Transmitted by the expert from Germany

Federal Office
for Information Security

Informal document **GRVA-21-28**
21st GRVA, 20-24 January 2025
Provisional agenda item 3

Process Guidelines for Derivation and Practical Evaluation of AI Security Requirements in Automotive

BSI TR Version 0.61 (Draft submitted for comments)

Federal Office for Information Security
P.O. Box 20 03 63
53133 Bonn
E-Mail: automotive@bsi.bund.de
Internet: https://www.bsi.bund.de

Table of Contents

1 Introduction 4
2 Scope 6
3 Terms and Definitions 8
3.1 Key Words 8
4 Challenges in Compliance to current Safety and Security Frameworks 12
5 Generic Requirements for AI-Systems 15
5.1 Prerequisites 15
5.1.1 Recommended Sources 15
5.1.2 Recommendation for the Derivation of Generic Requirements 16
5.2 Recommendation for Requirements Elicitation 18
5.2.1 Derivation from existing Standards (ASIL recommendations) 19
5.2.2 Additional Requirements 20
5.2.3 Recommendation Summary 22
5.3 List of Generic Requirements for AI Systems in Automotive 23
5.4 Requirements Structure 27
6 Generalized Audit Approach 29
6.1 Iterative Evaluation Scheme/Audit Process 29
6.1.1 Specification of Generic Requirements 30
6.1.2 Test Criteria 31
6.1.3 Recommended sources for Definition of Values and Thresholds for Specific Requirements and their Test Criteria 32
6.1.4 Recommended Guidance for Setup of the Test-Criteria 34
6.1.5 Challenges within the Process of Defining precise Values and Thresholds 36
6.1.6 Need of iterations and refining of the criteria 36
Prerequisites and Recommendation Summary 37
6.2 Testing Activities 38
6.2.1 Simulation Testing 39
6.2.2 Transition Testing 40
6.2.3 Real-World Testing 42
Appendix A.1 Exemplary Evaluation of AI Requirements based on a Use Case 44
Appendix B.1 Requirement Groups 53
Bibliography 54

1 Introduction

The distribution of AI systems across various domains has transformed industries on the one hand by enabling more complex functionalities in areas such as perception, decision-making and automation, and on the other hand by increasing system performance and efficiency. In sectors like healthcare, AI systems are utilized for diagnostics and personalized treatment plans, while in finance, they optimize trading strategies and fraud detection. In the automotive industry, the distribution of AI systems plays a major role in revolutionizing vehicle design, improving maintenance, usability and cybersecurity. Advanced Driver Assistance Systems (ADAS) are using AI algorithms to interpret data from various sensors, such as cameras, LiDAR, and RADAR, enabling features like lane-keeping assistance, adaptive cruise control, and automatic emergency braking. Furthermore, advancements in this field will progressively bring the industry nearer to the objective of fully Autonomous Driving (AD).

Assessing AI, thereby making use of transparent and practically applicable guidelines and standards, is essential for creating a safe, secure, and trustworthy system, particularly in safety-critical fields like automotive, where system failures can lead to serious harm or life-threatening scenarios. Evaluating AI systems, especially in complex applications, requires a comprehensive approach throughout the entire lifecycle, from model development and deployment to ongoing monitoring and enhancement.

The use of AI is typically accompanied by various challenges related to the topics of safety, security, robustness, and transparency. Qualitatively, models often function as "black boxes," making it difficult to understand, trace and explain decisions and their underlying processes. Addressing this opacity requires designing for interpretability and ensuring that the model functionality is safe and secure [1]. Nevertheless, it is difficult to determine the decision boundaries of a model in order to decide whether the system fulfills its intended purpose or demonstrates incorrect behavior under certain conditions and input.

Additionally, the effectiveness of AI systems is often determined by data quality, model design, and the system's ability to generalize across diverse conditions. As AI systems, especially in AD/ADAS, process massive amounts of data to make predictions and decisions, the computational burden and the need for high processing speeds add further complexity. Ensuring consistent performance across varying conditions and data quality remains a significant challenge, particularly in dynamic environments such as traffic situations. Since formal verification is – in most cases - not a viable option for AI, the creation of evidence often relies on empirical testing [2]. Required parameters for a comprehensive testing, including the characteristics of the test data (extent, scope, etc.) and test criteria still have to be determined on a case-by-case basis as there are currently no established empirical values to reference in this context.

During development and operation, AI systems are exposed to specific risks, including vulnerability to adversarial attacks as well as model and dataset manipulations that can compromise model integrity and performance. For example, evasion attacks might cause an AI model for traffic sign recognition to misclassify objects when confronted with slight, malicious alterations, such as patches or adversarial patterns. Data poisoning of the training data can create backdoors in the model or negatively influence the overall performance. Privacy attacks targeting training data and model may help to extract the manufacturer's intellectual property and can be a starting point for further adversarial attacks on the system [3, 4].

Addressing these vulnerabilities requires the consideration of both existing, domain-specific safety and security requirements as well as requirements specifically tailored to the characteristics of AI. Comprehensive robustness testing across various scenarios to identify failure points is a necessary foundation for verification of the claimed requirements [5]. Achieving safe and secure autonomous driving with AI is only possible by addressing the introduced aspects and developing appropriate evaluation approaches that account for AI-specific factors.

In general, an AI system is accompanied by additional components and (sub-)systems that form an entire system complex embedded in a sensori-motor loop with the environment (see Figure 1). This reflects in potential interaction effects with traditional software and hardware components, i.a., sensors, actuators, and other AI systems, but also in high-level system requirements, e.g., from standardization and regulation, which apply to the entire system and thus equally for both traditional and AI systems.

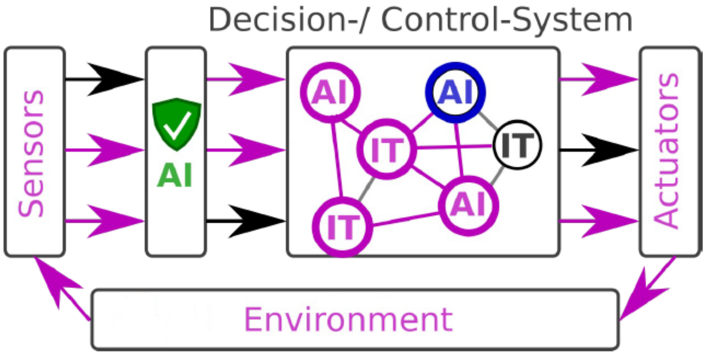


Figure 1: Schematic of an integrated AI-system in automotive. Decision-/Control-Systems are embedded in a sensori-motor-loop with the environment. Classical IT components using conventional software and AI components interact in different configurations depending on the task at hand (here depicted in purple for a fictive task). AI may additionally be used to detect and mitigate attacks via and problems with sensor data (here shown in green). Furthermore, AI may also be used by attackers (not shown).

The existing requirements need to be interpreted for AI systems and implemented in such a way that they can be applied in practice. In particular, the aforementioned AI characteristics must be taken into account. This document is intended to facilitate the step from existing high-level requirements for a system with integrated AI components to the concrete technical testing of these requirements. This process is supported by AI-specific requirements that cover the specifics of AI.

2 Scope

The Technical Guideline BSI TR-[TBD] introduces a comprehensive set of generic AI-specific requirements for the mobility sector with a focus on vehicles. Due to the complexity of the vehicle ecosystem and the numerous related AI use cases ADAS and AD functions in vehicles were used as a starting point for developing and for practically evaluating these requirements in depth. This guideline is designed in a generic and modular way to facilitate the application to other use cases and to facilitate the inclusion of insights from further use cases in future revisions. It outlines a structured approach for deriving and defining AI-specific requirements, along with detailed guidance for specifications and an iterative audit process to evaluate compliance and ensure that such systems are operating safely and securely. The audit requirements focus on both the AI system itself as well as its interaction with external systems, such as vehicle control units, sensors, or external communication interfaces.

Since AI systems aim to replicate or surpass complex human behavior in complex environments it is difficult to explain and often impossible to formally verify these systems. Thus, extensive empirical testing is necessary. The derivation of appropriate thresholds and metrics for their safety and security assurance is a challenge. Consequently, this technical guideline recommends an iterative approach to address this gap for AI-based mobility applications and gives in-depth examples for ADAS/AD systems.

This document presents a set of generic, use case-agnostic requirements designed to address the unique safety and security challenges of AI within the automotive domain. These requirements can be adapted to various use cases and risk levels, ensuring they align with both current and future regulatory frameworks and standards. While systems based on traditional software, including symbolic AI-systems such as decision trees, are already well covered by current regulation and standards there is high demand for an extension to connectionist AI-based systems such as the nowadays widely used deep neural networks (Figure 2). The proposed process shall support the definition of thresholds and identify gaps necessary for testing and auditing AI- systems at a technical level.

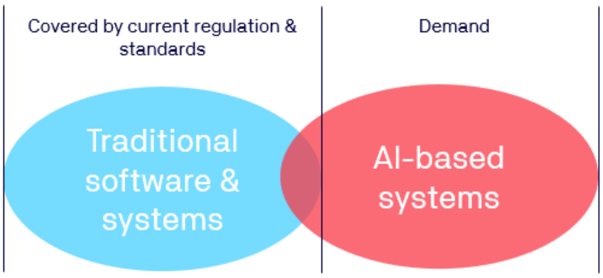

Covered by current regulation & standards
Demand
Traditional software & systems
AI-based systems

Figure 2 Demand for Standards, Best-Practices and Evaluation Methodologies for AI-based Systems.

Given the critical nature of ensuring the safety and security of ADAS and AD vehicles, the generic requirements focus on evaluating the robustness, transparency, and security of the AI's decision-making processes.

The scope of this technical guideline covers:

- Mapping and extension of requirements from applicable automotive safety standards, including ISO 26262:2018 [6] for functional safety, the Hazard and Risk Assessment (HARA) methodology and ISO 21448:2022 [7] for safety of the intended functionality (SOTIF), for AI specific properties.
- Robustness against AI related cybersecurity attacks in accordance with established AI security frameworks and state-of-the-art research.
- A list of generic, use case-independent requirements, adaptable to specific use cases and risk levels.

- Generalized iterative audit approach to standardize and acquire practical knowledge for auditing AI systems, particularly addressing the gap of standardization and established thresholds for safety and security-critical AI systems.
- Transition from generic to specific audit requirements, focusing on exploring methods and sources for defining and selecting thresholds and metrics for AI systems that aim to replicate non-quantifiable human behavior.

The requirements and processes defined in this document specifically focus on verifying the security and safety of the AI component of the system. They do not cover ethical considerations and compliance with relevant data privacy regulations (e.g., GDPR [8]).

3 Terms and Definitions

3.1 Key Words

In the following text, several key words of instructional nature are used. The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119 [9].

- **“shall”**: The following statement is an absolute requirement. Equivalent to “MUST”, “REQUIRED”.
- **“shall not”**: The following statement is an absolute prohibition. Equivalent to “MUST NOT”.
- **“should”**: There may exist one or several reasons why the instruction is only followed partly or not at all. When the referred subject is ignored, a clear justification shall be given. Equivalent to “RECOMMENDED”.
- **“should not”**: Again, when there is one or multiple reasons for performing the stated instruction, then it may be performed when given a clear justification. Equivalent to “NOT RECOMMENDED”.
- **“may”**: Indicates an optional instruction or task.

The following definitions are, if available, based on [10].

Advanced Driver Assistance Systems (ADAS) are a set of safety features in vehicles that assist drivers in controlling the vehicle. ADAS typically include functionalities such as automatic emergency braking, lane-keeping assistance, adaptive cruise control, and blind-spot detection, providing incremental levels of autonomy.

Autonomous Driving (AD) refers to the capability of a vehicle to operate (at least partly) without human intervention using a combination of sensors and vehicle systems to perform the driving task. Fully autonomous vehicles are designed to navigate and make decisions on the road under a wide range of conditions.

AI lifecycle consists of the design and development phase of the AI-based system, including but not limited to the collection, selection and processing of data and the choice of the model and the training process, the validation phase, the deployment phase and the monitoring phase. The life cycle ends when the AI-based system is no longer operational.

Artificial Intelligence (AI) is a set of methods or automated entities that together build, optimize and apply a model so that the system can, for a given set of predefined tasks, compute predictions, recommendations, or decisions.

AI system is a machine-based system that is capable of influencing the environment by producing an output (predictions, recommendations or decisions) for a given set of objectives. It uses machine learning and/or human-based data and inputs to perceive real and/or virtual environments; abstract these perceptions into models through analysis in an automated manner (e.g., with machine learning), or manually; and use model inference to formulate options for outcomes. AI systems are designed to operate with varying levels of autonomy.

AI system component is an element of a system using artificial intelligence. Examples are: image pre-processing filter, image segmentation and traffic sign classification components. There is a trend towards the usage of end-to-end learning of larger AI models that perform complex tasks without intermediate steps, e.g. including pre-processing, image segmentation and traffic sign classification in one AI model.

Automotive Safety Integrity Level (ASIL) is a risk classification scheme defined by the ISO 26262. It categorizes the risk of potential hazards in automotive systems into four levels (A to D) and a category for QM (Quality Management), where ASIL D represents the highest safety requirement. The ASIL level is determined by the severity, exposure, and controllability of risks, ensuring that adequate safety measures are applied to critical systems, such as those used in autonomous vehicles.

Bias is a systematic difference in treatment (including categorization/observation) of certain objects (e.g. natural persons, or groups) in comparison to others.

Black box is a system / software in which the detailed architecture and processing is unknown.

Black/White-box attacks: While a black-box attack is performed without knowledge of details of the structure and parameters of the AI-model that is attacked, a white-box attack uses such information.

Black/Grey/White box testing are tests of systems / software in which architecture and processing is unknown / partially known / known.

Connectionist AI (cAI) systems usually consist of many nodes, called neurons, which are connected with each other in specific patterns, depending on the AI model at hand. Examples of cAI systems are neural networks and support vector machines. In many applications cAI systems are more powerful when compared to sAI systems, e.g. in computer vision. In the majority of cases parameters of cAI systems may not be directly set by the developer. Instead, machine learning algorithms are used together with data to train these systems. The quality of the resulting cAI system is crucially dependent on the quality and quantity of the training data. In contrast to sAI systems cAI systems are in most cases not easily interpretable and not formally verifiable.

Conventional software is usually created by a process called traditional programming. The programmer manually codes rules using a programming language.

Dataset is a collection of data with a shared format and goal-relevant content.

Deep learning is a process whereby neural networks use multiple layers of processing intended to extract progressively higher-level features from data.

Explainability means a property of an AI-based system to express important factors influencing the system’s outcome in a way that humans can understand.

Generative Adversarial Network (GAN) unsupervised machine learning framework using two competing neural networks with the goal to train a model that produces outputs with realistic characteristics.

Generic Requirement refers to a broad, overarching requirement that outlines the fundamental performance, safety, or functional needs of an AI system or autonomous vehicle. It forms the basis for developing more specific requirements, ensuring that the system meets general expectations for operation, safety, or security.

Hardware-in-the-loop (HIL) is a simulation method where real-time testing of the vehicle’s hardware components, such as sensors and control units, is performed in conjunction with software simulations. HIL tests ensure that the system behaves as expected when interfaced with the actual hardware under simulated conditions before deployment in a vehicle.

Hazard and Risk Assessment (HARA) is a structured methodology used in the development of automotive safety systems to identify potential hazards, evaluate associated risks, and define necessary safety measures.

Machine learning (ML) is a collection of data-based computational techniques to create an ability to learn without following explicit instructions such that the model's behaviour reflects patterns in data or experience.

Machine learning model is a computer science construct that generates an inference, or prediction, based on input data.

Model is a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, process or data.

Operational Design Domain (ODD) defines the specific conditions and environments under which an automated driving function, including its AI components, is designed to operate. These include parameters such as weather, road types, traffic conditions, and speed ranges. Ensuring that the vehicle operates only within its ODD is essential for maintaining safety.

Predictability is a property of an AI-based system that enables reliable assumptions by stakeholders about the output.

Reliability is a property of consistent intended behaviour and results.

Resilience is the ability of a system to recover operational condition quickly following an incident.

Road User Detection (RUD) involves the identification and classification of all entities on the road, such as vehicles, pedestrians, cyclists, and animals, using a combination of sensors (e.g., radar, lidar, and cameras) and potentially AI algorithms. Effective RUD is essential for autonomous vehicles to make safe and informed decisions, particularly in complex or unpredictable traffic environments.

Robustness is the ability of a system to maintain its level of performance under a wide range of circumstances. This includes the ability of a system to cope with natural and malicious perturbations within the systems input space.

Safety is the condition of being protected from potential harm or danger, especially in the context of autonomous driving. AI in safety critical applications must be designed, implemented and tested in such a way that safety risks are minimized for the vehicle's occupants, other road users and the environment under all foreseeable conditions.

Security refers to the protection of AI systems, data, and communication channels in autonomous vehicles from unauthorized access, malicious attacks, or breaches. Ensuring robust security measures is crucial to preventing cybersecurity threats that could compromise the safety or operation of the vehicle.

Software-in-the-loop (SIL) refers to a simulation technique used during the development of autonomous driving systems. In SIL, the software components of the system are tested in a virtual environment that mimics real-world conditions, allowing for early validation of algorithms and safety-critical functions without physical hardware.

Specific Requirement provides detailed criteria that a system must meet to fulfill a particular functional or safety objective. For example, a specific requirement for an autonomous driving system employing AI might stipulate the minimum detection range for pedestrians or the maximum latency allowed in decision-making algorithms.

Supervised learning is a type of machine learning that makes use of labelled data during training.

Symbolic AI (sAI) explicitly encodes knowledge using symbolic representations. An example of such a system is a decision tree. Interpreting and formally verifying a sAI system is generally possible and much easier to achieve when compared to connectionist AI systems.

Testing Criteria determine the specific conditions, metrics, and benchmarks used to assess whether a system or a system component, e.g. an AD or ADAS system using AI, meets its specific requirements. These criteria may include performance thresholds, environmental simulations, and safety tests to evaluate the system's robustness, reliability, and safety under different scenarios.

Test Scenario is an evaluation setup designed to assess the ability of the system under test, e.g. an AD or ADAS system, regarding the test criteria. Test scenarios differ across audit stages: simulation utilizes digital images, transition testing employs digital data in a controlled real-world setting, and real-world testing involves physical requisites in actual environments.

Training is the process to tune the parameters of a machine-learning model.

Training data is a subset of input data samples used to train a machine learning model

Transparency of a system is property of a system to communicate information to stakeholders.

Trustworthiness is the ability to meet stakeholders' expectations in a verifiable way.

Uncertainty in AI systems refers to the presence of variables or conditions where the system's output or decision-making may not be fully reliable due to incomplete or ambiguous data. For AD and ADAS vehicles, managing uncertainty—such as unpredictable weather or unclear road markings—is critical to ensuring safe decision-making processes.

Unsupervised learning is a type of machine learning that makes use of unlabelled data during training.

