Proc. of the European Safety and Reliability Conference (ESREL 2026)
Edited by José C. Matos, Paulo B. Lourenço, Michael Beer, Daniel Oliveira, Edoardo Patelli, Hélder S. Sousa, Rui A. Silva and Monica Santamaria

doi: 10.3850/ESREL2026061419_esrel26-p26368-cd

Integrating Artificial Intelligence in the Development Process of Functional Safety according to ISO 26262

M.Sc. Nour el HoudaFizzi

Department of Safety Engineering, University of Applied Sciences Ruhr West, Germany.
E-mail: nourelhouda.fizzi@yahoo.com

Prof. Dr.-Ing. David Schepers

Department of Safety Engineering, Smart Mechatronics GmbH, Germany.
E-mail: david.schepers@hs-ruhrwest.de

M.Eng. Benjamin Engels

Department of Functional Safety, University of Applied Sciences Ruhr West, Germany.
E-mail: Benjamin.Engels@smartmechatronics.de

Safety-critical system development remains one of the most demanding areas of engineering, especially in the automotive domain, where system failures can have severe consequences. The increasing complexity of electronic and software-based systems has intensified the need for structured, verifiable development processes to meet required safety levels. ISO 26262 provides such a framework for achieving functional safety throughout the entire automotive product lifecycle. In this context, the rapid progress of Artificial Intelligence (AI) introduces new opportunities but also challenges that existing safety standards have not yet thoroughly addressed. This contribution investigates the potential use of AI supporting tools within the development process of safety-relevant control functions. Based on a literature review and a comparative analysis of normative and regulatory frameworks, including ISO 26262, IEC 61508, ISO/PAS 8800, and the EU AI Act, potential AI tools are identified for supporting hazard and risk analysis (HARA), requirements engineering, definition of system and software architecture, and verification and validation activities. Based on practical examples, the paper explores how AI-based methods such as Natural Language Processing and Machine Learning could support information extraction, classification, and consistency checking while maintaining normative compliance and process transparency. The results indicate that AI can conceptually enhance efficiency and assist engineers in the safety development process, especially when handling large and complex data sets, such as in requirements engineering and documentation activities. Nevertheless, there still exist challenges regarding limited explainability, reproducibility, and the absence of clear qualification procedures. In conclusion, this paper contributes to a structured understanding of how AI can be aligned with existing safety frameworks, and it underlines that the integration of AI should follow a gradual and well-defined process, supported by clear verification methods, human supervision, and regular evaluation to maintain safety, normative compliance, and trust in safety-critical system development.

Keywords: Artificial Intelligence, Road Vehicles, ISO 26262, Functional Safety, V-Model Development Process.

1. Introduction

The functional safety standard ISO 26262(ISO 26262 2018) is an international standard that defines requirements for safety-relevant control functions in the automotive sector. The standard's main objectives are the control of random hardware failures and the avoidance and control of systematic faults in electric, electronic, and programmable electronic safety-relevant systems. For this purpose, the standard defines hardware and software development requirements based on the specific risks arising from a malfunction of the safety-relevant system. As human errors during the development process significantly influence safety, avoiding systematic faults is a major priority. This paper systematically examines how AI tools can be integrated into the ISO 26262 development process to enhance efficiency and improve fault-avoidance performance. It further identifies the restrictions and limitations of using AI in the safety development process.

2. Artificial Intelligence (AI) in the Context of Functional Safety

The application of AI methods has so far only been regulated to a limited extent by existing standards. The current second edition of ISO 26262 does not explicitly deal with the application of artificial intelligence and machine learning. However, in the third edition of ISO 26262 (expected to be published in 2027), it is planned to explicitly address the application of Artificial Intelligence. At present, however, it is unclear at what depth this will happen (Hoffmann 2025). ISO 21448 deals with the "safety of the intended functionality" of a control system. The standard mainly addresses "performance limitations or insufficient situational awareness, with or without reasonably foreseeable misuse" (ISO 21448 2022). ISO/PAS 8800 addresses the safe integration of AI into safety-critical vehicle functions (ISO/PAS 8800 2024) and can be considered a supplementary specification to ISO 26262 and ISO 21448. It defines requirements for model behavior, training data quality, verification and validation methods, and the safety argumentation for AI-based systems, but does not refer to supporting AI tools in the development process.

