systematic and randomized experiments. This is due to the nature of requirements engineering and safety analysis, which requires subjective evaluation by experts who have accumulated years of subject matter experience. Although in both papers C and E, we also gathered some quantitative data to support the conclusions drawn from qualitative data. In Paper C, 11 out of 14 experts recommended our approach, and in Paper E, we asked each expert to rate the quality of the prototype's output.

Construct Validity: As the research goal requires the applicability, efficiency, and effectiveness of the treatments in a real industrial setup, validation shall be done through industrial experts or settings. In papers A, C, and E, this is done through semi-structured interview studies. Mitigation strategies such as reviewing the detailed interview protocols and conducting pilot studies are employed to assess the effectiveness of the interview process before the data collection stage.

Internal Validity: Controlling the environment in an industrial setting is hardly possible, which introduces the risk of the existence of other influential factors on the results for both interviews and case studies. Moreover, the background, knowledge, and experience of participants, who are the main source of data in papers A, C, and E, are influential factors that are not fully under the control of the researchers. Defining the participants' profiles and careful selection of them were applied to reduce these influential factors.

Dependability and Credibility: In paper B, all steps in the SLR are followed and documented to improve the consistency and repeatability of the findings. However, the repeatability of the findings in papers A, C, and E are based on the subjective opinions of industrial experts, which highly depend on their organizational processes and their level of familiarity with the subject, which may be seen as a threat to dependability of these studies. Relying on multiple experts with diverse backgrounds from different companies was employed as a mitigation strategy to improve dependability.

In Papers D and E, LLM is the main influential component of the prototype with respect to output quality. Thus, the stochastic nature of LLM threatens the repeatability of the experiments. Iterative output generation by the prototype and random selection of the samples to be sent for review were employed to reduce the stochastic effect of LLM on the quality of the results. Additionally, due to the immaturity of open-access models at the time of the study, a state-of-the-art commercial black-box model, GPT-4, was utilized. As the model is continuously being improved, the effectiveness of the prompts may vary between versions. However, the pipeline and the processes described in the prompts are generalizable, as they remain unaffected by changes in the model itself.

Confirmability: Although the experts who participated in the studies are transparent and free with their opinions, measures such as recorded interview questions and anonymization of influential factors were used to avoid biasing the participants by the researcher. For instance, in Paper C, the experts were not informed which method was proposed by us and which one was the original, to avoid bias. Similarly, in Paper E, experts were not informed whether the results were generated by an AI or an engineer.

External Validity: Practical aspects in automotive and regulatory frameworks specific to automotive might limit the generalizability of the results of this thesis to other domains. Although the findings and approaches proposed in this thesis can be seen as potential candidates in other domains, they require further validation regarding their applicability and effectiveness. However, the application of the adapted STPA proposed in paper C can also be extended to other safety-critical applications such as avionics, as the complexity, modularization, and confidentiality of each module follow similar constraints as automotive. The generalizability of the findings is not seen as a threat, as international standards in the automotive domain are widely used in the industry, which harmonizes the way of working. Moreover, due to distributed development and its global network, harmonization occurs in other aspects such as techniques outside standards. Furthermore, the reported findings are presented at international industrial conferences and discussed to identify potential weaknesses of the findings for other parties. As LLMs are a fast-evolving field with a non-deterministic nature, the validity of the proposed approaches shall be studied carefully, although they are reported to be potential tools to be considered in system engineering.

Residual Risks: As Wohlin and Rainer [53] identified, research in software engineering is threatened by various factors such as misinterpretation of data by producers or consumers, vested interests and biases, and inappropriate study designs. Although we followed best practice frameworks and guidelines to avoid or mitigate these threats, it is challenging to claim completeness in identifying all possible issues. However, we have conducted a thorough methodological design to ensure that any other threats to validity are reduced and accounted for to the best of our ability. The study designs are provided in detail in the papers so that any potential missing threats to validity can be identified during the peer review process, although according to Wohlin and Rainer [53], the peer review process can sometimes fail to catch mistakes.

Chapter 5

Conclusions and Future Research

In this thesis, we explored the challenges and existing solutions for improving the speed of each iteration loop in the DevSafeOps way of working. In our first contribution, we reported on the clustered challenges and mapped existing solutions to each. Major gaps were then identified and reported. Change impact analysis, safety analysis, requirements engineering, and architectural design are the major activities on the left side of the V-Model that demand significant attention in the safety-critical development process compared to other applications. Change impact analysis is an essential step in safety planning for any change request to identify required activities that affect the safety argument in each iteration of DevSafeOps. After providing a bird's-eye view of the DevSafeOps landscape, we focused on the left side of the V-Model, which represents the first step in the development process. The focus was on design phase challenges and approaches, as the safe-by-design approach is one of the main arguments on safety of the AD, and requirements resulting from this phase serve as the inputs and foundation to the implementation and verification phases.

Traceability Information Model (TIM) is identified as a potential approach in paper B for improving speed in impact analysis. To satisfy both traceability and modularization in activities such as STPA, it is essential to enable multi-abstraction and modularization, which was the focus of paper C. This allows each team to perform the analysis for their module while still being able to track changes across the entire product and maintain coherence with the current version of the product. Traceability is an enabler for the automation of impact analysis, while modularization is an approach proposed for handling the complexity of such a software-intensive system.

In papers D and E, we developed an LLM-based prototype for safety analysis and requirements engineering. We reviewed its performance with the help of experts, and their review comments were reported in paper E.

The continuation of this research toward implementation, verification, validation, and monitoring approaches is the next natural step. This involves developing new methods and tools or identifying and adapting existing ones to ensure their efficiency and effectiveness throughout iterative DevOps loops. Additionally, constructing a comprehensive and maintainable safety argumentation of AD in DevOps iterative loops is crucial.

Future work can also include case studies of the proposed solutions from papers B and C in real industrial setups, reporting on their effectiveness, efficiency, and practical applicability. Moreover, comparing different solutions for specific challenges will be beneficial. For instance, reporting on the strengths and weaknesses of various safety analysis methods can help to select the most effective approach for each use case.

Furthermore, future studies can be built on the identified weaknesses of LLMs in performing safety analysis and requirements engineering, as discussed in paper E, by developing strategies to avoid or mitigate them. This would lead to enhancing the reliability and confidence in using LLMs in safety-critical systems.