Validation is done to ensure software usability and capacity to fulfil the customer needs.

Validation data is data used to assess the performance of a final machine learning model.

Verification is done to ensure the software is of high quality, well-engineered, robust, and error-free without getting into its usability.

White box is a system / software in which the detailed architecture and processing is known.

4 Challenges in Compliance to current Safety and Security Frameworks

The integration of AI-specific requirements into existing regulatory frameworks and standards is essential for ensuring both the safety and security of AI- systems, particularly in critical domains like automotive, ADAS, and autonomous driving. However, adapting traditional processes to account for the unique properties of AI systems presents significant challenges. The established standards for safety and security, while comprehensive for traditional software, may not directly address the complexities and risks introduced by AI, such as the "black box" nature of models, non-linearity, unpredictable behavior, and uncertain interpretability of results. Therefore, these challenges necessitate the extension and modification of existing regulatory frameworks to adequately capture AI-specific risks.

HARA Process as a Foundation for Security and Safety Requirements

The Hazard and Risk Assessment (HARA) process, defined in standards like ISO 26262:2018 and ISO 25119 [11], provides a structured approach to identifying, evaluating, and mitigating risks in safety-critical systems. HARA focuses on evaluating hazards based on factors like exposure, controllability, and severity, which leads to the derivation of safety requirements, often classified by Automotive Safety Integrity Levels (ASIL). If AI components are to be included in system design and development, a corresponding expert shall be consulted for the process. AI specifics may have influence on the ASIL-determining factors, especially hazard exposure. The following points should be considered:

Lack of Transparency and Explainability (Black Box Nature of AI): Unlike traditional (software) systems, connectionist AI models such as neural networks, including deep learning systems, acts as a “black box”, making it difficult or almost impossible for complex AI systems to trace and explain their decision-making processes. This impedes risk evaluation, the derivation of safety measures and the estimation of the residual risk.

Non-linear, unpredictable Behavior: Connectionist AI models often exhibit an unpredictable behavior to unseen data, making the task of assessing hazards and verifying safety measures complex.

Uncertainty in Testing Results: AI systems, particularly those using connectionist approaches like neural networks, can produce faulty or unintended outputs when exposed to new data, as even minor changes in the input can lead to significant changes in their responses. This uncertainty creates difficulties in determining the significance of test results and whether they sufficiently demonstrate safety or performance.

The following Table 1 summarizes the most significant challenges in aligning AI systems with existing safety and security frameworks.

Table 1: Key challenges in adapting AI systems to traditional safety and security frameworks.

Topic	Traditional Software and Systems	AI-based systems	Challenges in Compliance
Transparency / Explainability	Instructions line by line provide full transparency and explainability However: complex code can still be very hard to read and understand for humans	The functionality of AI systems is often like a "Black Box”.	Difficult to trace decisions and outputs back to specific inputs and define the underlying rules.

Topic	Traditional Software and Systems	AI-based systems	Challenges in Compliance
Verifiability	Established methods for software unit verification available	Often large deep neural networks (DNNs) with no specific software units to verify	Formal verification of the AI model almost unfeasible due to lack of tools and methods
Testability	Established methods for software testing	Uncertain robustness and significance of testing results	Traditional testing methods are less effective and new, AI-specific approaches, are required.
Availability of standards	Standards / Best Practices available, such as ISO 26262, 21448, ASPICE	Standards and frameworks covering AI-Systems within safety and security domains only partially available, such as ASPICE 4.0 with Machine Learning Engineering (MLE) sub processes, respectively missing instructions for adaptation in practice.	Low availability of established best practices makes conventional frameworks hard to apply.

Specific Challenges in alignment with existing standards

AI-specific risks must be integrated into existing safety and security frameworks to ensure compliance. Several existing standards provide valuable foundations, though they need to be adapted for AI systems. The most significant aspects for the most relevant regulation for road vehicles can be listed as follows:

ISO 26262:2018: As a foundational standard for functional safety in road vehicles, ISO 26262 offers a structured HARA process. While it provides guidelines for traditional systems, additional layers must be added to address AI-specific risks, including model robustness, explainability, and interpretability.

ISO 21448:2022 (Safety of the Intended Functionality - SOTIF): This standard focuses on systems where no component failures occur, but hazards may still arise due to inadequate functionality, which is highly relevant for AI. SOTIF helps ensure that AI systems behave safely in real-world scenarios, even in the absence of faults. When implementing SOTIF for AI Systems, the challenge lies in validating that models trained on limited datasets can generalize safely across diverse, unpredictable conditions.

ISO/DPAS 8800 [12]**:** As an emerging standard currently under development, ISO/DPAS 8800 will address safety requirements specific to AI in road vehicles. This standard will fill existing gaps by providing a framework for mitigating AI risks and vulnerabilities, with a focus on safety-critical AI systems. The preliminary scope includes a generic process of risk assessment for both AI and non-AI systems. It does not include testing criteria, specific values, or thresholds.

UNECE R155 and R156 [13, 14]**:** These regulations cover cybersecurity and software updates, ensuring that connected and automated vehicles are protected against cyberattacks. AI systems are particularly vulnerable due to their reliance on large datasets and complex models, which can be targeted for adversarial manipulation. UNECE R155 is critical for embedding AI security measures within a broader regulatory framework, ensuring that AI models remain secure and resilient against attacks.

ASPICE 4.0: ASPICE (Automotive Systems Process Improvement and Capability dEtermination) is a process assessment model derived from ISO/IEC 15504 for standardizing software and system engineering in the automotive industry. ASPICE provides a framework for assessing and improving development processes across system, software, and hardware levels. In version 4.0, ASPICE has incorporated specialized guidelines in the Machine Learning Engineering Process Group to address the unique challenges of ML development. Traditionally, ASPICE primarily addresses deterministic systems, while ML introduces probabilistic behaviors and requires specific processes for handling data-driven development.

5 Generic Requirements for AI-Systems

The embedding of Artificial Intelligence (AI) into vehicles requires compliance to (existing) standards with regard to safety and security. These are crucial to maintain a certain performance, reliability, security and safety of these systems and enable a successful type-approval of the vehicle. This chapter outlines a methodical approach to derive generic requirements for AI- systems in vehicles from existing security and safety standards and regulations, especially considering domain specific regulations and standards. To allow for a deep technical grounding as a first step the primary focus is put on AI functionalities within systems targeting ADAS/AD features. Due to their generic nature the process and the requirements may already be applied in a broader context and future revisions will incorporate insights from other use cases.

5.1 Prerequisites

Before initiating the definition and derivation of generic requirements, several key considerations need to be addressed. A first consideration is identifying and analyzing the relevant domain-specific regulations and standards, which may include (automotive related) ISO standards, UNECE regulations, and industry-specific norms. Additionally, best practices, technical guidelines, and other use-case-specific requirements related to automotive technologies shall be considered.

5.1.1 Recommended Sources

The following Table 2 provides an overview of potential sources for standards, best practices, frameworks etc. with regard to the automotive sector, specifically focusing on AI and AD/ADAS systems. It highlights the scope of each standard, indicating whether it addresses AI-specific concerns, AD/ADAS-specific requirements, or both. It is important to notice that in addition to identifying existing regulations, actively monitoring upcoming regulations is also strongly recommended.

Table 2: Overview of the most relevant safety and security standards, highlighting their relevance to AI and AD/ADAS systems.

Standard	Topicu	Also addresses AI-specific requirements	AD/ADAS specific
Automotive SPICE 4.0 [15]	Process assessment for development processes in automotive.	Yes	No
ISO 26262:2018 [6]	Requirements ensuring the functional safety of road vehicles.	No	No
ISO 21448:2022 [7]	Guidance on design, verification, and validation measures to ensure safety of the intended functionality.	No	Partially
ISO/IEC TR 24028 [16]	Survey on approaches regarding the trustworthiness of AI systems, such as explainability, risks/threats, and their mitigation strategies.	Yes	No
ISO/IEC TR 24029-1 [17]	Background of existing methods for robustness assessment of neural networks.	Yes	No
IEC 61508:2010 [18]	General safety requirements towards electrical/electronic/programmable electronic safety-related systems.	Partially	No

Standard	Topicu	Also addresses AI-specific requirements	AD/ADAS specific
UNECE WP.29 (R155/R156) [13]	Cybersecurity and software updates for connected and automated vehicles.	No	Yes
ISO/DPAS 8800 (Upcoming)	Addressing AI risks in safety-related systems in road vehicles.	Yes	Yes
UL 4600 [19]	Safety standard focused on evaluating the safety of fully autonomous systems, including risk assessment, validation, and safety arguments for autonomous vehicles.	No	Yes
ISO/SAE 21434 [20]	Requirements for cybersecurity risk management in the engineering of electrical and electronic systems within road vehicles, covering the entire vehicle lifecycle.	No	Yes
ISO 25119 [11]	Safety requirements for the development and design of control systems in agricultural and forestry machinery, ensuring functional safety throughout the machinery's lifecycle.	No	Partially

5.1.2 Recommendation for the Derivation of Generic Requirements

In the following, a recommended process utilized to derive generic requirements is introduced. As described in the introduction to Chapter 4, the scope of this document is to provide guidance on defining general safety and security requirements specific to AI and AD/ADAS systems, outlining how to specify these requirements for an actual Use-Case and derive corresponding test criteria. The generic requirements are defined across the entire life-cycle of the system and encompass aspects such as robustness, security interpretability, monitoring, and documentation. One of the most important aspects is to determine the potential risk level of the system. These requirements should take into account the unique characteristics of AI systems, including challenges in explainability and vulnerabilities to adversarial and unseen inputs.

The derivation process begins with a comprehensive analysis of multiple sources (see Table 2) such as norms, standards, technical reports, research studies and other sector related regulation. The most significant source is the ISO 26262 which acts as the foundation, providing extensive coverage of requirements and processes for ensuring functional safety in automotive electronic systems. In addition, the UL 4600 is also used to complement ISO 26262 by addressing additional safety concerns, while ISO/SAE 21434 and UNECE R 155 provide critical security-related guidelines. As a first step, all relevant requirements should be identified and gathered from applicable sources, taking into account the appropriate safety and/or security levels.

5.1.2.1 Identification of risk levels – ASIL categorization

For example, the ISO 26262 focuses on the development of systems, hardware, and software, offering detailed procedures and safety requirements for vehicles. The implementation of these safety measures can vary based on the system's assessed integrity level. A key concept in ISO 26262 is the Automotive Safety Integrity Level (ASIL), which classifies system risk into four dedicated levels, ASIL A to ASIL D. Minor hazards can be categorized as QM (Quality Management) and do not impose any requirements. The corresponding risk level is specified by the potential hazard a system poses. Based on the factors exposure, severity, and the system's controllability, the ASIL is determined according the scheme outlined in Table 3. For example, ASIL D represents the highest risk level, associated with the most severe consequences, highest exposure, and lowest controllability. In cases where Quality Management (QM) methods already mitigate basic risks, it may not be necessary to apply the more stringent methods recommended for higher ASIL classifications.

Table 3 Overview on the derivation of the ASIl classifications.

Severity	***Exposure***	***Controllability***		
		Simple	***Normal***	***Difficult, uncontrollable***
Light and moderate injuries	Very low	QM	QM	QM
	Low	QM	QM	QM
	Medium	QM	QM	A
	High	QM	A	B
Severe and life threating injuries, survival probable	Very low	QM	QM	QM
	Low	QM	QM	A
	Medium	QM	A	B
	High	A	B	C
Life threatening and fatal injuries	Very low	QM	QM	A
	Low	QM	A	B
	Medium	A	B	C
	High	B	C	D

5.1.2.2 Identification of underlying methodologies

A widely recognized and established framework for system development is the V-Model [21], depicted in Figure 3: its stages include system-level design, subsystem integration (architectural and unit design) and code implementation on the left side, and corresponding testing and verification of units, integration and the whole system on the right side. This structured approach facilitates the implementation and verification of safety measures at every level of development. The V-Model ensures systematic traceability from high-level safety and security requirements down to individual components and their corresponding tests, all in accordance with ISO 26262.

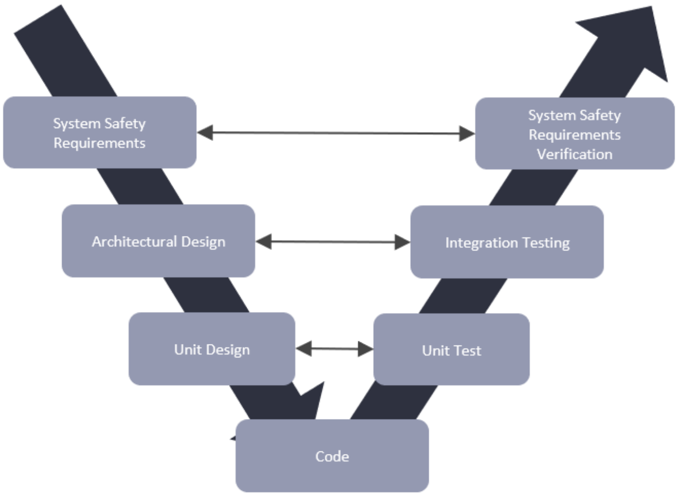
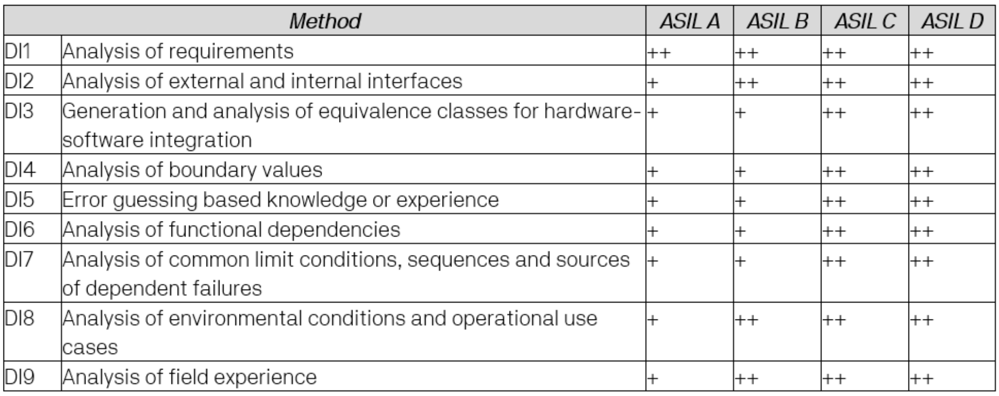

System Safety Requirements
System Safety Requirements Verification
Architectural Design
Integration Testing
Unit Design
Unit Test
Code

Figure 3: Illustration of the V-Model.

For each stage of the V-Model, ISO 26262 defines certain requirements and methodologies with regard to the ASIL level. The rigor and depth of the required activities increase as the ASIL level rises, ensuring that the safety measures correspond to the potential risk.

These methodologies (see Figure 4) range from requirements analysis and boundary condition evaluation to more advanced techniques such as analyzing functional dependencies and error guessing. The recommended methods vary depending on the ASIL level, with higher levels requiring more comprehensive approaches to cover complex interactions, dependencies, and potential failures. For example, while certain methods may suffice for lower ASIL levels, higher levels demand more in-depth techniques to ensure that every critical aspect of the system has been evaluated thoroughly.



	Method	ASIL A	ASIL B	ASIL C	ASIL D
DI1	Analysis of requirements	++	++	++	++
DI2	Analysis of external and internal interfaces	+	++	++	++
DI3	Generation and analysis of equivalence classes for hardware-software integration	+	+	++	++
DI4	Analysis of boundary values	+	+	++	++
DI5	Error guessing based knowledge or experience	+	+	++	++
DI6	Analysis of functional dependencies	+	+	++	++
DI7	Analysis of common limit conditions, sequences and sources of dependent failures	+	+	++	++
DI8	Analysis of environmental conditions and operational use cases	+	++	++	++
DI9	Analysis of field experience	+	++	++	++

Figure 4: Example for ISO 26262:2018 [x] requirement methodologies (cp. Appendix B.1 Requirement Groups for a list of the group abbreviations)..

Subsequently general requirements are extracted from these identified aspects. These requirements shall be comprehensive, covering the most critical elements of safety and security, and sufficiently detailed to enable thorough evaluation and provide adequate coverage across a wide range of risks and threats.

5.2 Recommendation for Requirements Elicitation

When deriving generic and AI-specific requirements, it is crucial to follow a structured process as shown in Figure 5, ensuring that both safety- and security-critical and AI-specific elements are adequately mapped to each other. The previous discussion on the V-Model framework and its alignment with ISO 26262 provides a foundation for understanding how the integration of AI-specific properties can align with existing standards. This is particularly important when identifying the potential risks and threats posed by AI-based systems, such as robustness to unseen data, adversarial perturbations, and interpretability/explainability challenges.