A fundamental source that systematically deals with the integration of AI into safety-critical systems is ISO/IEC TR 5469 (ISO/IEC TR 5469 2021). This technical guide explicitly distinguishes between the use of AI within safety-critical functions and its use in the development process. Another essential legal framework for AI use is the European Union's AI Act, adopted in 2024(EU AI Act 2024). It divides AI systems into categories based on the potential risk to safety and fundamental rights.

In non-safety systems, the use of AI is already widely established forsystem and software development. AI technologies are widely used to automate repetitive and complex tasks, reducing development time and improving quality and efficiency (Bringmann 2025; Baqar 2025). Large Language Models (LLMs) are increasingly being used to partially automate development processes through code wizards (Siebert and Jedlitschka 2024; Khade and Sambhe 2025). A systematic literature review by Wang and Chung (2022) (Wang and Chung 2022) revealed that AI technologies are already applied in industrial engineering processes for safety-relevant systems. The authors underline that developing such systems requires greater traceability and verifiability, whether AI is used in the product itself or as a supporting tool in the development process.

The following challenges should be considered when using AI tools in the development process:Black-box character of the tools (Hassija 2023), dependence of data quality (Nivedhaa 2024), hallucinations of LLMs(Siebert 2025), lack of ISO 26262 conformity (ISO 26262 2018), and integration of the tools into existing processes (Braun et al. 2025).

3. Possibilities for Integrating AI Tools in the ISO 26262 Development Process

Possibilities for integrating AI tools were systematically analyzed for each stage of the V-model development process of ISO 26262 (Fig. 1). The focus was put on the software development process, as the use of AI in software development is already well established and well-documented (Siebert and Jedlitschka 2024; Khade and Sambhe 2025). Some steps of the V-model for software development were merged into a single step, indicated by the color scheme in Fig. 1.

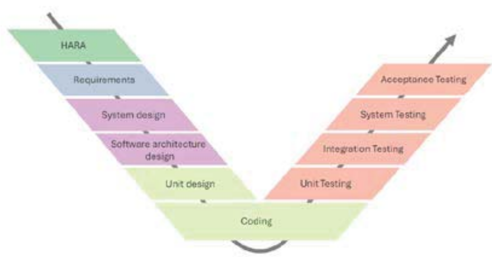

Fig. 1. Focus on the different stages of the V-model.

3.1. *Hazard and risk analysis (HARA)*

The HARA shall identify possible dangers in the relevant modes of operation. AI tools may support this process.

3.1.1.*Situation analysis and hazard identification*

LLMs may analyze technical documents, use cases, and former HARA reports to automatically identify dangerous situations and categorize them semantically (Hillen et al. 2024). AI-based tools like SAHARA (Simulation Aided Hazard Analysis Risk Assessment) model diverse scenarios (e.g., road and atmospheric conditions) to detect and describe potential critical system behavior(Barreto and Bachir 2020).

3.1.2.*Classification of the dangerous event*

ISO 26262 classifies risks based on parameters: severity, exposure, and controllability. Systems such as SAHARA use sensor data and historical accident reports to quantitatively assess these parameters (Barreto and Bachir 2020;Diemert and Weber). Additionally, ML algorithms may analyze former incidents to determine the required ASIL (Hillen et al. 2024). LLMs may cluster similar risks and prioritize them by risk level.

3.1.3.*Derivation and definition of safety goals*

LLMscan be used to derive safety goals from the ASIL classification. The proposed safety goals can automatically be checked for compliance with ISO 26262 (Hillen et al. 2024).

3.1.4.*Evaluation of applicability and conformity*

The use of LLMs can offer opportunities for a more efficient and objective HARA. With no specific regulatory framework for the use of LLMs in functional safety, their use should always be combined with human validation to ensure compliance with safety standards, as LLMs, for example, may hallucinate. But there is no fundamental reason not to use LLMs under human supervision to support HARA, as the method is iterative and AI can accelerate and structure the process(Barreto and Bachir 2020; Hillen et al. 2024;Diemert and Weber; Nouri 2024).

3.2. *Requirements management*

The key criteria of requirements management according to ISO 26262 are completeness, traceability, consistency, testability, and changeability of requirements throughout the entire safety lifecycle.