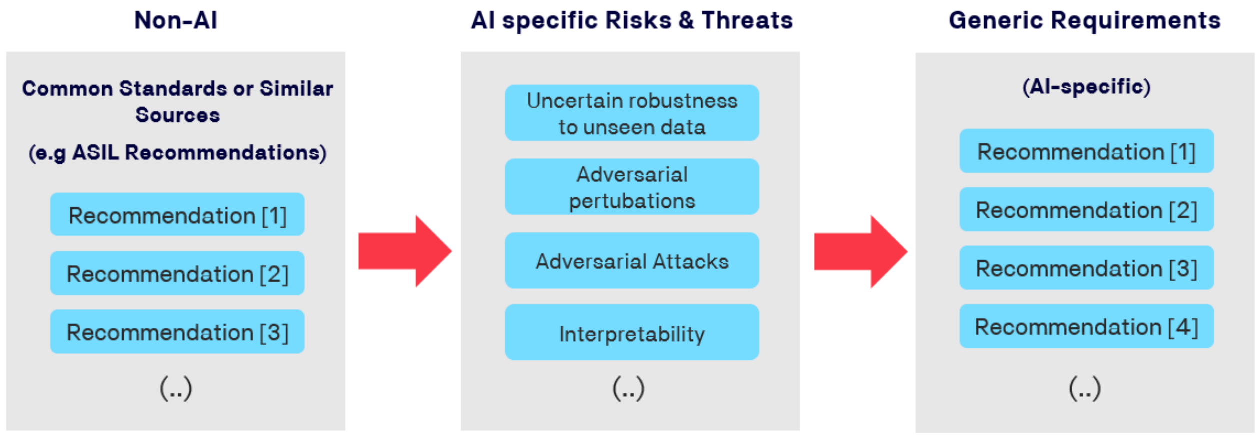
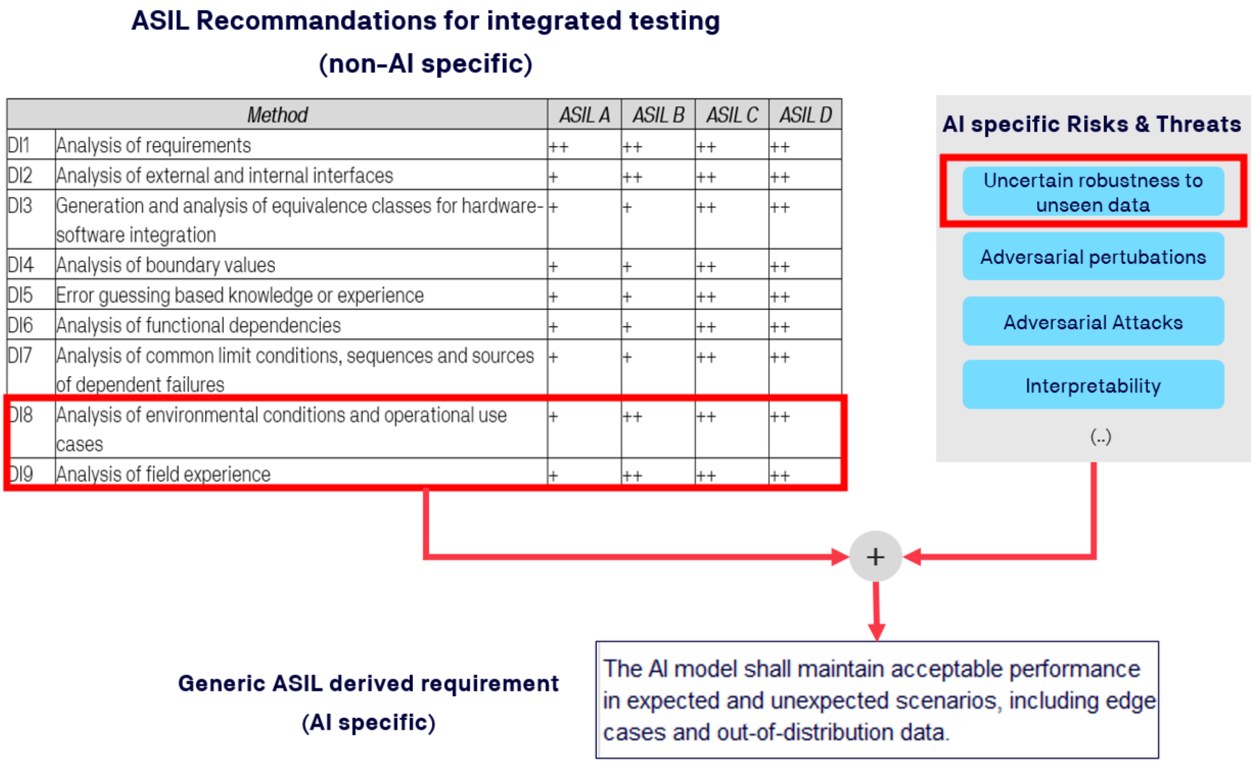


Figure 5: Illustration of the requirements elicitation process for AI-specific generic requirements.

5.2.1 Derivation from existing Standards (ASIL recommendations)

In general, the V-Model approach remains applicable to AI-driven systems. However, the embedding of AI requires additional considerations, particularly in terms of risk mitigation and safety validation, which differ from traditional (automotive) systems. Thus, the mapping of both requirement sources —ASIL recommendations and AI-specific risks and threats— is a critical step in the process. This alignment ensures that established methodologies are systematically addressed to the specific vulnerabilities of the AI system, resulting in new requirements that cover both conventional and AI-driven risks.



ASIL Recommandations for integrated testing (non-AI specific)

	Method	ASIL A	ASIL B	ASIL C	ASIL D
DI1	Analysis of requirements	++	++	++	++
DI2	Analysis of external and internal interfaces	+	++	++	++
DI3	Generation and analysis of equivalence classes for hardware-software integration	+	+	++	++
DI4	Analysis of boundary values	+	+	++	++
DI5	Error guessing based knowledge or experience	+	+	++	++
DI6	Analysis of functional dependencies	+	+	++	++
DI7	Analysis of common limit conditions, sequences and sources of dependent failures	+	+	++	++
DI8	Analysis of environmental conditions and operational use cases	+	++	++	++
DI9	Analysis of field experience	+	++	++	++

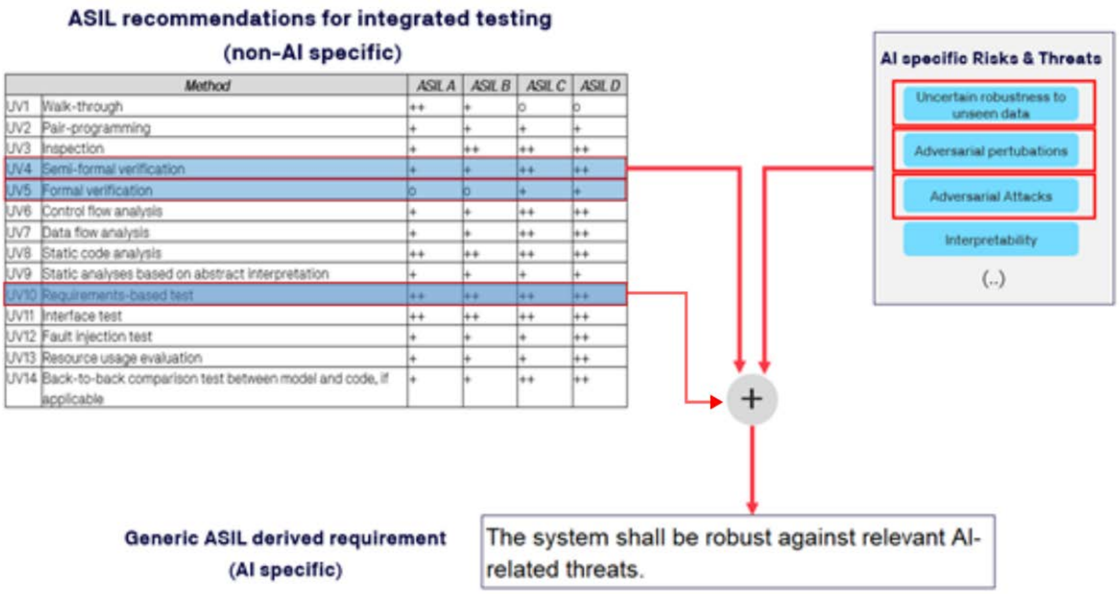
AI specific Risks & Threats
Uncertain robustness to unseen data
Adversarial pertubations
Adversarial Attacks
Interpretability
(..)
+
Generic ASIL derived requirement (AI specific)

The AI model shall maintain acceptable performance in expected and unexpected scenarios, including edge cases and out-of-distribution data.

Figure 6: Illustration of mapping the ISO 26262:2018 recommendations to AI-specific risk and threats in order to derive a AI-specific generic requirement related to the ODD.

The example in Figure 6 illustrates the process of mapping ASIL recommendations with AI-specific risks and threats to derive AI-specific generic requirements. In the first example, the AI risk of uncertain robustness to unseen data is mapped to traditional safety methods, such as DI8[1] (analysis of environmental conditions) and DI9 (analysis of field experience), to generate the requirement that the environmental context must correspond to the operational design domain (ODD). This addresses the AI's ability to function in diverse, real-world environments. The AI-related risk of uncertain robustness to unseen data refers to the challenge AI systems face when encountering input conditions, they were not trained or validated on, which may lead to unpredictable and unintended behavior.

[1] Cp. Appendix B.1 Requirement Groups for a list of the group abbreviations.



ASIL recommendations for integrated testing (non-AI specific)

Method		ASIL A	ASIL B	ASIL C	ASIL D
UV1	Walk-through	++	+	o	o
UV2	Pair-programming	+	+	+	+
UV3	Inspection	+	++	++	++
UV4	Semi-formal verification	+	+	++	++
UV5	Formal verification	o	o	+	+
UV6	Control flow analysis	+	+	++	++
UV7	Data flow analysis	+	+	++	++
UV8	Static code analysis	++	++	++	++
UV9	Static analyses based on abstract interpretation	+	+	+	+
UV10	Requirements-based test	++	++	++	++
UV11	Interface test	++	++	++	++
UV12	Fault injection test	+	+	+	++
UV13	Resource usage evaluation	+	+	+	++
UV14	Back-to-back comparison test between model and code, if applicable	+	+	++	++

Generic ASIL derived requirement (AI specific)

The system shall be robust against relevant AI-related threats.

Figure 7: Illustration of mapping the ISO 26262:2018 recommendations to AI-specific risk and threats in order to derive AI-specific generic requirements related to (semi-)formal verification.

The second example in Figure 7 above highlights the mapping of ASIL recommendations from the category “Unit Tests”, such as semi-formal verification (UV4), formal verification (UV5), and requirements-based testing (UV10), to AI-specific risks like uncertain robustness to unseen data, adversarial perturbations, and adversarial attacks. By aligning these non-AI-specific testing methods with AI-related risks and threats, three new generic AI-specific requirements have been derived.

5.2.2 Additional Requirements

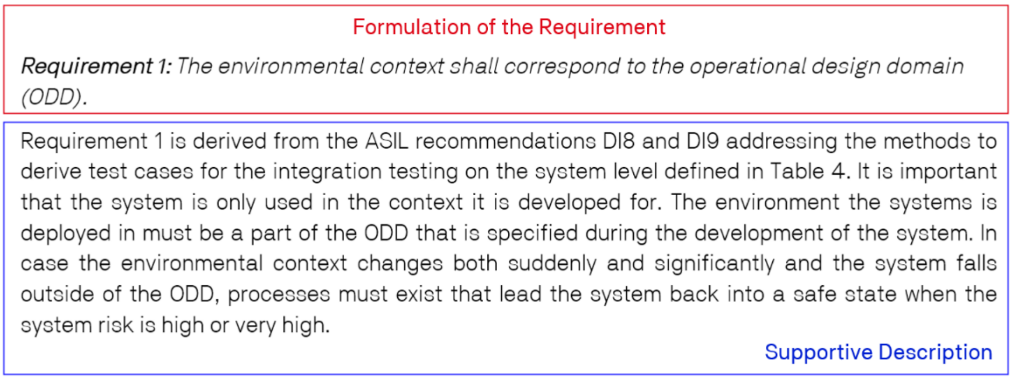
While the prior described structured approach to deriving requirements from existing methodologies (e.g. ISO 26262) provides a solid foundation, it may not fully address all AI-specific aspects for an AI-based system in its intended use-case. As a result, additional requirements may need to be introduced to cover remaining safety and security concerns. This section provides recommendations for the definition of additional requirements. To develop additional requirements, the following aspects should be considered and evaluated for coverage by existing methodologies. If the current methodologies are insufficient or fail to fully address these aspects, new requirements should be introduced. These additional requirements shall be also categorized by a four-level risk and damage classification (low to very high), which directly maps to ASIL A through D. Recommended aspects for definition are:

- **Residual AI model robustness considerations:** Shall be tested against potential threats, with increasing rigor as risk levels rise.
- **Dataset traceability and verification**
- **Dataset coverage and size**: shall be able to represent the operational input domain, with more stringent requirements for higher risks.
- **Datasets must be managed** by a structured approach.
- **Dataset uncertainty and safety verification** should be conducted, with higher priority for higher risk levels.
- Adequate **dataset preparation** is necessary, with more emphasis on higher risk scenarios.

Considering the aspects above, several supplementary requirements can be added to those derived from existing standards to address residual AI-specific risks and threats. This approach will yield to a comprehensive set of general requirements for an AI system, covering all relevant aspects.

5.2.2.1 Recommended Structure of generic AI-specific requirements

To ensure a clear and accurate definition and to offer proper guidance for the requirement's application, it is further recommended to include a description of the requirement's purpose. The following illustration in Figure 8presents an example of a resulting requirement consisting of a generic formulation and a supportive description.



Formulation of the Requirement

Requirement 1: *The environmental context shall correspond to the operational design domain (ODD).*

Requirement 1 is derived from the ASIL recommendations DI8 and DI9 addressing the methods to derive test cases for the integration testing on the system level defined in Table 4. It is important that the system is only used in the context it is developed for. The environment the systems is deployed in must be a part of the ODD that is specified during the development of the system. In case the environmental context changes both suddenly and significantly and the system falls outside of the ODD, processes must exist that lead the system back into a safe state when the system risk is high or very high.

Supportive Description

Figure 8: Example of the recommended structure of a generic requirement.

5.2.2.2 Remark

The entire approach of deriving generic requirements should be considered as an iterative process, necessitating continuous refinement as more use-cases and domains are considered and new AI-related risks, vulnerabilities, and attacks emerge. Additionally, it is highly recommended to monitor and incorporate newly introduced industry-specific or domain-specific standards and frameworks (e.g. the upcoming ISO 8800).

5.2.3 Recommendation Summary

The described process of deriving generic requirements for AI-based systems, particularly in safety- and security critical applications like ADAS/AD, follows a structured approach to ensure a high coverage of both conventional and AI-specific risks. The following illustration in Figure 9 outlines the entire process:

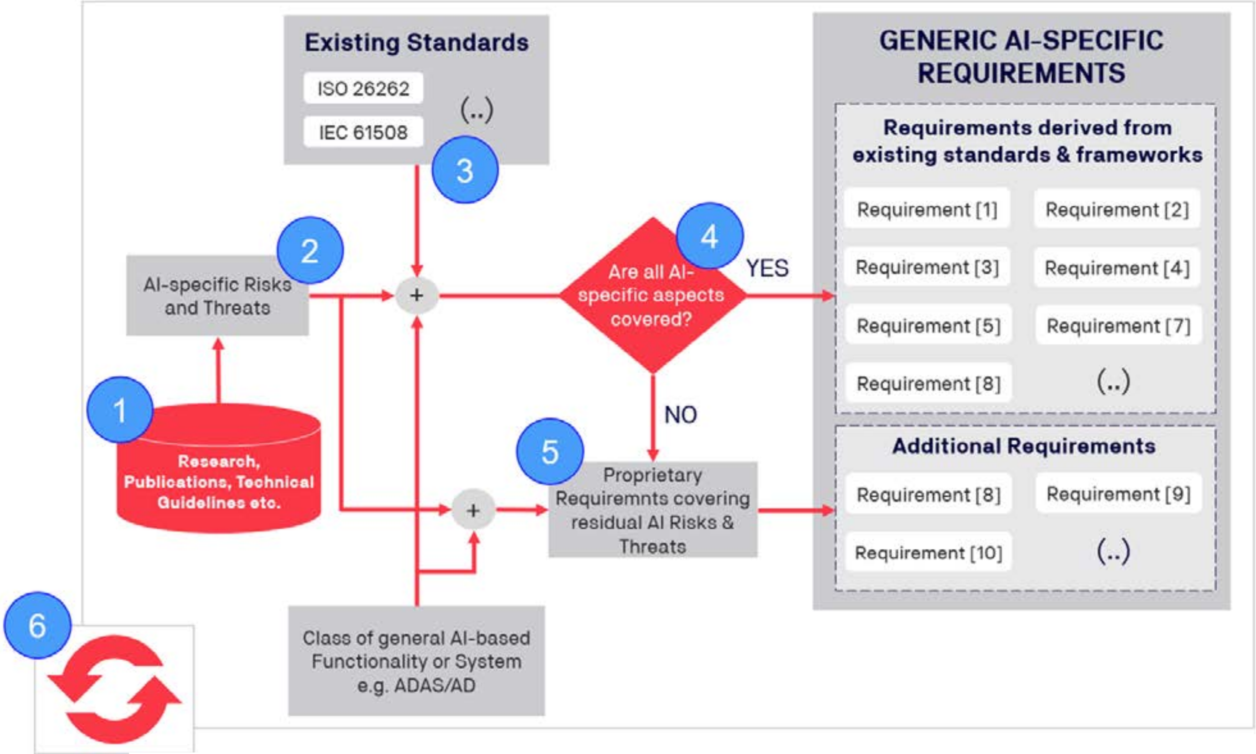

Existing Standards
ISO 26262
IEC 61508
(..)
3
2
AI-specific Risks and Threats
1
Research, Publications, Technical Guidelines etc.
4
Are all AI-specific aspects covered?
YES
NO
5
Proprietary Requiremnts covering residual AI Risks & Threats
Class of general AI-based Functionality or System e.g. ADAS/AD
6
GENERIC AI-SPECIFIC REQUIREMENTS
Requirements derived from existing standards & frameworks
Requirement [1]
Requirement [2]
Requirement [3]
Requirement [4]
Requirement [5]
Requirement [7]
Requirement [8]
(..)
Additional Requirements
Requirement [8]
Requirement [9]
Requirement [10]
(..)

Figure 9: Process of deriving generic requirements for AI-based systems. The most significant steps are numbered (blue circles) and explained in the text.

The recommended most significant steps are:

1. **Analyzing existing Standards and Frameworks:** Begin by utilizing well-established standards, such as ISO 26262, ISO/PAS 21448, and IEC 61508, which offer a solid foundation for defining system safety, functional integrity, and operational safety measures for traditional automotive systems.
2. **Identify AI-Specific Risks and Threats of the Application:** AI systems introduce new risks that aren't fully addressed by traditional safety standards. These risks include e.g. uncertain robustness to unseen data, adversarial attacks, and perturbations, as well as issues around interpretability. These risks must be identified through research, publications, and guidelines relevant to the specific AI application.
3. **Mapping of Non-AI Requirements and Methodologies to AI-Specific Aspects:** To bridge the gap between conventional safety methods and AI-specific risks, non-AI methodologies must be mapped to AI-specific challenges. This mapping involves adapting traditional safety approaches to AI-related problems. Key guidelines for this mapping include:
   a. **Relevance:** Determine if existing methodologies (e.g., formal verification, requirements-based testing) are relevant to AI-specific risks, such as robustness or vulnerability of AI models.
   b. **Adaptability:** Assess how conventional safety testing methods can be adapted to address AI-specific challenges. For example, requirements-based testing can be modified to validate AI models against adversarial inputs or out-of-distribution data.
   c. **Risk Coverage:** Ensure that mapped methodologies sufficiently address both functional safety and the added risks introduced by AI technologies.

4. **Evaluate Coverage by Existing Requirements:** Once AI-specific risks are identified, assess the current set of requirements derived from established standards to see if they comprehensively address these AI-specific risks. This includes reviewing the coverage of AI-related vulnerabilities within traditional safety frameworks.
5. **Introduce Additional Requirements in case of Gaps:** If existing requirements fall short in addressing AI-specific risks, introduce additional, proprietary requirements to bridge these gaps. These supplementary requirements must be derived to cover areas such as dataset robustness, AI model verification, operational domain coverage, and resilience to adversarial inputs. Requirements should also be mapped to specific risk levels, ensuring alignment with the Automotive Safety Integrity Level (ASIL) framework where applicable.
6. **Iterative Refinement and Update:** The process is iterative, requiring continuous refinement as new AI-related risks, vulnerabilities and attacks emerge.

5.3 List of Generic Requirements for AI Systems in Automotive

In the development and evaluation of AI systems within the automotive domain, ensuring security, reliability, and performance is critical. The following high-level requirements (summarized in Table 4) provide a structured approach to assessing key aspects of an AI system's design, operation, and robustness. These requirements cover a wide range of concerns, including system architecture, data integrity, performance monitoring, robustness against threats, and safety mechanisms. By adhering to these guidelines, automotive AI systems can achieve the necessary levels of trustworthiness and safety, ensuring reliable performance under diverse and challenging conditions. Each requirement is assigned to its corresponding life cycle phases, i.e. when the requirement is applicable and effects the system, from ISO 5338 [22]. Additionally, each requirement is assigned to one out of the following 7 general categories as ground pillars for safe and secure AI systems: *Design & Development, Data Management, Performance, Robustness, Monitoring, Explainability* and *Interfaces*. The life cycle and the category assignments help to create a comprehensive and refined overview to elucidate the specific points of application for each requirement and their influenced aspects enhancing the safety and security of the system.

Table 4: List of generic security requirements.

ID	Description	Life Cycle Categories	Category
1	The AI design and development process shall adhere to existing standards and regulations, and it shall be tracked and documented.	Design & Development	Design & Development
2	The system shall implement safety mechanisms to prevent failures of the AI component.	Design & Development, Operation & Monitoring	Design & Development
3	The least complex AI model architecture shall be chosen to limit risks and enhance explainability.	Design & Development, Verification & Validation	Design & Development
4	The datasets shall be managed according to standardized methods, and all key processes shall be well-documented.	Design & Development, Verification & Validation	Data Management
5	The datasets shall undergo quality assessments and be adequately prepared for training and testing.	Design & Development, Verification & Validation	Data Management
6	The AI system shall be developed, tested, and operated within its operational design domain.	Design & Development, Verification & Validation, Deployment, Operation & Monitoring	Performance
7	The AI model shall consistently meet performance requirements.	Verification & Validation, Deployment, Operation & Monitoring	Performance
8	The AI model and system shall be tested against test scenarios created by domain experts.	Verification & Validation	Performance

ID	Description	Life Cycle Categories	Category
9	The AI model shall maintain acceptable performance in expected and unexpected scenarios including corner cases and out-of-distribution data.	Verification & Validation, Operation & Monitoring	Performance
10	The system shall be robust against relevant AI-related threats.	Verification & Validation, Operation & Monitoring	Robustness
11	The system shall monitor and validate the inputs and outputs of the AI model to ensure correctness and safety.	Operation & Monitoring	Monitoring
12	The system shall continuously track AI model-related feedback, incidents and its state during operation.	Operation & Monitoring	Monitoring
13	The system shall provide explanations of the AI model's decisions, particularly for incidents or errors.	Verification & Validation, Operation & Monitoring	Explainability
14	The pre- and postprocessing of the AI model's in- and output shall be suitable.	Design & Development, Verification & Validation, Operation & Monitoring	Interfaces
15	The interfaces between system components and the AI model shall be properly configured, coordinated and designed.	Design & Development, Verification & Validation, Deployment	Interfaces

In the following, the 15 generic security requirements are introduced in more detail. The relation to the requirements and recommendations from ISO 26262 and a short description is given.

ID	Description	Life Cycle Categories	Category
1	The AI design and development process shall adhere to existing standards and regulations, and it shall be tracked and documented.	Design & Development	Design & Development

Based on: MC-family, NU-family, DU-family, UV-family, from ISO 26262

The AI design and development process must comply with all applicable standards and regulations to ensure quality, safety, and legal adherence. This process shall be tracked and documented at each stage to maintain accountability, facilitate audits, and support system improvements. The documentation will include version control for AI model(s) and involved training data, testing results, as well as compliance checks. Industry standards and best-practices, e.g., ASPICE, should be adhered to wherever possible ensuring a transparent and traceable workflow.

ID	Description	Life Cycle Categories	Category
2	The system shall implement safety mechanisms to prevent failures of the AI component.	Design & Development, Operation & Monitoring	Design & Development

Based on: EH-family from ISO 26262

Fail-safe mechanisms and parallel redundant models should be implemented to prevent complete system failures and ensure the system remains operational under adverse conditions, minimizing the risk of critical failure.

ID	Description	Life Cycle Categories	Category
3	The least complex AI model architecture shall be chosen to limit risks and enhance explainability.	Design & Development, Verification & Validation	Design & Development

Additional Requirement (not based on available standards)

The system should select the simplest model architecture capable of solving the task making it easier to explain and verify model behavior.

ID	Description	Life Cycle Categories	Category
4	The datasets shall be managed according to standardized methods, and all key processes shall be well-documented.	Design & Development, Verification & Validation	Data Management

Based on: DV-family from ISO 26262

The origin of datasets should be traceable, verified, and documented to ensure integrity. Proper dataset versioning and tracking of labeling processes are essential to ensure data quality and transparency throughout the AI system lifecycle.

The datasets should be prepared using standardized methods, with clear documentation of the processes involved. This includes dataset characteristics and key processes to ensure consistency and reproducibility across system evaluations.

ID	Description	Life Cycle Categories	Category
5	The datasets shall undergo quality assessments and be adequately prepared for training and testing.	Design & Development, Verification & Validation	Data Management

Based on: DV-family from ISO 26262

Datasets should adequately cover the system's operational input domain. The system must assess and quantify dataset uncertainty, ensuring data integrity and robustness. Additionally, training, testing, and evaluation datasets must be large enough to provide meaningful results and remain independent of one another to avoid bias or overfitting or underfitting, ensuring a robust evaluation process.

ID	Description	Life Cycle Categories	Category
6	The AI system shall be developed, tested, and operated within its operational design domain.	Design & Development, Verification & Validation, Deployment, Operation & Monitoring	Performance

Based on: DI8, DI9, FP3 from ISO 26262

The AI system's environment should correspond to its operational design domain (ODD), which has to be well defined in advance, and the sensor setup must align with the system's development/training setup to ensure reliability and consistency between development and deployment phases.

ID	Description	Life Cycle Categories	Category
7	The AI model shall consistently meet performance requirements.	Verification & Validation, Deployment, Operation & Monitoring	Performance

Based on: DE3, DE4, EH6, ET2, FP2 – 6, IV3, IV4, RS2, RS3, ST3 from ISO 26262

Key performance indicators (KPIs) should be above a defined threshold, performance must meet worst-case error allowances, and the system should ensure reproducibility in real environments. The system should also automatically respond to performance issues when critical errors are encountered after deployment.

ID	Description	Life Cycle Categories	Category
8	The AI model and system shall be tested against test scenarios created by domain experts.	Verification & Validation	Performance

Based on: DU3, DU4, RS-family, FP6 from ISO 26262

Test cases should be derived from knowledge and experience, focusing on error-prone scenarios. The system must also provide explanations for failed tests and errors, ensuring issues are properly understood and addressed.

ID	Description	Life Cycle Categories	Category
9	The AI model shall maintain acceptable performance in expected and unexpected scenarios including corner cases and out-of-distribution data.	Verification & Validation, Operation & Monitoring	Performance

Based on: UV10, DU3, DU4, RS-family from ISO 26262

The system must handle out-of-distribution data, boundary inputs, and corner cases without a significant drop in performance. Defined thresholds must be met even in these exceptional situations, ensuring system reliability.

ID	Description	Life Cycle Categories	Category
10	The system shall be robust against relevant AI-related threats.	Verification & Validation, Operation & Monitoring	Robustness

Based on: MC-family, UV4, UV5, UV10, UV12, IV3, ET2, FP4 from ISO 26262

AI models should use state-of-the-art robustness mitigation strategies. The system must ensure resistance to security and operational threats, providing reliable performance under adverse conditions.

ID	Description	Life Cycle Categories	Category
11	The system shall monitor and validate the inputs and outputs of the AI model to ensure correctness and safety.	Operation & Monitoring	Monitoring

Based on: ED-family, EH6 from ISO 26262

Inputs should be monitored for anomalies before being processed by the model, and outputs should be checked for plausibility during execution.

ID	Description	Life Cycle Categories	Category
12	The system shall continuously track AI model-related feedback, incidents and its state during operation.	Operation & Monitoring	Monitoring

Based on: ED3 - 5 from ISO 26262

Continuous tracking of system feedback and state during operation is essential. All tracked data should be reproducible to allow error diagnosis, correction, and system optimization. System errors should be logged for future analysis and correction.

ID	Description	Life Cycle Categories	Category
13	The system shall provide explanations of the AI model's decisions, particularly for incidents or errors.	Verification & Validation, Operation & Monitoring	Explainability

Based on: DU1, UV10, UV14 from ISO 26262

The AI model must offer explanations for decisions, particularly in boundary values, corner cases, and failed tests. These explanations support the evaluation of requirements and aid in identifying the root causes of system malfunctions.

ID	Description	Life Cycle Categories	Category
14	The pre- and postprocessing of the AI model's in- and output shall be suitable.	Design & Development, Verification & Validation, Operation & Monitoring	Interfaces

Based on: ED-family, EH6 from ISO 26262

Corresponding to the monitoring aspect from Requirement 11, a suitable pre- respectively post-processing shall be present. The components shall ensure system reliability and security, e.g., by correcting unsuited, damaged, or manipulated inputs and outputs.

ID	Description	Life Cycle Categories	Category
15	The interfaces between system components and the AI model shall be properly configured, coordinated and designed.	Design & Development, Verification & Validation, Deployment	Interfaces

Based on: CI1 - 3, IV2, MC6, NU1 – 4 from ISO 26262

All communication, interfaces, and signals between components must be synchronized, with explicit architectural and software unit designs documented to ensure system consistency, reliability, and traceability.

5.4 Requirements Structure

The derived *Generic (AI) Requirements* are not yet applicable and must undergo further specification resulting in *Specific Requirements*, e.g., for the use case and system at hand (see Section 6.1.1). In addition, parameters may be set, especially for technical requirements, in order to be able to carry out concrete testing activities. Finally, precise *Test Criteria* for evaluation (see Section 6.1.2) must be established, on the basis of which it is ultimately decided whether a requirement is fulfilled by the system and/or corresponding processes or not.

This top-down hierarchical approach as illustrated in Figure 10 starts with *Generic Requirements* and their assigned general *Categories* representing their covered aspects. Every category can have multiple *Generic Requirements*.

Note: Not every category needs to be represented by a *Generic Requirement*. The requirements are chosen based on the respective system and the safety and security analysis that shall be conducted.

The specification of a *Generic Requirement* to the use case and the system leads to *Specific Requirements*. The specification of one *Generic Requirement* leads to at least one S*pecific Requirement*. Overall, the set of resulting *Specific Requirements* must be complete in the sense that all objectives required by the *Generic Requirements* are covered. During the specification process, the *Generic Requirements* shall be tailored to the existing use case and the system-under-test. In particular, this means the refining of the requirements by interpreting and defining parameters in the description of the *Generic Requirement.* The formulation of the generic requirements shall be rephrased so all generic placeholders are replaced by use case and/or system-specific definitions.

The next level includes the *Test Criteria*. Again, a *Specific Requirement* has a minimum of one, but can have multiple *Test Criteria* assigned. The evaluation of the *Test Criteria* decides on the fulfilment of the *Specific Requirement.* A *Test Criterion* is not rigid, but is initially defined and then, if necessary, iteratively adapted in the course of the audit process, taking the existing use case and the system behaviour into account.

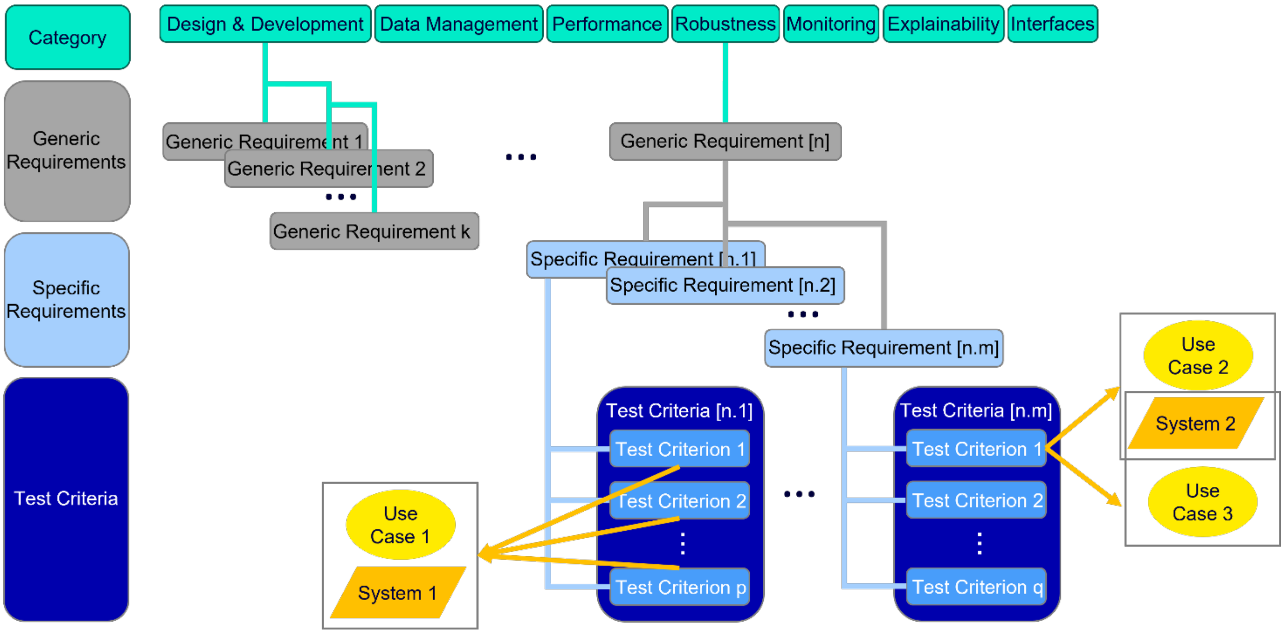


Figure 10: Hierarchical structure of the requirements for AI systems.

Once defined, the final *Test Criteria* can be applied for the requirement evaluation. A defined *Test Criterion* may not just be valid for the use case and system it was designed for, but can also be applied to other, similar use cases and systems. This reuse has to validated by a dedicated requirement derivation process as described.

6 Generalized Audit Approach

This chapter establishes a general audit approach and evaluation process alongside the introduced hierarchical requirements structure from Section 5.4. In order to transition from the generic audit requirements, outlined in Section 5.3, which are independent of specific use cases and technical aspects such as attacks or metrics, to specific requirements tailored for auditing AI systems, a generalized audit approach is proposed. Due to the lack of standardization, the additional lack of experience and practical knowledge in setting appropriate thresholds or attack parameters for safety and security critical AI systems, this document recommends an iterative approach to close this gap for AI in vehicles with a focus on AD and ADAS systems.

Since for more established technologies, thresholds and measures for safety and security are generally agreed upon due to standardization or due to being physically quantifiable. However, AI systems, which aim to replicate or surpass complex human behavior in complex environments, present a challenge because this behavior is difficult to quantify. As a result, deriving specific thresholds for AI system testing is not straightforward.

Applying this audit iteratively to comparable use cases will produce a generalized set of specific requirements, including test criteria, for a group of use cases.

6.1 Iterative Evaluation Scheme/Audit Process

To derive common specific requirements and test criteria for comparable use cases, AI components and risk levels, it is recommended to apply the generalized audit approach iteratively for specific use cases. The audit approach is depicted in Figure 11.

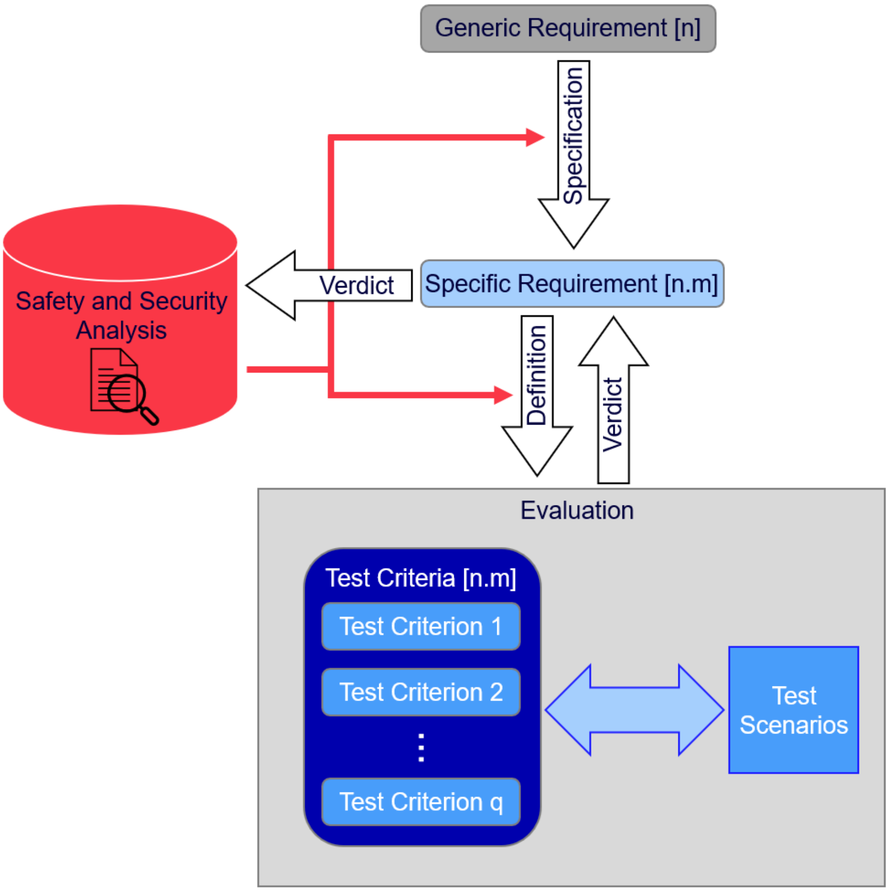


Figure 11: Audit process from generic requirement over evaluation to verdict.

Similar to the established process in the automotive domain, safety and security analyses for the AI component and the entire system using established methods as suggested in for example ISO 26262 and the upcoming ISO/DPAS 8800 standards should be performed. These analyses should be conducted by domain experts and experts with relevant knowledge of both the use case and AI technologies.

Based on these safety analyses, relevant generic requirements outlined in Section 5.3 should be selected. It should be considered whether specific faults, errors, or attacks could occur, lead to hazards, and negatively impact the system or the user. Resulting in a tailored set of relevant requirements for the use case. The generic requirements should be adjusted with details specific to the AI system, resulting in a set of specific requirements for the use case under test. The safety analyses also help to derive the necessary test criteria including thresholds, attacks, or metrics to allow testing at a technical level and quantify the residual risk of the system.

This results in a set of specific requirements and associated test criteria for the use case. Tests are then conducted across various abstraction levels to provide a pass/fail verdict for each specific requirement based on the defined test criteria.

To derive sets of standardized specific requirements and thresholds, more practical knowledge regarding AI safety and security in automotive use cases has to be gained. This is done iteratively by conducting the audit process on a wide range of comparable use cases, AI components and risk levels. Resulting in practical knowledge on common errors, suitable metrics, and thresholds. Which in turn will create sets of specific requirements and test criteria and test methods for a diverse set of automotive AI-systems.