3.2.1.D*efinition of functional requirements*

NLP-based models can extract safety-relevant information from structured and unstructured text sources. This helps to identify functional requirements and transfer them into the Functional Safety Concept (FSC)(Zhao et al. 2020). LLMs may analyze requirements defined in natural language and derive technical models to support the specification of the Technical Safety Concept (TSC) (Bozyigit et al. 2024).

3.2.2.*Structuring and classification of requirements*

AI models may help to allocate requirements to pre-defined categories. For this purpose, pre-categorized data can be used to train a model in a supervised manner(Talele and Phalnikar 2021). Unsupervised learning can be used to classify requirements with semantic similarities (Zhao et al. 2020).

3.2.3.*Quality and consistency of requirements*

Requirements must fulfill the criteria mentioned in 3.2. Compliance with these criteria may be automatically checked using LLMs. Models that are based, for example, on ISO/IEC/IEEE 29148 may serve as a basis (Lubos et al. 2024). Traceability may be improved by using NLP-based models (Guo et al. 2024).

3.2.4.*Administration and maintenance of requirements*

Safety-related requirements should be documented, maintained, versioned, and traceably adapted throughout their entire life cycle in response to modifications.AI models can analyze semantic relationships between requirements and identify which system elements could be affected by changes(Cheng 2024).NLP-based tools enable the automatic comparison of different versions of requirements, which helps with audits and reviews (Guo et al. 2024).

3.2.5.*Evaluation of applicability and conformity*

AI-supported processes, particularly those in natural language processing, enable efficient analysis of large amounts of text, content structuring, and comprehensible preparation(Umar and Lano 2024). LLMs can provide helpfulsupport for these activities (Lubos et al. 2024). Despite these possibilities, AI still faces various problems in this area. The results of generative models are often not comprehensibly justified, can vary, and are heavily dependent on the input data.

3.3. *System and software architecture design*

A well-designed architecture not only forms the basis for the technical implementation of safety requirements but also facilitates subsequent verification and validation (Nguyen 2023; Eisenreich et al. 2024).In particular, LLMs and related NLP technologies may perform various tasks throughout the architectural design process.

3.3.1.*Definition of system and software architecture*

AI tools may support the automatic generation of domain models and use case scenarios, the creation of multiple architecture variants, the quantitative evaluation of architectures, the documentation of architecture decisions, error detection, and the optimization of architecture models(Nguyen 2023; Eisenreich et al. 2024; Fischer and Lanquillon 2024).

3.3.2.*Evaluation of applicability and conformity*

The current literature indicates that AI tools are not yet mature enough for the automatic architectural design of safety-critical systems. Although LLMs offer support in variant creation, evaluation, and documentation, they lack domain-specific knowledge, transparency, and validation mechanisms. Nevertheless, architectural proposals generated by AI can serve as a source of ideas or a starting point for further refinement and may help to analyze design decisions more objectively(Guo et al. 2024; Nguyen 2023; Eisenreich et al. 2024; Fischer and Lanquillon 2024).

3.4. *Software unit design and implementation*

The goal of these phases is to develop a precise software unit design to support verification and implementation at the source code level.

3.4.1.*Definition of software units and software implementation*

During software unit design, LLM agents can assist with translating requirements into design structures, predicting errors, providing pattern-based design suggestions, and analyzing system graphs (Klemmer et al. 2024; Geissler et al. 2024; Belzner et al. 2024). For software implementation, code generation based on textual specifications (Klemmer et al. 2024) or test-driven code generation (Belzner et al. 2024), as well as subsequent code optimization (Klemmer et al. 2024), structural improvements, and automatic evaluation of these suggestions (Barenkamp et al. 2020) are possible.

3.4.2.*Evaluation of applicability and conformity*

Although AI-supported tools such as LLMs can be helpful in particular(early) development phases(e.g., brainstorming, restructuring existing functions, or in the early prototyping phase), their performance is not yet sufficient for the direct implementation of safety-critical functions according to safety standard ISO 26262(Umar and Lano 2024; Klemmer et al. 2024; Barenkamp et al. 2020).

3.5. *Verification and validation activities*

Verification and validation (V&V) are essential processes for ensuring functional safety in accordance with ISO 26262 andare integral to the entire safety lifecycle.