The outcome of applying this audit process iteratively will contribute to the standardization of specific requirements, test criteria, and procedures for comparable use case classes.

6.1.1 Specification of Generic Requirements

The generic requirements represent high-level requirements, comparable to those formulated in existing standards and best practices. They address security-relevant topics that need to be considered for the development, deployment, and operation of a secure AI systems (see Figure 12). Their technology-independent nature facilitates a better understanding of their content by non-technical personnel and simplifies the integration into an overall safety and security management.

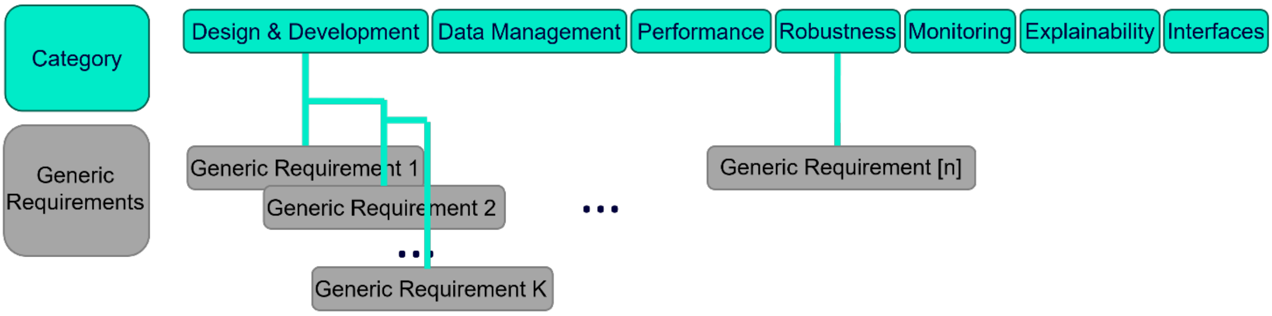


Figure 12: Generic Requirements and AI safety and security categories.

On the other hand, this means that the generic requirements do not explicitly address the actual use case and the system to be tested. In some cases, the generic requirements are specific enough to be applied directly. In others, they need to be tailored to the existing use case and the system at hand.

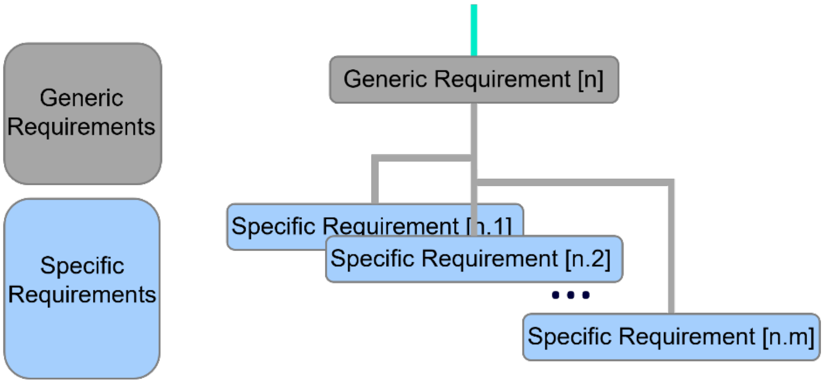

Generic Requirements
Specific Requirements
Generic Requirement [n]
Specific Requirement [n.1]
Specific Requirement [n.2]
• • •
Specific Requirement [n.m]

Figure 13: Generic Requirement specification to Specific Requirements.

The resulting specific requirements (seeFigure 13) include use case- and system-specific information and represent an important intermediate step towards the desired technical evaluation of the requirements. The process ensures that the requirements are not only generic but adapted to the specific threats, operational constraints, and performance demands of the AI system in question. The following subjects, among others, shall be considered during specification:

- AI system characteristics, such as performance, model architecture or input and output.
- Operational environment of the system and expected inference data.
- Internal components of the vehicle (e.g. redundant systems)
- Development process including used training datasets and hyperparameters.
- Risks and threats (e.g., identified during HARA) and their impact on the system.
- Corner cases of the use case or the AI system.
- Additional features and accompanying components or systems related to the AI system, e.g., pre- and postprocessing, explainability methods or monitoring and logging functionality.

The above aspects tailor the generic requirements to the specifics of the AI use case and system, ensuring their implementability and relevance.

6.1.2 Test Criteria

To ensure testability/auditability, test criteria shall be established. These criteria provide a technical foundation for the adapted specific requirements, functioning as strict conditions that serve as the basis for evaluating the system. They outline the criteria that shall be met by the system, either directly or through relevant documentation or processes, to fulfil the requirement.

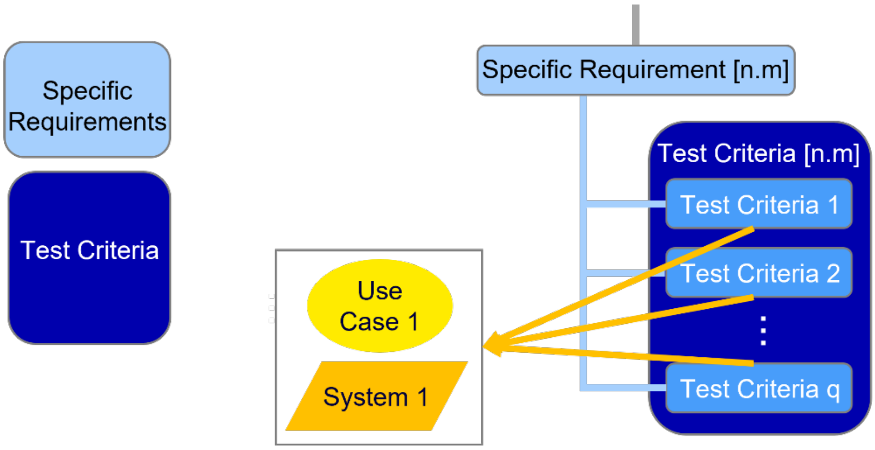


Figure 14: Test Criteria for a Specific Requirement.

A requirement consists of at least one criterion but can include multiple criteria that need to be satisfied as illustrated inFigure 14. Generally, test criteria can be classified into two different directions:

Quantitative Criteria: These involve specific threshold values for measurable parameters related to the system's functionality, reliability, usability, performance, and security. Typically, the verification of these thresholds is done through empirical testing.

Qualitative Criteria: These criteria are not measurable or expressible in numbers. This applies particularly to completeness verification, where it shall be ensured that all necessary components for proper operation or evaluation are in place (e.g., through document review). Additionally, qualitative criteria may require expert assessment in specific domains.

In practice, a combination of both quantitative and qualitative criteria is often used to evaluate a requirement. When it is not possible to define quantitative criteria (due to a lack of regulatory or normative thresholds or empirical data), or if they are only partially formulated, a mix of criteria is employed. For example, an adversarial attack might be tested by selecting a particular attack type, defining a realistic effort that an attacker might make, and testing the system's security against it. While the attack's impact can be measured empirically, estimating the effort, and selecting the attack method relies, at least partially, on qualitative expert judgment. The same applies to setting specific thresholds, where risks shall be assessed, and prior experience or expert insights are factored in.

Alongside the test criteria, a test approach shall be developed, including corresponding test scenarios. These test scenarios shall be designed so that their results can be assessed against the defined criteria, leading to a judgment on whether the system meets the specific requirements or fails to do so. More information on test scenarios and testing in general can be found in Section 6.2.

To implement the test criteria and the corresponding test scenarios on a technical level, several parameters shall be defined. These parameters depend on the formulation of the test criteria and the specific requirement, including coverage of the use case and AI system. Most parameters definitions concern both the qualitative and quantitative aspects of a criterion and typically involve:

- Defining the subject for completeness verification.
- Establishing metrics for measurement and evaluation.
- Setting thresholds in line with the defined metrics.
- Determining the content, type, and sizes of test datasets, i.e., a minimum number of data samples to ensure statistical relevance of the test results.
- Including additional elements based on the type and formulation of the test criteria.

The definition of these parameters may be derived from relevant standards, norms, regulations or technical guidelines. If such sources are unavailable, empirical data, domain expertise, and best practices should be used to guide the definition process. This includes the use of results from comparable use cases and systems, and corresponding security considerations. It is recommended to provide a clear justification of why the specific parameter value was chosen including the basis on which the decision was made, e.g., by stating statistical proof or reference to proven test criteria of similar system evaluations. In the following, recommended guidelines and sources are described how to define such parameters and thresholds.

6.1.3 Recommended sources for Definition of Values and Thresholds for Specific Requirements and their Test Criteria

Suggested sources to support the definition process may include input and alignment from real-world feedback, simulations (Hardware-in-the-Loop/Software-in-the-Loop), and benchmarks of human performance. In detail:

1. **Natural and Adversarial Perturbations**

Possible Aspects for Threshold Definition:

Perturbation Magnitude: Rather than providing a fixed (precise) threshold for perturbation magnitude (e.g., L2-norm or L-infinity norm), the acceptable limits must be derived from domain-specific testing. For example, the maximum distortion of the input data, which can be tolerated while maintaining a specified accuracy (e.g., 94%) needs to be determined based on empirical tests against known adversarial strategies.

Adversarial Success Rate: A precise adversarial success rate threshold is impractical in a generic approach. Instead, thresholds should be defined based on the system's deployment context and adversarial threat models (e.g. black-box scenario with limited attempts etc.). For example, an acceptable success rate might be defined as allowing no more than 5% of attacks to lead to incorrect predictions, depending on the criticality of the application.

Human Perceptibility: Thresholds related to adversarial robustness may be aligned with human recognition benchmarks. For example, the system should maintain a certain robustness against adversarial perturbations (e.g. as patch-based attacks) that are clearly identifiable as suspicious manipulation by a specific group of human observers.

2. Feedback from Real-World Evaluation

Possible Aspects for Threshold Definition:

Error/Failure Rates: Instead of setting fixed error/failure rates, thresholds should be dynamically or iteratively set based on real-world operational feedback. For example, acceptable error rates for pedestrian detection or traffic sign recognition should be based on the system's deployment in diverse real-world conditions. The data gathered from these real-world scenarios may (iteratively) adjust the acceptable level of system performance under real-world constraints.

Environmental Variability: In some cases, the environment (e.g., lighting, contrast, weather) may impact performance, making fixed thresholds unfeasible. Instead, thresholds for accuracy or reliability under specific conditions (e.g., low-light environments) can be derived from operational data and feedback over time, rather than being defined and fixed generically.

3. HIL/SIL Testing

Possible Aspects for Threshold Definition:

Simulated Environment Accuracy: Possible thresholds related to system accuracy during simulations may be set initially and may be adjusted based on testing results in simulations that mimic real-world conditions. For example, minimum acceptable accuracy levels in a traffic simulation can be set based on HIL/SIL test results.

Robustness under Simulated Stress: The threshold for system stability during simulated edge cases (e.g., high traffic density, sudden braking scenarios) should be determined through repeated tests in simulation environments, rather than by applying fixed, generic values.

4. Human Performance Benchmarks

Possible Aspects for Threshold Definition:

Human Error Rates: Instead of applying a fixed human error rate as a threshold, human performance data from relevant tasks can be used as a baseline or initial guidance. For instance, the system should aim to match or exceed human accuracy in traffic sign recognition, but specific thresholds must be defined based on human performance studies in similar contexts.

Comparability to Human Judgment: Possible thresholds for AI systems may aim to approach human-level decision-making in complex scenarios (e.g., pedestrian recognition), but this should be quantified in specific scenarios relevant to the system's operation, rather than through a "one-size-fits-all" approach.

5. Setting Initial Values and Thresholds Based on Evaluator's Choice and Experience

Possible Aspects for Threshold Definition:

Expert Judgment: Evaluators can set provisional or initial values using industry knowledge, regulatory standards, or domain experience to begin the evaluation process. These initial thresholds may provide a starting point, which can be refined as testing progresses.

Applicability Testing: After setting an initial threshold, evaluators can conduct pilot tests to assess if the chosen values are reasonable and relevant. For example, an initial threshold for error rate could be established based on industry standards or domain related experience and then adjusted as more system-specific data becomes available.

Iterative Refinement: Initial values should be reviewed and modified as insights emerge. Evaluators should justify adjustments based on test results, industry developments, and feedback from ongoing evaluations. These iterative cycles may be repeated until a satisfactory level of reliability or other benchmark is achieved.

6.1.4 Recommended Guidance for Setup of the Test-Criteria

Once a specific requirement is defined, the next step is to determine precise values and thresholds for the associated test criteria. These values and thresholds serve as benchmarks to assess the requirement (e.g. quantifiable resilience against identified vulnerabilities etc.). In the following, a structured process is suggested that provides guidance on selecting relevant sources and parametrizing values and thresholds.

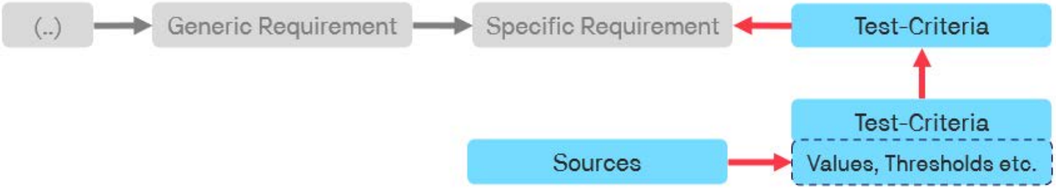
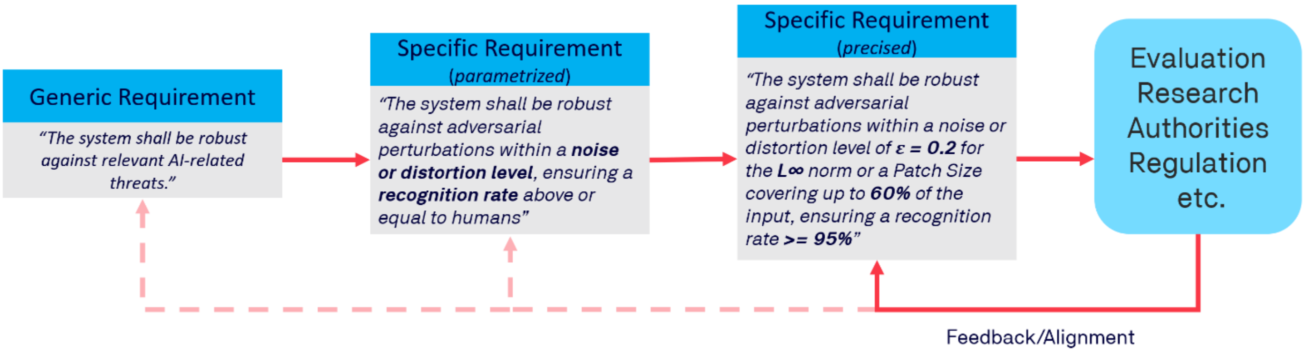


Figure 15: Final stage of the specification process including definition of precise Test-Criteria

The table below addresses the most important aspects of setting precise, context-specific values and thresholds for test-criteria. Key tasks include: translating high-level requirements (e.g., robustness against perturbations) into measurable metrics, selecting data sources that reflect real-world and simulation conditions, aligning thresholds and values through empirical tests. This process is considered as a guidance to support a structural step-by-step approach.

No.	Step	Task	Description
1	**Parameterize the specific requirement**	Contextualize for use-case and operational domain	Define the parameters of the specific requirement, e.g. (adversarial) noise level, lightning conditions, no. of malicious attempts, success rate etc.
2	**Identify and Select relevant sources**	Choose relevant sources and metrics applicable to the specific requirement and the associated parameters	Select controllable benchmarks and parameters of relevant source. E.g. for **Adversarial Perturbations:** Epsilon, Patch Size, Iterations, Success Rate, Step Size, Vector Norm Constraint **HIL/SIL Testing:** Latency, Sensor accuracy, Systems' noise, error rates etc. **Feedback from real world evaluation:** Error rates, response times, camera noise levels etc. Human Performance: Error rate baseline of a group of people (e.g. from trials) **Evaluators Choice:** Domain-specific benchmarks from current standards, initial values form pilot testing, industry averages etc.
3	**Setup**	Define precise, measurable metrics, align the metrics across sources	Setup initial values and thresholds of the parameters, ensuring alignment across other sources. E.g. define the adversarial perturbation noise level with an Epsilon value of 0.2 or a Patch Size covering 60% of the input, aiming for an error rate comparable to human performance or within the limits of human interpretability. An additional source to consider could be input from domain experts or domain-specific standards.
4	**Feedback and Refinement**	Testing the Criteria, gathering feedback	Conduct (iterative) evaluations of the test-criteria and the defined thresholds and values across actual test (e.g. real-world within adversarial scenarios,) and gathering feedback on AI-Systems performance. Adjust the parameters accordingly.



"The system shall be robust against relevant AI-related threats."

"The system shall be robust against adversarial perturbations within a **noise or distortion level**, ensuring a **recognition rate** above or equal to humans"

"The system shall be robust against adversarial perturbations within a noise or distortion level of **ε = 0.2** for the **L∞** norm or a Patch Size covering up to **60%** of the input, ensuring a recognition rate **>= 95%**"

Figure 16: Recommended process of specifying generic requirements through parametrization, precise values and feedback.

The illustration above shows the basic process of specifying a generic requirement by precise values. This process involves parametrization, setting up specific thresholds and value, receiving feedback, and refining based on outcomes gathered from various sources such as evaluations, research, or regulatory and authority feedback. It has to be noted, that this feedback may also affect the prior stages including the generic requirement itself. E.g. the feedback might necessitate adjustments or alignments to the generic and/or specific requirements, not just the precise parameter values. However, for the sake of clarity, the focus here is limited to the impact on the (precise) specific requirement.

6.1.5 Challenges within the Process of Defining precise Values and Thresholds

The suggested sources can be seen as supporting recommendation for the process of defining values and thresholds. However, setting fixed (precise) values and thresholds (e.g. for the amount of (adversarial) noise, patch-size, acceptable robustness etc.) for testing AI-based systems is highly challenging and often unfeasible due to several critical factors:

Huge Diversity and Variety of Applications and Systems: AI-based systems differ significantly in terms of architecture (e.g., deep neural networks, convolutional neural networks, decision trees, etc.), functionality, and deployment environments (superior system, system interaction, etc.). A fixed threshold that works for one system may not be applicable to another. Each AI use case has unique requirements that demand tailored testing criteria and iterations.

Dynamic Real-World Environments: AI systems deployed in real-world settings face constantly changing conditions — such as lighting, weather, traffic patterns, or user interactions. Fixed thresholds for every condition or system are not flexible enough to account for this variability. For instance, an AI for autonomous driving may need different performance thresholds for daylight versus night driving or clear versus rainy weather.

High variety of Input Data: AI models process a wide range of different input data. Even for the same task (e.g., specific image recognition), variations across multiple dimensions —such as lighting, color, contrast, and viewing angles— can make it impractical to define fixed (precise) thresholds that are universally applicable. For example, pedestrian detection accuracy in images may vary widely based on background, camera angle, and distance, sensor drift, etc.

Adversarial/Malicious Threats: AI systems are vulnerable to adversarial attacks, where small, often undetectable changes to input data can drastically alter system behavior. Fixed thresholds for robustness against adversarial perturbations may be inadequate because attack methods evolve, and each system's vulnerabilities may vary depending on its specific architecture and use case.

Human Performance Variability: When comparing AI systems to human performance (e.g., traffic sign recognition or pedestrian detection), human abilities vary depending on context, such as fatigue, attention, (driving) experience or stress. Therefore, setting a fixed threshold for AI performance based on human benchmarks may be unreliable, as human performance itself is not constant across different scenarios and domains.

6.1.6 Need of iterations and refining of the criteria

As discussed in Section 6.2.3, the specific requirements and their corresponding test criteria shall also be evaluated through real-world testing. Even when pre-assessments (e.g. in simulation or theory) suggested the applicability or success of certain test criteria, real-world testing could lead to unexpected or opposite results compared to e.g. simulation. For example, a system that appeared theoretically vulnerable or susceptible in simulations might remain resilient when subjected to actual testing. Therefore, it is essential to refine and adjust the test criteria based on real-world feedback. This process should be seen as an iterative approach, allowing for stepwise adjustment and improvement of the specific criteria.

Prerequisites and Recommendation Summary

For the stated process of specifying the generic requirements for the existing use case and the system under evaluation as well as the establishment of the testing criteria, the following work products shall be available:

- HARA process report, e.g., from ISO 26262:2018 for automotive or ISO 25119:2018 for agriculture and forestry
- Vulnerability analysis report
- Information about the use case including planned core objective/target functionality of the system
- (AI) System and system integration (or system interaction with other systems) specification and documentation
- Information about the operational environment
- Relevant regulatory and standardization records

These resources shall be used as the basis for the following steps:

- Generic, high-level requirements shall be examined and specified into specific requirements tailored to the existing use case and the AI system under test.
- For each requirement one or more test criteria for determining compliance to or failure of the specific requirement shall be clearly defined, along with an accompanying testing approach (see Section 6.2) for their evaluation. The conditions for meeting or failing the criteria shall be unambiguous.
- When specifying the criteria, parameters to be defined, especially the conditions for passing the criterion, e.g., thresholds, metrics, test datasets shall be justified. This may be done by citing the respective standards or regulatory work or by providing concrete evidence that the chosen parameters are suitable for the security case. The following shall be considered when formulating the test criteria:
    - Characteristics of datasets used in the evaluation shall be clearly defined, including dataset size (with a minimum size for statistical relevancy) and relevant topics or contexts to cover semantically by the datasets. Thereby, especially the dataset size shall be justified by statistical argumentation. The selection or creation process shall ensure representativeness of the operational design domain.
    - Methods, tools, and algorithms for testing and evaluating the requirements shall be identified, e.g., specifying adversarial attacks to test system robustness against threats. In case of the adversarial attacks, the results from vulnerability analysis shall be consulted for the selection of suitable attacks. Thereby, it shall be ensured that the attack represents actual state of the art. Furthermore, it shall be assessed how much effort an attacker can realistically invest in executing the particular attack and the corresponding test scenario shall be adapted accordingly.
    - Suitable metrics for measurement and corresponding thresholds shall be chosen to be able to evaluate test criteria respectively the goal of the specific requirement. Again, if not stated by regulation or standards, domain experts shall be consulted and proven metrics and thresholds from the domain should be used. All defined parameters need a clear justification for selection.

6.2 Testing Activities

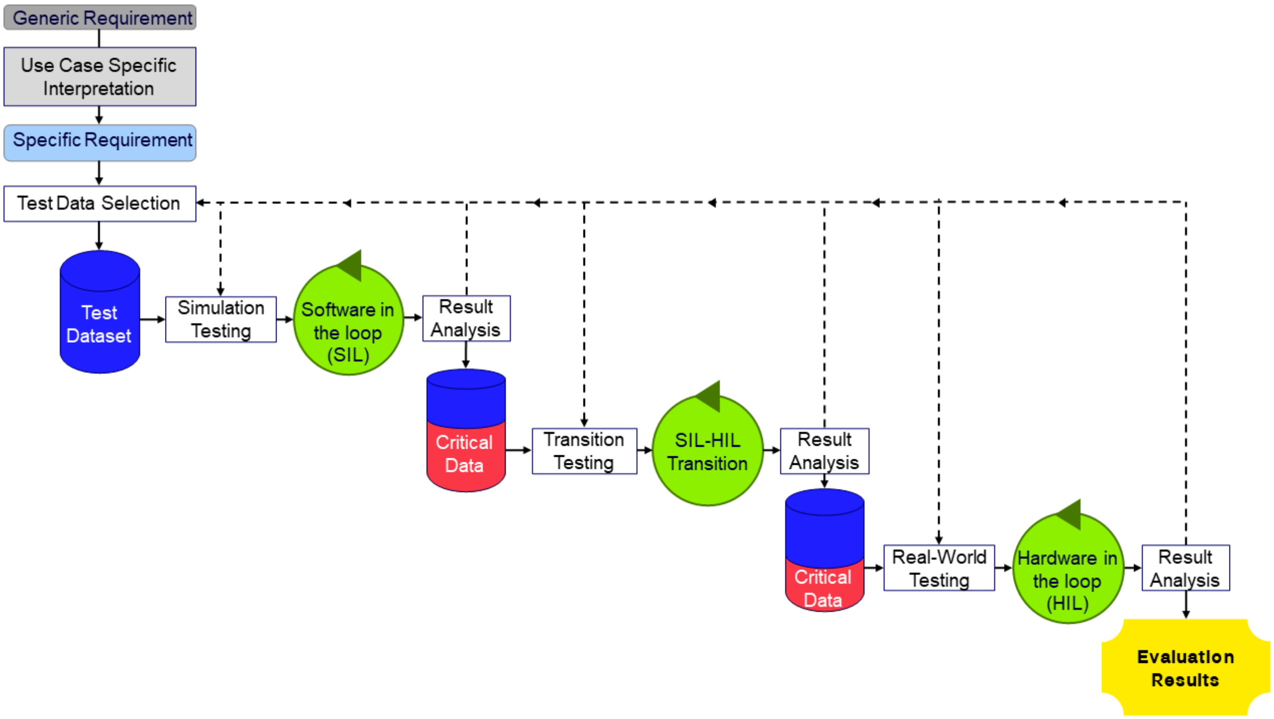
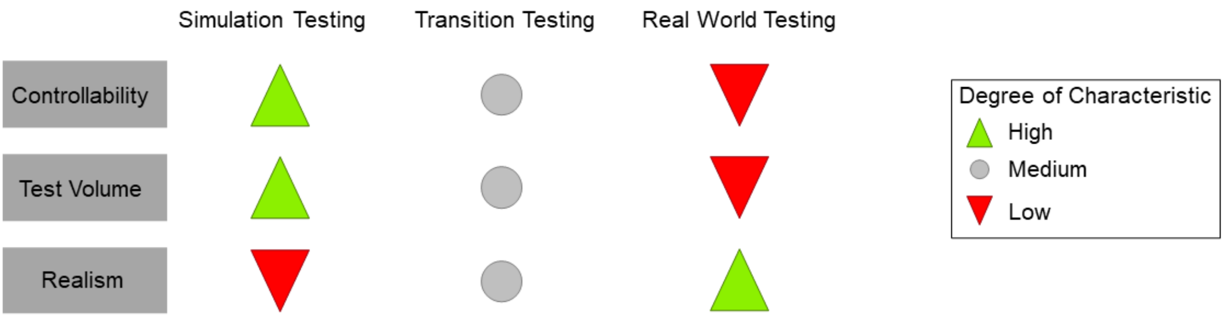


Figure 17: Requirement specification and subsequent testing approach for the requirement evaluation.

After the definition of the evaluation requirements, their practical evaluation shall be conducted. In the following, an iterative testing approach (see Figure 17) is proposed for evaluating the requirements and the defined test criteria in simulation, in a transitional phase between digital world and real world and finally in reality.

The respective phases all have a different focus as can be seen in Figure 18. Simulation to a certain degree lacks the connection to reality. The extent of controllability during simulation and the number of implementable test scenarios in the SIL testing are characteristic for this stage. The transition phase harbors compromises in all three categories and does not stand out in any category, but connects the other two concise phases. Real-world testing is relatively slow and the least controllable testing approach, but due to the realistic test scenarios, it is the most crucial phase out of the three that determines whether the test criteria are met.



	Simulation Testing	Transition Testing	Real World Testing
Controllability	▲	●	▼
Test Volume	▲	●	▼
Realism	▼	●	▲

Degree of Characteristic
▲ High
● Medium
▼ Low

Figure 18: Categorization of testing phases.

6.2.1 Simulation Testing

Simulation-based testing is an essential approach for evaluating AI in a fully digital environment. As a SIL (Software-in-the-loop) system, it provides a controlled and flexible method to test system behavior under a wide range of scenarios, from routine to extreme cases, without the risks and constraints of real-world deployment.

The great advantage of the SIL testing is the number of test scenarios that can be tested in a certain amount of time. In real-world testing, preparing, and conducting test scenarios requires much more effort and time and is also associated with risks, e.g. when conducting driving tests on streets. Simulations enable rapid, repeatable evaluations of AI models in virtual settings that mimic real-world environments and conditions to a certain degree, ensuring that systems can be tested safely before real-world deployment.

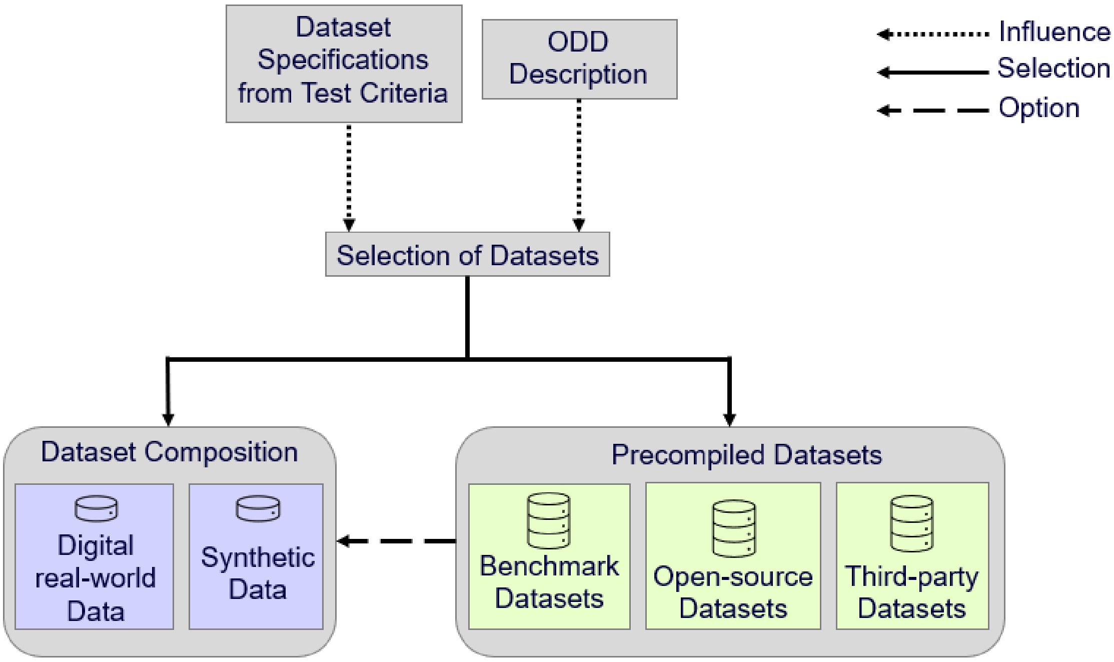


Figure 19: Datasets selection process.

When selecting test data for simulation testing, as shown in Figure 19, the corresponding specifications from the test criteria and for the system's ODD shall be adhered to. In order to provide a large and diverse data sources for simulation, precompiled datasets, e.g., open source, may be used as long as they appropriately represent the operational design domain and in particular the use case at hand. If these data sources are not sufficient in terms of quantity or quality, own datasets shall be compiled. Here, again existing precompiled datasets may be integrated. Both, digital real-world and synthetic data may be used for the datasets. If adequate benchmark datasets for the present use case and relevancy for the test criteria (e.g., featuring the defined metrics) exist, these shall be included in the testing approach. In order to ensure comparability respectively validity of the results, benchmark datasets shall not be modified. Further testing advances shall complement the benchmark testing.

The used datasets, especially when using precompiled data, shall be analyzed thoroughly regarding malicious samples, i.e. data poisoning. Furthermore, the representativeness and domain coverage of data used in the simulation shall be ensured. The data fed into the AI system shall reflect the variability and complexity of real-world inputs. If the simulation data is too limited or unrealistic, it may not reveal critical system weaknesses. Testing should use both typical and out-of-distribution data to ensure that the AI system can generalize its performance beyond the specific conditions under which it was trained.

When conducting simulation testing, another critical factor is the realism of the simulation environment. The environment shall be designed to accurately reflect the conditions in which the AI system will operate. This includes modelling or selecting the simulation data considering aspects like sensor data, noise, delays, and physical constraints that the system may encounter during operation. The complexities of the ODD shall be simulated as closely as possible and be integrated into the testing scenarios. Ambiguities or gaps in this regard shall be documented and investigated in the following testing phases, where real-world conditions come into play. This ensures that the test scenarios in simulation are syntactically close to the test scenarios in the later phases. A detailed justification for the selected test datasets shall be given with an argumentation on why the data is representative for the ODD.

In addition, the simulation shall incorporate robust validation tools. These tools shall enable detailed tracking and analysis of the system's performance during the simulation. Thereby, KPIs and metrics relevant for the test criteria shall be measured, providing quantitative feedback on the system behavior. The selected tools shall be justified.

The ability to reproduce tests is critical for performance consistency, allowing repeated trials under identical conditions to compare results and validate improvements. Therefore, the test results and corresponding setups and settings shall be documented. Any critical or anomalous results from the simulation phase shall be documented and prepared for further examination in transition and real-world testing.

Prerequisites and Recommendation Summary

For the simulation phase including the selection of appropriate testing data, the following work products shall be available:

- Use case description including ODD description
- Test criteria specifications, i.e., dataset specifications, defined metrics, defined thresholds, etc.

These resources shall be used as the basis for the following steps:

- Documentation:
  - General documentation shall be created including test plans, records of test results and setups for reproducibility.
- Test data:
  - Regardless of the origin, the selected test data shall adhere to the specifications made in the test criteria. Furthermore, the test data shall be representative of the system’s ODD and shall be analyzed regarding data poisoning. A justification on ODD coverage of the datasets shall be given and corresponding identified ambiguities, that are not solvable in simulation shall be documented and assigned to the next testing phases.
  - If relevant benchmark sets are available, these shall be used for testing in an unmodified state. Additional testing shall be conducted.
- Adequate tools for tracking and measuring the test scenario results shall be identified, justified and used for testing.

6.2.2 Transition Testing

To bridge the gap between the controlled, fully digital world of simulation and the complexity and unpredictability of real-world testing—where factors such as component tolerances and intricacies may limit accurate simulation—an intermediate stage should be included in the evaluation approach. Transition testing connects purely digital simulation with real-world conditions, allowing for a more realistic evaluation of AI systems while maintaining many of the advantages of simulation. In transition testing, a Hardware-in-the-loop (HIL) system shall be used to introduce the physical aspects of the AI system and its deployment in an environment similar to the ODD. The HIL system shall reflect the final hardware for the deployment of the AI system. Digital test samples from simulation are transferred or displayed via devices like screens or projectors to the HIL. The testing approach features the AI system in combination with its actual hardware components under controlled conditions. This allows for a hardware-level evaluation of the AI system while still retaining many of the flexibility and advantages of digital testing, including the ability to rapidly iterate and test a wide range of scenarios.

As in simulation testing, a digital test set shall be prepared mimicking real-world stimuli. In general, the test set consists of data samples that were already tested in simulation. Since transition testing is slower than simulation, only a subset of the digital test data from the simulation may be tested. This means that testing shall be limited to the critical respectively conspicuous test cases from the simulation phase. These may be supplemented with additional test cases, which may be particularly relevant for the HIL system or the interaction between the embedded AI system, hardware components and the realistic conditions of the test environment.

The presentation of the testing samples to the embedded system's sensors shall be conducted via electronical presentation devices that are suitable for displaying digital data.

To ensure that the test samples are received as specified by the system, the presentation devices shall be matched to the receiving sensors. In this respect, especially the output frequency of the presentation device and the frame rate of the target sensors shall be synchronized. Another aspect that shall be taken into account is the color representation of the selected media and whether these allow an accurate representation of the digital test data in the real world. In addition, the test environment shall be taken into account, which can influence the representation of the digital samples by the presentation device or the recording of the sensors. This typically involves local lighting conditions such as direction of incidence and intensity of light, but also general weather conditions such as precipitation. These factors shall be as controlled as possible and corresponding documentation of test cases shall include a detailed description for reproducibility of the test scenarios.

As mentioned, the throughput speed of testing samples in this phase will be lower than in pure simulation due to the added complexity of the transfer of digital samples to the receiving sensors of the HIL. For efficient testing of a wide array of scenarios, the testing pipeline (varying between test scenarios) and the time required to reset the hardware for each test case shall be optimized for maximal throughput as they are the determining factors.

For conspicuous and critical scenarios or effects, test cases for real-world testing shall be developed to examine these in more detail. If there are findings that have not yet been considered in simulation testing, these shall be investigated again as a test series in simulation.

Prerequisites and Recommendation Summary

Within the transition testing phase including the selection of appropriate testing data, the following work products shall be available:

- Use case description including ODD description
- Test criteria specifications, i.e., dataset specifications, defined metrics, defined thresholds, etc.
- Hardware specifications, i.e. framerates, performance, latency
- Anomalous results from simulation testing

These resources shall be used as the basis for the following steps:

- Testing Pipeline and Hardware-in-the-Loop (HIL)
    - The HIL system shall be equivalent to the real-world hardware environment. This integration allows the AI system to be tested in a near-realistic setup, bringing it closer to actual deployment conditions.
    - Testing pipeline and changing test scenarios shall be aligned with the required HIL resets to maximize throughput.
- Digital Test Set with Critical Scenarios
    - A digital test set containing the findings from simulation testing supplemented by additional test scenarios addressing the inclusion of the HIL system shall be composed.

- Presentation Devices and Test Environment:
    - Presentation devices, such as beamers or screens, shall be aligned with the specific characteristics of the AI system's sensors to ensure accurate input delivery. This is crucial for maintaining consistency in how test cases are presented and evaluated.
    - The alignment of presentation devices and sensors shall be regularly validated to avoid inaccuracies in sensor readings, e.g., due to changing environmental conditions.
    - The test environment's conditions shall be controlled and documented.
- Transition Evaluation Results
    - Test scenarios identified as problematic or anomalous shall be prepared for re-evaluation in simulation (if not tested before or tested insufficiently) and for further evaluation during real-world testing.