3.5.1.V*erification of requirements, software architecture, unit design, and integration*

During verification activities, AI models can provide valuable support for the tasks at hand. These essentially include the analysis of requirements quality (Ezzini et al. 2023), the verification of normative requirements (Lubos et al. 2024), the comparison of safety architectures against logical data flow (Umar and Lano 2024), the analysis and consistency check of architecture models (Fan et al. 2023), the testing of unit design principles (Cotroneo et al. 2024), the generation of unit test cases (Alshahwan et al. 2024), the detection of anomalies in integration data and log files (Landauer et al. 2023), or the simulation of interaction scenarios or the detection of error states through fault injection testing (Geissler et al. 2024).

3.5.2.*Support of validation*

Validation ensures that the system meets safety-related requirements in a real-world application context. To this end, AI models can analyze and evaluate test data from validation runs (e.g., sensor data or vehicle behavior) (Mouaffari et al. 2022). Furthermore, deep generative models (e.g., cGANs) can generate realistic, synthetic test scenarios that cover specific operating conditions (Decker et al. 2023).

3.5.6.*Evaluation of applicability and conformity*

The use of AI-supported processes can provide valuable support for verifying and validatingsafety systems in selected areas.Nevertheless, it is challenging to ensure that all generated results are verifiable, traceable, and documented to comply with ISO 26262 requirements (Landauer et al. 2023; Mouaffari et al. 2022; Decker et al. 2023; Arora et al. 2023).

5. Practical Example: Performance of AI Tools in Requirements Analysis

Section 3.2 made it clear that requirements engineering (RE) is an emerging field for supporting the development process of ISO 26262, as it can handle large amounts of text.Incorrect or incomplete requirements are among the leading causes of project delays and cost increases (Leiderer and Prasad 1995). Therefore, RE, supported by AI tools, is becoming the focus of research. A systematic analysis of 105 studies (2019–2024) on generative AI in RE reveals clear research priorities (Cheng 2024). The identification and analysis of requirements dominate research activities, with LLMs considered promising in this area. Technological focal points include automated requirements identification(Almonte et al. 2025), classification and structuring (Xanthopoulou et al. 2025; Alhoshan et al. 2023), quality assessment and analysis of requirements (Lubos et al. 2024), and traceability and change management (Guo et al. 2024).

5.1.*Methodological basics for the use of generative AI in RE*

Studies show that the quality of generative AI results depends on the choice of prompt techniques, appropriate quality criteria, and clearly defined evaluation procedures.The literature distinguishes four dominant learning paradigms for RE tasks: few-shot learning, zero-shot learning, chain-of-thought, and one-shot learning. In addition, there are four prompt types for LLMs in RE:Instruction-based, iterative, description-based, and question-based prompting(Cheng 2024; Ronanki et al. 2023). The study results suggest that a combination of few-shot learning and instruction-based prompts is the most promising strategy for requirement generation. Iterative prompting approaches can also be helpful.Usingan actual research project on the topic ‘Robustness of sensors and sensor systems against environmental conditions for highly automated driving’, exemplary requirements engineering was conducted with support from the large language model GPT-4. The outputs were analyzed, and the key results are summarized in the following two sections.

5.2.*Scenario 1: Deriving stakeholder requirements*

This scenario examined whether GPT-4 could derive valid stakeholder requirements from use cases described in natural language. The generated requirements were compared with the requirements manually created in the research project to evaluate consistency and quality.Three use cases from the project were used for the investigation:

(i) ROS 498: Measurement data transfer/recording
(ii) ROS-499: Maintenance/assembly/disassembly
(iii) ROS-500: Switch on/off/standby

Three prompt strategies were used to control the requirements generation: Instruction-based, few-shot, and question-based prompting. For evaluation purposes, each generated requirement was compared with the corresponding ground-truth requirement. The assessment was carried out on a 4-point scale, see Table 1.

Table 1. Evaluation of generated requirements

4	Very high requirement match

3	High requirement match
2	Partial requirement match
1	No/low requirement match

As a result of the analysis, the average match value per use case between the GPT-generated requirements and the ground-truth requirements was determined. The results are summarized in Table 2. The evaluation shows that GPT-4 is capable of generating relevant stakeholder requirements from natural-language use cases.