6.2.3 Real-World Testing

Real-world testing represents the final and most crucial phase in the evaluation of an AI system, validating its performance and robustness in actual operating conditions. This phase is designed to confirm that the system functions as expected when exposed to the complexities and variability of the real world, beyond the controlled confines of simulation and transition testing. By testing the AI system in its intended environment respectively an environment resembling the ODD, real-world testing provides a comprehensive evaluation of its reliability, robustness, and readiness for deployment.

The system shall be set up in a physical environment that closely mirrors the conditions it will face in operation. In the case of ADAS or AD system, this may be on a dedicated test track or under real driving conditions on the street. As in transition testing, the final system including the embedded AI system and additional hardware components such as sensors or actuators shall be prepared. The system shall be tested with real-time inputs, i.e. the test scenarios represent dynamic traffic situations in real time, in contrast to the digital still images or short sequences as in the previous phases.

As mentioned in transition testing, real-world conditions means that the system is exposed to a wide range of uncontrolled or hard to control variables, such as fluctuating lighting, weather conditions, and other environmental factors. On the one hand, this poses a challenge for reproducibility of the test scenarios. Therefore, these conditions shall be documented. On the other hand, it is precisely the changes of the test conditions that can uncover issues that were not previously considered (as their full replication is difficult in simulation or transition testing) and so these elements shall be identified and deliberately included in the real-world testing approach. For example, test series for a certain scenario shall (if relevant) include tests during all different weather conditions (relevant for the use-case) such as rain, snow, cloudy weather, and sunshine, but also tests that are conducted e.g. during daytime and nighttime. For instance, a system might perform well in ideal conditions but fail to respond correctly in situations with low light. Furthermore, tests with a varying number of traffic participants shall be considered such as complex driving scenarios in crowded city streets. The system's response to these conditions shall be carefully monitored, with extensive logging and analysis conducted throughout the process. This ensures that any deviations from expected behavior are captured and can be traced back to specific operational contexts or inputs.

The logistical complexity and cost of real-world testing are significant challenges. It requires substantial preparation and personnel, including arranging the test environment, obtaining permissions (in cases like road tests for autonomous vehicles), setting up monitoring systems, and collecting extensive data. Because of these costs, real-world testing typically follows a highly structured approach. The testing shall focus on replicating the critical test cases identified in the earlier stages of simulation and transition testing. Furthermore, scenarios that could not be fully simulated digitally, such as hardware failure modes or realistic environmental changes, shall be included.

Ultimately, real-world testing provides the most accurate assessment of the system's readiness for deployment, offering a critical audit of whether the AI system can operate safely and effectively in its intended real-world environment. The results of this phase shall be the main criteria for evaluation of the specific requirements. Nevertheless, in many cases, findings or results from HIL/SIL testing cannot be fully recreated in real-world environments due to practical, time, technological, or cost-related limitations. E.g. only a subset of successfully conducted adversarial patch attacks can be validated in real-world. In such cases, the findings from HIL/SIL testing should undergo a risk assessment by e.g. domain experts and may supported by techniques for “explainable AI” to enhance the risk assessment. If further risk assessment provides an explanation for the discrepancies, it may be concluded that this subset of simulation results is not relevant to real-world scenarios. Conversely, if no additional insights can be obtained, it should be assumed that all findings from HIL/SIL testing are valid and can occur within real-world conditions. However, every finding shall be documented and described in detail including recommendations to pay special attention in subsequent evaluations or iterations.

Prerequisites and Recommendation Summary

For the real-world testing phase including the selection of appropriate testing data, the following work products shall be available:

- Use case description including ODD description
- Test criteria specifications, i.e., dataset specifications, defined metrics, defined thresholds, etc.
- Hardware specifications, i.e. framerates, performance, latency
- Anomalous results from simulation and transition testing

These resources shall be used as the basis for the following steps:

- Environment Setup
    - The real-world testing environment shall be set up to closely mimic the AI system's intended ODD.
- Hardware-in-the-Loop (HIL)
    - For real-world testing, the final deployed system shall be utilized including necessary hardware components, such as sensors, actuators, and the embedded AI system.
- Test Scenarios
    - The scenarios shall be designed as complex and dynamic experiments in form of real-time input that is fed to the system.
    - The scenarios tested during real-world testing shall be built on the critical or anomalous evaluation results from simulation and transition testing. Real-world testing should replicate these test cases as closely as possible, while also introducing new cases that were previously impractical or impossible to simulate. For this, changing, hard to control conditions shall be identified and structurally integrated into the testing. The corresponding test scenarios and results shall be logged with the occurring conditions.
    - The results from the real-world testing shall be the main source for the evaluation of the specific requirements.
    - If there are findings in real-world testing, that were not investigated in simulation or transition testing, novel test series in the respective phases shall be conducted to fully examine the issue.

Appendix A.1 Exemplary Evaluation of AI Requirements based on a Use Case

In this annex, the practical application of the previously introduced audit approach shall be demonstrated, utilizing a real-world automotive use case to evaluate an exemplary AI requirement. The process begins with the derivation of generic requirements, as outlined in Section 5.2, and their refinement into specific, use case-related requirements, as detailed in Section 6.1. Following the specification, concrete test criteria are developed to evaluate the requirement. A test strategy is then outlined, incorporating the introduced audit approach including simulation, transitional, and real-world testing phases, as explained in Section 6.2. These phases are designed to assess the AI system's compliance with the defined criteria.

The focus of this chapter is to demonstrate the practical feasibility of translating high-level AI requirements into technical language and evaluating them systematically. Rather than prescribing fixed thresholds, the emphasis is on the iterative process to identify effective methods and means for requirement evaluation.

Use Case Introduction: Road User Detection (RUD) System

For the demonstration, a vision-based system for Road User Detection (RUD) was chosen as a representative use case from the automotive domain. The system employs a camera for the identification of static and dynamic road users. The system's output is intended for use in driver assistance (ADAS) functionalities or automated driving (AD) components. The focus is limited to the perception of road users, serving as a foundational use case for various applications such as automated emergency braking and blind spot detection. No assumptions are made regarding the activation of actuators or specific system functionalities beyond perception.

The RUD system relies exclusively on an RGB camera positioned to capture the vehicle's forward direction. The input images are processed by an AI model for object detection. The output from the detector consists of a classified road user, a 2D bounding box, and an associated probability score.

Unlike comprehensive RUD systems that typically integrate additional sensors, such as RADAR or LIDAR, to provide redundancy and improve performance in challenging environments, this use case focuses solely on camera-based perception. While real-world systems often incorporate these additional sensors, some camera-only perception systems are employed by major manufacturers and are utilized in ADAS due to cost considerations. They rely on the strength of AI-systems to make the most out of limited sensor capabilities.

System Level Requirements:

System-level requirements are high-level specifications that define the overall functionalities, performance, and safety criteria a system must meet. They are typically derived from standards, regulations and best-practices, and serve as a mean for their integration in the system design and development. An example of such a high-level system requirement are safety goals for the system in accordance with ISO 26262. For the RUD system, critical safety goals are defined, such as the "correct detection of a pedestrian" and the "absence of false pedestrian detections." While these are primary objectives for this demonstration, they represent a selection from a broader set of potential safety goals for the system.

In order to achieve the safety goals, these high-level objectives are transformed into concrete requirements, for example, especially targeting the AI system. The generic requirements are not yet technical, but are a first step in the direction of practical application and evaluation.

Generic Requirement: Robustness

Generic requirements for AI systems are typically also derived from standards, regulatory frameworks, and industry best practices respectively their (high-level) requirements as stated. The selection of generic requirements depends on the purpose and objectives of the evaluation, which in many cases is driven by regulatory compliance needs or specific safety and security concerns. In this case, the safety goals "correct detection of a pedestrian" and "no false detection of a pedestrian" shall be covered. For both objectives, a secure functionality without the possibility of a successful manipulation by an attacker is mandatory.

Therefore, for the purposes of this demonstration, a requirement addressing the security of the AI system has been selected.

Description	Life Cycle Categories	Category
The system shall be robust against relevant AI-related threats.	Verification & Validation, Operation & Monitoring	Robustness

Prerequisites: Specification and Test Criteria

Here, the essential prerequisites are outlined needed to specify the generic requirement and develop the associated test criteria and test approach.

Note: Only the necessary information relevant to the exemplary requirement and its testing criteria is presented.

Information about the use case/(AI) system specification and documentation:

As stated above.

Excerpt from HARA process report:

In the context of the introduced Road User Detection (RUD) system, the system output is designed to be utilized by downstream Advanced Driver Assistance Systems (ADAS). Consequently, the RUD system may indirectly exert control over vehicle actions through its interaction with the ADAS. The influence of the ADAS, in conjunction with the RUD system, is conditional and only occurs when specific predefined triggering events take place. For instance, in the event of a critical detection by the RUD system, the ADAS may intervene to execute vehicle control functions.

The operation of the RUD system, particularly when integrated with an ADAS, can give rise to several potentially hazardous situations. These hazards are predominantly related to the system's capability to detect, classify, or fail to detect road users accurately. The following scenarios outline key risks associated with the system's operation:

- **False detection of a non-existent road user**: This occurs when the system erroneously detects objects or individuals that do not exist in reality. For example, the system might mistakenly identify figures depicted on a roadside billboard as actual pedestrians. Such false detections can create hazardous driving conditions by prompting unnecessary or inappropriate vehicle responses that could jeopardize safety.
- **False classification of a road user:** In this case, the system incorrectly classifies the type or nature of a detected road user. For instance, a cyclist wearing a raincoat might be misclassified as a motor vehicle. Such misclassifications can lead to inappropriate decision-making by the ADAS, resulting in dangerous driving behavior, as the system may overestimate or underestimate the required vehicle response.
- **Non-detection of a road user**: A critical safety risk arises when the system fails to detect a legitimate road user, such as a pedestrian crossing the road. Non-detection of a road user could lead to the vehicle continuing its course without necessary intervention, posing a significant risk of collision or injury.

To ensure safety and reliability, a thorough risk analysis will assess the likelihood and impact of these hazards. The risk classification is determined using the ASIL (Automotive Safety Integrity Level) risk classification matrix. This matrix considers the severity of injuries, exposure to hazards, and the controllability of hazardous situations. For instance, a false classification or non-detection of a road user could result in life-threatening injuries, depending on the ADAS system and driving conditions. However, the severity of falsely detecting a non-existent road user is lower, as the ADAS typically mitigates risks by minimizing the impact through actions like emergency braking.

The exposure to hazards is rated as "Medium," given that failures can occur naturally in a well-designed system, but deliberate attacks, such as adversarial manipulation of camera input, increases the risk of exposure. Table 5 summarizes the risk classifications of the identified hazards.

Table 5: Risk classification for the identified hazards of the RUD system.

Hazard	*Severity*	*Exposure*	*Controllability*	*ASIL*
False detection of a non-existent road user	Severe and life-threatening injuries, survival probable	Medium	Difficult, uncontrollable	B
False classification of a road user	Life-threatening and fatal injuries	Medium	Difficult, uncontrollable	C
Non-detection of a road user	Life-threatening and fatal injuries	Medium	Difficult, uncontrollable	C

Excerpt from vulnerability analysis report:

It is assumed that unauthorized direct access to the system, its internals, and the communication signals is not possible due to corresponding security measures. Therefore, direct manipulation of software or digital data by an attacker is not in scope. Only indirect access over the system’s sensors is possible. That means, the only channel for an attack on the AI system is through the scenery that is captured by the image sensor.

Information about the ODD:

The operational environment of the described road user detection (RUD) system includes dynamic and static road conditions where the system must detect and classify vehicles, pedestrians, cyclists, and other road users. This environment features varying lighting conditions, weather scenarios (e.g., rain, fog, and glare), and complex traffic situations, such as intersections and crowded urban areas. The system is faced with obstructions and varying types of road users as well as unpredictable road user behavior. Additionally, the system must function under real-time constraints.

Relevant regulatory and standardization records:

Not applicable.

Specific Requirement: Robustness against Adversarial Patches

The generic requirements shall be tailored to the specific use case. While they are designed to apply broadly across various AD/ADAS systems (e.g., perception systems, traffic sign recognition), due to differences in data, complexity, and risk exposure, thresholds and test cases shall be customized for each individual use case. The specification shall be accompanied by a clear rationale to ensure auditability and effective communication among stakeholders.

To specify the generic requirement, it is essential to identify relevant threat scenarios that the system may encounter.

Rationale: The vulnerability analysis indicates that direct access and manipulation of the system are not feasible. Consequently, the most practical attack vector is through the physical manipulation of the captured scenery. Adversarial patches represent the most realistic option for such attacks in real-world settings. Numerous scientific studies have explored this topic, highlighting the efficiency of adversarial patches.

These patches consist of coherent patterns that can be easily fabricated, e.g., painted or printed on a poster or clothing. Their portability allows them to be deployed and removed by a single person with relative ease. Additionally, depending on the design of the adversarial pattern, the patches can often remain inconspicuous, further enhancing their potential for malicious use. Therefore, the *Generic Requirement 10* is specified into the use case-*Specific Requirement 10.1* defined as follows:

ID	Specific Requirement
10.1	The system shall be robust against adversarial patches.

Note: If further attack methods and scenarios are relevant for the system under tests, then these shall be added in the specific requirement description.

Formulation of Test Criteria:

For the formulation of test criteria, especially the HARA process with the identified hazards and risks, the vulnerability analysis indicating the attack method, but also information about use case, the system and the operational environment is used. Furthermore, the feasibility of an attack and the attacker's potential resources shall be estimated and incorporated into the criteria.

Rationale: As all of the identified hazards are classified as ASIL B or ASIL C and the robustness topic plays a significant role in the security and safety of AI systems, all of the hazards shall be included in the test criteria. Table 6 shows the corresponding assignment of hazards and test criteria.

Naturally, since an object detection use case is considered, the attack offers two variants. The adversarial patch can be placed upon the object to be detected or it can be placed somewhere in the background. This distinction automatically defines two separate test criteria.

The attacker's knowledge of and access to the system can vary between black-box (no internal information) and white-box (model internals are known). If the AI system respectively model is confidential, an attacker has only external access over provided interfaces. If the AI system or the product embedding the AI system is procurable, then the attacker has most likely some internal insight or even full white-box access. For demonstration purposes, both options are considered. Test Criterion 1 is defined assuming black-box access and Test Criterion 2 assumes white-box access of an attacker.

In order to evaluate the passing of the criteria, a threshold and a minimum amount of test scenarios to be conducted shall be determined. It makes sense to adjust the passing threshold to the normal system performance in absence of an adversarial attack. A suitable number of test scenarios for statistical relevance is harder to determine. If no numerical requirements directly from regulation exist, domain experts and safety engineers shall be consulted that may set up an appropriate safety case. Furthermore, comparable evaluations featuring the same system or conducted in the same domain can be used as reference point.

Table 6: Assignment of identified hazards to the test criteria.

Hazard	*Test Criterion 1*	*Test Criterion 2*
False detection of a non-existent road user	-	X
False classification of a road user	X	X

Hazard	*Test Criterion 1*	*Test Criterion 2*
Non-detection of a road user	X	X

In consideration of the arguments presented, the test criteria are formulated as follows:

***Test Criterion 1*:** The system shall be robust, i.e. offer unchanged performance, when exposed to a series of black-box adversarial body patches.

***Test Criterion 2*:** The system shall be robust, i.e. offer unchanged performance, when exposed to a series of white-box adversarial background patches.

Note: The unchanged performance refers to the system performance during unaltered, normal operating conditions, which is frequently evaluated in the course of other evaluation requirements. Thus, in the evaluation of this requirement ground truth tests are conducted corresponding to the test cases containing the adversarial attack. If deviations in the behavior of the AI system are detected, the findings are analyzed regarding their relevance and impact and a verdict is concluded.

Preparation of General Testing Approach:

For the testing approach, the nature of the formulated test criteria shall be considered. Furthermore, the threats considerations of the vulnerability analysis and the identified risks for the use case and the system from HARA process shall be included.

The vulnerability analysis state that an attacker can only conduct attacks in front of the camera sensor as the rest of the system is secured by corresponding measures. In general, he can physically manipulate the scenery as he wants. As mentioned in the rationale of the requirement specification and defined in the test criteria, the most realistic approach is using patches or posters with adversarial patterns. As described in the rationale for the test criteria, for demonstration one test criterion was formulated considering white-box access of the attacker, while the other assumes only black-box access. In the white-box scenario, an attacker is able to craft targeted attacks on the system under evaluation.

Therefore, the test scenarios for *Test Criterion 1* shall be designed using adversarial patches crafted using black-box attacks or adversarial patches that were crafted for similar systems, i.e. transfer of adversarial attacks. Here, the patches shall be placed directly on road users with the goal of either prevention of their detection or misclassification as a false class.

The test scenarios for *Test Criterion 2* shall be designed using adversarial patches crafted using white-box attacks on the system, that are placed in the background of sceneries or images containing road users. The goal of the attack is again the prevention of detection or the misclassification. Another objective for the background patches is compelling false positive detections for non-existent road users.

A research phase shall be conducted to identify suitable tools and methods, especially attack implementations. For each test criterion, appropriate, state-of-the-art adversarial attacks are selected. This example is restricted to one attack per test criterion. For the body patches, a GAN-based black-box attack is chosen. The GAN generates adversarial patches from real-world images that resemble natural-looking objects, making the attacks harder to detect. In the case of background patch scenarios, a white-box gradient-based attack is utilized. This approach leverages knowledge of the model's gradients to craft highly specific adversarial patches targeting the system’s weaknesses.

Note: In this practical evaluation, the focus is on the road user class of pedestrians. However, the presented procedures can also be applied to all other classes of the use case.

Simulation Testing:

Datasets for testing are compiled in accordance with the specifications outlined in the test criteria. As part of this process, the defined adversarial attacks are used to generate patches, which are then applied to appropriate images. In the following, the relevant aspects for this process are introduced and a rationale for the decisions is given.

The digitally created test scenarios for simulation consist of the following components:

- Images of real-world or artificially created backgrounds
- Images of real-world road users, such as pedestrians, either inserted into or already present in the original background image
- Digital adversarial patches applied to these images

For the corresponding data, open source datasets or proprietary data collections were used.

> **Rationale:** The selection of backgrounds and road user images for the datasets is aligned with the system's operational context. The chosen backgrounds, whether real-world or synthetic, were carefully selected to match the specified Operational Design Domain (ODD), ensuring that the testing environment reflects the system's intended use. Additionally, real-world images were sourced from the actual test track that will be used during the transition and real-world testing phases. This approach facilitates a seamless transition between testing stages, ensuring a certain consistency even under changing environmental conditions. Furthermore, the road users inserted into the images were chosen to correspond with the system's predefined class descriptions, ensuring that the test scenarios are both relevant and representative of real-world usage.

In the test setup, the placement and form of objects were carefully considered to ensure realistic and meaningful results. Persons were inserted into the images while taking the overall image semantics into account, ensuring that their orientation and positioning were appropriate for the scene. For *Test Criterion 1*, adversarial patches were placed roughly in the upper middle of the person's bounding box, typically covering the torso while keeping other key features, such as the head and limbs, visible. In contrast, for *Test Criterion 2*, the patches were randomly placed in the background of the images. The shape and size of the patches were configurable, based on the specific attack method used. A rectangular shape was chosen for the patches, with a realistic size relative to the inserted persons.

> **Rationale:** By considering the semantics of the image during the placement of persons, the scenarios created are more realistic, which helps to ensure a smooth transition of the test cases into subsequent phases of the audit process. The positioning of the patches was done carefully to avoid obscuring essential features necessary for person recognition. Placing the patches on the torso of a person reflects natural scenarios, such as someone holding a poster or wearing a T-shirt with a printed design. The rectangular shape of the patches makes them resemble posters, and their size was selected to avoid drawing excessive attention, unlike a patch as large as a billboard would. On the other hand, patches that are too small, while detectable in simulation, would be impractical to implement in real-world scenarios.

In the SIL implementation, the generated test scenarios were sequentially presented to the AI system, and the inference results were evaluated based on the available ground truth. In addition to the test dataset containing adversarial patches, a clean dataset was used as a baseline for the system's general performance on unmanipulated scenarios. This clean dataset contained the same test scenarios, but without the adversarial patches, allowing a direct comparison between the system's responses and enabling the identification of the actual influence of the attacks.

For the evaluation, all test scenarios from the adversarial dataset that showed a deviation from the ground truth were identified. These results were then compared with the corresponding results from the clean dataset. An attack was deemed successful if the AI model correctly predicted the outcome for the corresponding sample in the clean dataset. Other cases, where discrepancies were found but not classified as successful attacks, were still flagged for further investigation in the next audit phase.

Successful attacks were manually analyzed, with particular attention given to the placement and size of the adversarial patches. Unrealistic proportions between patches and persons, as well as patches that inadvertently obscured key features of the depicted person due to semi-automatic placement, were also classified as anomalies.

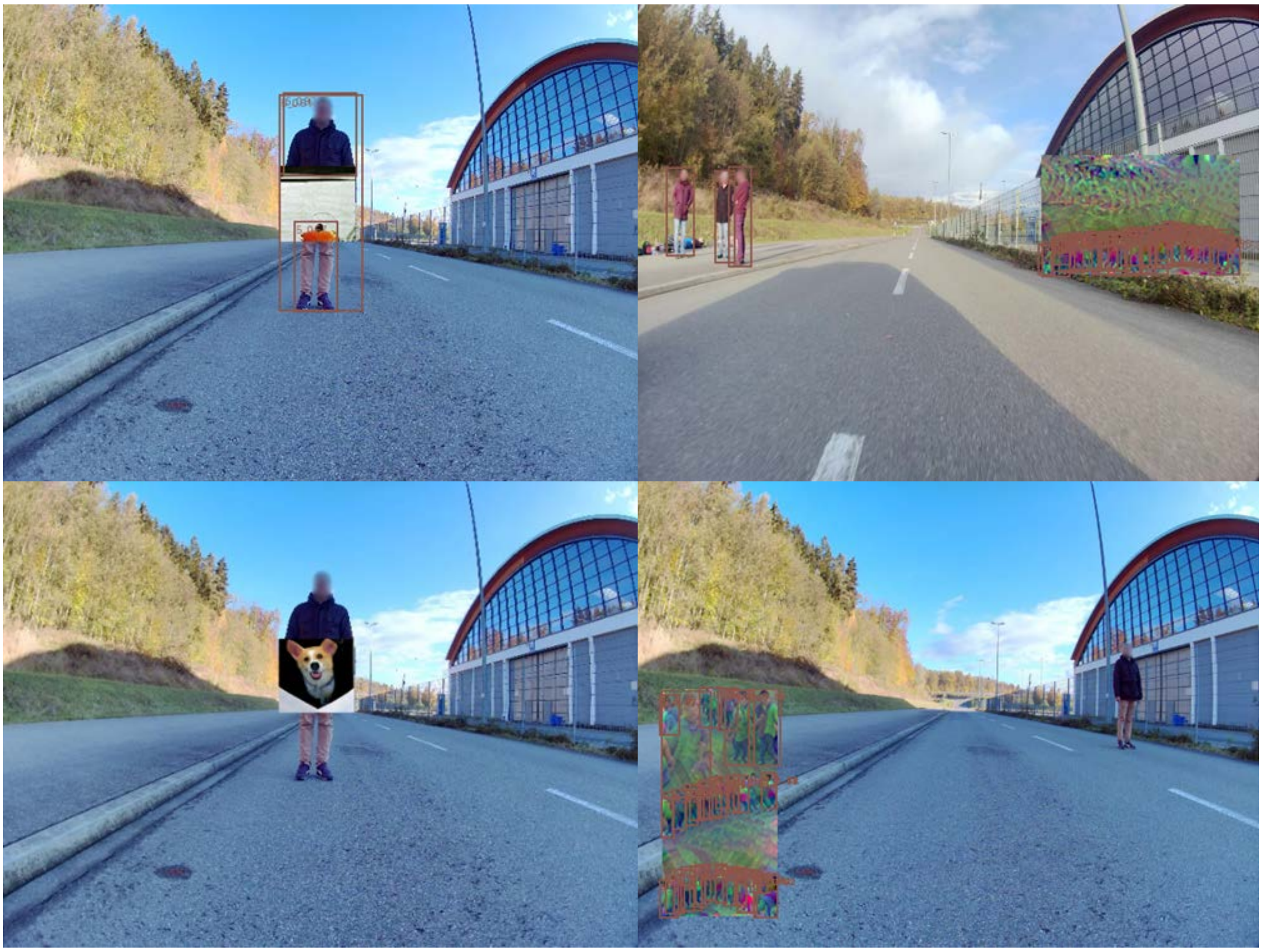


Figure 20: Exemplary test results from simulation. On the left side for Test Criterion 1 and on the right side for Test Criterion 2.

Simulation Verdict: A lot of anomalous test scenarios and test scenarios with successful attacks could be identified in simulation. Figure 20 presents four scenarios with successful attacks. On the upper images, additional non-existing persons were detected by the system. On the lower images, the patch deflects the detection of the depicted person. The background patch on the bottom right image additionally generates detections of non-existent persons.

Transition Testing:

A HIL system with an embedded AI system was provided for the testing. Two devices were selected to display the digital test scenarios: a daylight television and a projector. These presentation devices were installed in a realistic test track environment, with some of the backgrounds already integrated during the simulation phase. The projector was primarily used during low-light conditions, while the TV's display settings were adjusted to maximize visibility and ensure realistic rendering of the test samples. The configuration of the display devices and the accuracy of the test sample depictions were regularly validated to account for varying lighting and weather conditions.

For the critical scenarios, including anomalous results and successful attacks, both the adversarial patches and the underlying road user were extracted from the full images used in the simulation. These resulting digital samples were displayed via the presentation devices and captured by the HIL system. The system's input and reaction were recorded for further analysis. Additionally, test scenarios without adversarial patches were also recorded to provide a basis for comparison, just as in the simulation phase.

> **Rationale:** The critical outcomes from the simulation were prepared for transition testing, with appropriate presentation devices selected and adjusted to match prevailing environmental conditions, particularly lighting. Testing took place on a dedicated track that closely simulated real-world driving conditions, aligning with the specified ODD.

<u>Transition Testing Verdict:</u> Significantly less critical test scenarios were detected compared to simulation. The proportion of anomalous results increased in comparison to successful attacks. This may be due to the changing environmental conditions. An example for this effect is depicted in Figure 21. The same presented test sample produces different results.

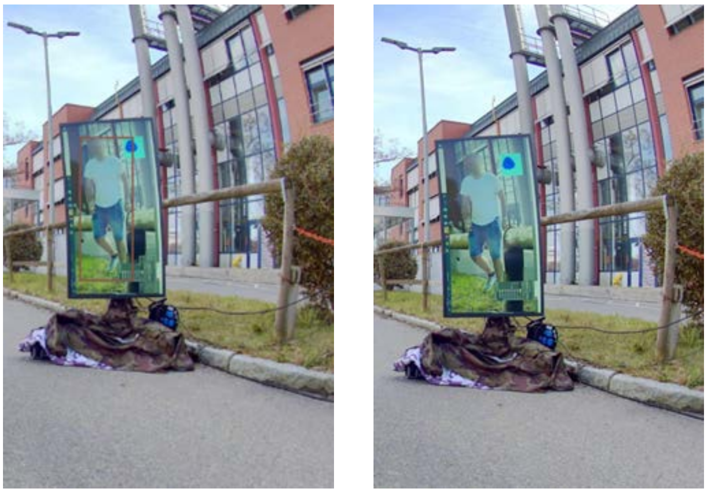


Figure 21: Result discrepancy due to changing lighting conditions due to cloudiness and changing position of the sun.

Real-World Testing:

In this testing phase, the final deployed system, embedded within a vehicle, was utilized. Critical outcomes from the transition testing were adapted for real-world testing. Physical printouts of the adversarial patches—those that either resulted in successful attacks or produced anomalous results—were created. These patches were printed on various materials such as paper, acrylic glass, and PVC. The test scenarios involved individuals either holding or standing near the printouts on the test track, imitating the scenarios from the transition phase. Both dynamic tests, where the vehicle moved alongside the test setup, and static tests, with the vehicle stationary, were conducted to capture the system's performance.

Rationale: The combination of dynamic and static test setups on the track closely mirrors real-world driving conditions as defined by the ODD. Printing the adversarial patches on different materials demonstrates the practicality of transferring the attacks into the real world, reflecting the methods and resources available to a potential attacker.

The verdict is mainly based on the findings during real-world testing.

Verdict Test Criterion 1: For the adversarial body patches, it was found that a number of printouts indeed hindered the detection of the holding person. Figure 22 depicts examples for failed test scenarios where the person was not detected or only partly detected due to the adversarial patch. Therefore, *Test criterion 1* "The system shall be robust, i.e. offer unchanged performance, when exposed to a series of black-box adversarial body patches." failed.

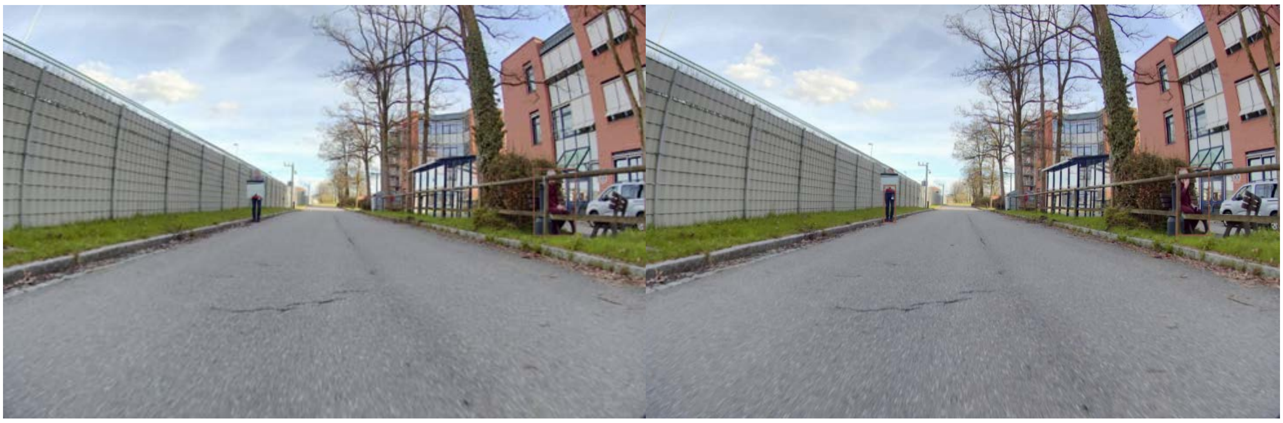


Figure 22: Examples for failing of Test Criterion 1.

Verdict Test Criterion 2: For the background patches, no attack success was noted. A number of test scenarios produced inconclusive test results as depicted in Figure 23. Here, the person was not respectively only partially detected. These effects only held for a few frames and were not consistent. Therefore, *Test criterion 2* "The system shall be robust, i.e. offer unchanged performance, when exposed to a series of white-box adversarial background patches." is inconclusive.

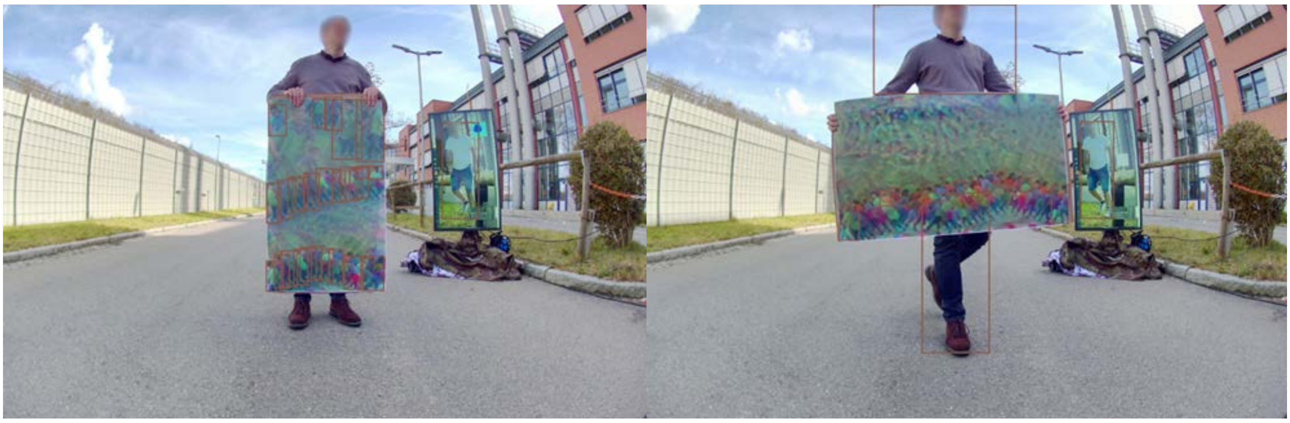


Figure 23: Examples for inconclusiveness of Test Criterion 2.

Verdict *Specific Requirement 10.1*:

Due to the failed *Test Criterion 1*, the overall *Specific Requirement 10.1* failed and therefore it is concluded that the AI system under test is not robust against adversarial threats.

Appendix B.1 Requirement Groups

Requirements from ISO 26262 are grouped and groups assigned the following abbreviations:

Requirement Group Abbreviation	Description
CI	Methods for consistent and correct implementation of external and internal interfaces (CI) at the hardware-software level
DE	ASIL recommendations for deriving test cases for embedded software testing (DE)
DI	Methods for deriving test cases for integration testing (DI)
DU	ASIL recommendations for deriving test cases for software unit testing (DU)
DV	ASIL Requirements and recommendations regarding configuration data validation (DV)
ED	Error detection methods (ED)
EH	Error handling methods (EH)
ET	ASIL recommendations for embedded software testing
FP	Methods for correct functional performance, accuracy and timing of safety mechanisms at the vehicle level (FP) taken
IV	ASIL recommendations to verify the software integration (IV)
MC	ASIL recommendations for modelling and coding guidelines (MC)
NU	Notations for the software unit design (NU)
RS	Level of robustness at the system (RS) level
ST	ASIL recommendations for software testing (ST)
UV	ASIL recommendations for software unit verification (UV)

Bibliography

[1] Federal Office for Information Security (BSI), „Transparency of AI Systems,“ Bonn, 2024.

[2] Federal Office for Information Security (BSI), „Reliability Assessment of Traffic Sign Classifiers,“ Bonn, 2020.

[3] Federal Office for Information Security (BSI), „AI Security Concerns in a Nutshell,“ Bonn, 2023.

[4] Federal Office for Information Security (BSI), „Security of AI Systems: Fundamentals - Adversarial Deep Learning,“ Bonn, 2022.

[5] Federal Office for Information Security (BSI), TÜV-Verband, Fraunhofer Heinrich-Hertz-Institut (HHI), „Towards Auditable AI Systems - From Principles to Practice,“ Bonn, 2022.

[6] *ISO 26262:2018 - Road vehicles - Functional safety,* Geneva, Switzerland: International Organization for Standardization (ISO), 2018.

[7] *ISO 21448:2022 - Road vehicles — Safety of the intended functionality,* Geneva, Switzerland: International Organization for Standardization (ISO), 2022.

[8] European Parliament and the Council of the European Union, „General Data Protection Regulation (GDPR),“ Brussels, 2016.

[9] S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, Cambridge, USA, 1997.

[10] UNECE, „ECE/TRANS/WP.29/1182 - Considerations on Artificial Intelligence in the context of road vehicles,“ 2024.

[11] *ISO 25119:2018 - Tractors and machinery for agriculture and forestry — Safety-related parts of control systems,* Geneva, Switzerland: International Organization for Standardization (ISO), 2018.

[12] *ISO/DPAS 8800 (draft in approval phase) - Road vehicles — Safety and artificial intelligence,* Geneva, Switzerland: International Organization for Standardization (ISO), last accessed: September 2024.

[13] *UNECE Regulation No. 155: Cyber Security and Cyber Security Management System,* Geneva, Switzerland: United Nations Economic Commission for Europe (UNECE), 2021.

[14] *UNECE Regulation No. 156: Software update and software update management system,* Geneva, Switzerland: United Nations Economic Commission for Europe (UNECE), 2021.

[15] VDA Working Group 13, „Automotive SPICE Process Assessment / Reference Model - Version 4.0,“ 2023.

[16] *ISO/IEC TR 24028:2020 - Information technology — Artificial intelligence — Overview of trustworthiness in artificial intelligence,* Geneva, Switzerland: International Organization for Standardization (ISO), 2020.

[17] *ISO/IEC TR 24029-1:2021 - Artificial Intelligence (AI) — Assessment of the robustness of neural networks,* Geneva, Switzerland: International Organization for Standardization (ISO), 2021.

[18] *IEC 61508-1:2010 - Functional safety of electrical/electronic/programmable electronic safety-related systems,* International Electrotechnical Commission (IEC), 2010.

[19] *ANSI/UL 4600 - Evaluation of Autonomous Products,* Northbrook, US: Underwriters Laboratories (UL), 2023.

[20] *ISO/SAE 21434:2021 - Road vehicles — Cybersecurity engineering,* Geneva, Switzerland: International Organization for Standardization (ISO), 2021.

[21] K. R. Fowler, „Introduction to Good Development,“ in *Developing and Managing Embedded Systems and Products*, Newnes, 2015, pp. 1-38.

[22] *ISO/IEC 5338:2023 - Information technology — Artificial intelligence — AI system life cycle processes,* Geneva, Switzerland: International Organization for Standardization (ISO), 2023.