Table 2. Average match value per use case

Use Case	Av. Value	Interpretation
ROS-498	2.08	Moderate to good match
ROS-499	1.93	Partially correct, but incomplete
ROS-500	2.75	Weak to moderate match

In many cases, key aspects were correctly identified and meaningfully translated into requirements.Fundamental problems included a lack of precision in the formulations, an inappropriate level of abstraction, and a limited range of contexts. These gaps indicate thatGPT-4 is linguistically powerful but lacks the domain-specific background knowledge that human experts are expected to have. These results reflect different technical weaknesses of generative AI models, including limited controllability of the model's output, limited reproducibility in some cases, and limited explainability.

5.3.Scenario 2: Quality assessment of existing requirements

This second scenario examines whether GPT-4 is capable of evaluating the formal quality of already formulated system requirements. Real system requirements from the exemplary research project served as the basis. These were created manually and cover various aspects of the system. The aim was to check whether GPT-4 can identify weaknesses in the requirements and provide suggestions for improvement. Before the evaluation, GPT-4 was given a brief contextual description of the project to understand the project's scope and requirements. An instruction-based prompt was then used in the zero-shot learning paradigm to instruct GPT-4 to check each requirement for the quality characteristics of uniqueness, completeness, verifiability, and comprehensibility.Overall, the evaluation shows that all GPT assessments are formally comprehensible and plausible. The comments are consistent with recognized RE criteria and do not contain any illogical or "hallucinatory" suggestions.This shall be demonstrated using the followingsystem requirement as an example:

"*The onboard sensor technology should include a data processing unit for classifying weather phenomena.*"

GPT-4 delivered the following analysis results:

(i) **Uniqueness:**Ambiguous, unclear which weather phenomena (*Not OK*)
(ii) **Completeness:**No details on data processing (*Not OK*)
(iii) **Verifiability:**No specification for measurement data (*Not OK*)
(iv) **Comprehensibility:** Logically derivable from context (*OK*)

At the same time, it is apparent that GPT-4 primarily evaluates requirements based on linguistic and formal criteria, without reliably distinguishing between the respective requirement levels. As a result, unnecessary detail is sometimes added, or terms are classified as unclear that are actually unambiguous in the project context. From a technical perspective, it was also noted that GPT does not correctly classify terms such as meteorologically defined visibility, which is due to a lack of domain-specific knowledge.

6. Conclusion

This contribution offers a broad literature survey of how AI tools might be applied at different stages of the V-model in ISO 26262. As the field of AI-supported software engineering is already well-established, the focus was on software development. The literature review and the exemplaryapplication to an own development project have shown that the application of AI tools in the software development process may enhance efficiency and assist engineers in the safety development process. Especially requirements engineering and documentation activities deliver promising results, andthe creativity of engineers might be stimulated. Nevertheless, there remain challenges regarding the limited explainability, reproducibility,and correctness of the results. To fulfill the safety requirements of ISO 26262, all AI-generated artefacts must be manually verified and validated. A human-in-the-loop approach could solve these problems, but it must be considered that humans tend to switch from task-oriented to target-oriented behavior (VDI-EE 4030 2022). This means that if the AI tools in general deliver good results, the human verification and validation activities might be neglected over time. Another problem that could not be discussed in depth is the absence of clear qualification procedures for AI tools.ISO 26262 requires the qualification of tools used in the development process (ISO 26262 2018). A possible solution was suggested in a white paperon trusted AI (Winter et al. 2021). And finally,everyone who uses server-based AI tools should consider that technical secrets may be disclosed to the AI tool provider.

References

Almonte, J. T., S. A. Boominathan et al. (2025). Automated Non-Functional Requirements Generation in Software Engineering with Large Language Models: A Comparative Studyhttp://arxiv.org/pdf/2503.15248v1.

Alhoshan, W. et al. (2023).*Zero-Shot Learning for Requirements Classification: An Exploratory Study*. http://arxiv.org/pdf/2302.04723v2.

Alshahwan, N. et al. (2024).*Automated Unit Test Improvement using Large Language Models at Meta*. http://arxiv.org/pdf/2402.09171v1.

Arora, C., J. Grundy, and M. Abdelrazek (2023).*Advancing Requirements Engineering through Generative AI: Assessing the Role of LLMs*. http://arxiv.org/pdf/2310.13976v2.

Baqar, M. and R. Khanda (2025). *The Future of Software Testing: AI–Powered Test Case Generation and Validation*. Intelligent Computing. Lecture Notes in Networks and Systems, Vol. 1424, pp 276–300. Springer.

Barenkamp, M., J. Rebstadt, and O. Thomas, *Applications of AI in classical software engineering*.DOI: 10.1186/s42467-020-00005-4.

Barreto J., R. A. and Z. Bachir (2020). *SAHARA: Simulation Aided Hazard Analysis and Risk Assessment Methodology*. Risk Analysis XII, pp. 41 – 53. DOI:10.2495/RISK200041.

Belzner, L., T. Gabor, and M. Wirsing (2024). *Large Language Model Assisted Software Engineering: Prospects, Challenges, and a Case Study*. https://doi.org/10.1007/978-3-031-46002-9_23.

Bozyigit, F. et al. (2024). *Generating domain models from natural language text using NLP: a benchmark dataset and experimental comparison of tools*. DOI: 10.1007/s10270-024-01176-y.

Braun, M., J. P. Leuteritz, M. Link, H. Zaiser, and D. Kutzias (2023). *Leitfaden zu Strategie und Wandel für den KI-Einsatz*. Frauenhofer IAO. https://www.digital.iao.fraunhofer.de/de/publikationen/Leitfaden_StrategieUndWandel_KI-Einsatz.html (accessedDec. 15, 2025).

Bringmann, B. (2025). *KI-Studie: Beschleunigung der KI-Transformation*. https://www.deloitte.com/de/de/ Industries/technology/research/ki-studie.html. (accessedDec. 15, 2025).

Cheng, H. et al. (2024). Generative AI for Requirements Eng.: A Systematic Literature Review. http://arxiv.org/pdf/2409.06741v2.

Cotroneo, D. et al. (2024)DeVAIC: A Tool for Security Assessment of AI-generated Code. http://arxiv.org/pdf/2404.07548v2.

Decker, T., A. R. Bhattarai, and M. Lebacher (2023). *Towards Scenario-based Safety Validation for Autonomous Trains with Deep Generative Models*.DOI: 10.1007/978-3-031-40923-3_20.

Diemert S. and J. H. Weber (2023). Can Large Language Models assist in Hazard Analysis?http://arxiv.org/pdf/2303.15473v1.

Eisenreich, T. et al. (2024). From Requirements to Architecture: An AI-Based Journey to Semi-Automatically Generate Software Architectures.DOI: 10.1145/3643660.3643942.

EU AI Act (2024). *REGULATION (EU) 2024/1689*. Official Journal of the European Union.

Ezzini, S. et al.(2023). *AI-based Question Answering Assistance for Analyzing Natural-language Requirements*. http://arxiv.org/pdf/2302.04793v1.

Fan, G., X. Xie, X. Zheng, Y. Liang, and Di Peng (2024). *Static Code Analysis in the AI Era: An In-depth Exploration of the Concept, Function, and Potential of Intelligent Code Analysis Agents*. http://arxiv.org/pdf/2310.08837v1.

Fischer M. and C. Lanquillon (2024).*Evaluation of Generative AI-Assisted Software Design and Engineering: A User-Centered Approach*. https://doi.org/10.1007/978-3-031-60606-9_3.

Geissler, F. et al. (2024). *Concept-Guided LLM Agents for Human-AI Safety Codesign*. https://arxiv.org/abs/2404.15317.

Guo J. L. C., J.-P. Steghöfer, A. Vogelsang, and J. Cleland-Huang. Natural Language Processing for

Requirements Traceability. Available: http://arxiv.org/pdf/2405.10845v1.

Hassija, V. et al. (2023). *Interpreting Black* Box *Models: A Review on Explainable Artificial Intelligence*. https://doi.org/10.1007/s12559-023-10179-8 (accessed Dec. 15, 2025).

Hillen, D., C. Helten, and J. Reich. *Towards LLM-augmented Situation Space Analysis for the Hazard and Risk Assessment of Automotive Systems*. DOI: 10.18420/inf2024_58.

Hoffmann, J. (2025). Hoffmann, J. *ISO 26262 Edition 3: Standardization Timing, Vocabulary, and Management of Functional Safety*. https://sres.ai/functional-safety/iso-26262-edition-3-standardization-timing-vocabulary-and-management-of-functional-safety/ (accessed Dec. 15, 2025).

ISO 21448 (2022). *Road vehicles — Safety of the intended functionality*. International Standard of the International Organization for Standardization (ISO).

ISO 26262 (2018). *Road Vehicles — Functional Safety, Parts 1 to12*. International Standard of the International Organization for Standardization (ISO).

ISO/IEC TR 5469 (2021).*Artificial Intelligence – Functional Safety and AI Systems*, International Organization for Standardization (ISO) und International Electrotechnical Commission (IEC).

ISO/PAS 8800 (2024). *Road vehicles — Safety and artificial intelligence*. International Standard of the International Organization for Standardization (ISO).

Khade, P. S. and Sambhe, R. U. (2025). *Artificial Intelligence in Software Development: A Review of Code Generation, Testing, Maintenance, and Security*. DOI: https://doi.org/10.47191/ijcsrr/V8-i4-08.

Klemmer, J. H. et al. (2024).*Using AI Assistants in Software Development: A Qualitative Study on Security Practices and Concerns*. DOI: 10.1145/3658644.3690283.

Landauer, M. et al. (2023). *Deep learning for anomaly detection in log data: A survey*.DOI: 10.1016/j.mlwa.2023.100470.

Lederer, M.,and J. Prasad (1995). *Causes of inaccurate software development cost estimates*.DOI: 10.1016/0164-1212(94)00092-2.

Lubos S. et al. (2024).*Leveraging LLMs for the Quality Assurance of Software Requirements*. DOI: 10.1109/RE59067.2024.00046.

Mozaffari, S. et al. (2022). *Deep Learning-Based Vehicle Behavior Prediction for Autonomous Driving Applications: A Review*. DOI: 10.1109/TITS.2020.3012034.

Nguyen-Duc, A. et al. (2023). Generative Artificial Intelligence for Software Eng. - A Research Agenda. http://arxiv.org/pdf/2310.18648v1.

Nivedhaa, N. (2024). *A Comprehensive Review of AI's Dependence on Data*. International Journal of Artificial Intelligence and Data Science (IJADS). Volume 1, Issue 1, pp. 1-11, Article ID: IJADS_01_01_001.

Nouri, A. et al. (2024). *Welcome Your New AI Teammate: On Safety Analysis by Leashing Large Language Models*. Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI, Lisbon, Portugal, pp. 172–177.

Ronanki, K. et al. (2023). *Requirements Engineering using Generative AI: Prompts and Prompting Patterns*. http://arxiv.org/pdf/2311.03832v1.

Siebert, J. (2025). *Halluzinationen von generativer KI und großen Sprachmodellen (LLMs)*. https://www.iese.fraunhofer.de/blog/halluzinationen-generative-ki-llm/ (accessedDec. 15, 2025).

Siebert, J. and Jedlitschka, A. (2024). *Generative KI im Software Engineering: Szenarien und künftige Herausforderungen*. https://www.iese.fraunhofer.de/blog/generative-ki-softwareentwicklung/ (accessedDec. 15, 2025).

Talele P. and R. Phalnikar (2021). *Classification and Prioritisation of Software Requirements using Machine Learning – A Systematic Review*. DOI: 10.1109/Confluence51648.2021.9377190.

UL Solutions.(2024) *What to Expect with Version 3 of ISO 26262*.https://www.ul.com/sis/blog/what-to-expect-with-version-3-of-iso-26262 (accessed Dec. 15, 2025).

Umar, M. A. and K. Lano (2024). *Advances in automated support for requirements engineering: a systematic literature review*. DOI: 10.1007/s00766-023-00411-0.

VDI-EE 4030 (2022). *Consideration of human reliability in the design of autonomous systems*. DIN Media Verlag. Berlin

Wang Y. and S. H. Chung (2022). *Artificial intelligence in safety-critical systems: a systematic review*. IMDS, Vol. 122, no. 2, pp. 442–470, DOI: 10.1108/IMDS-07-2021-0419.

White J. et al. (2024).*A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT*. http://arxiv.org/pdf/2302.11382v1

Winter, P. M. et al. (2021).*Trusted Artificial Intelligence: Towards Certification of Machine Learning Applications*. TÜV Austria White Paper. http://arxiv.org/pdf/2103.16910v1

Xanthopoulou, G., M. Siavvas, I. Kalouptsoglou et al. (2025). *Software Requirements Classification: From Bag-of-Words to Transformer*. https://doi.org/10.1007/978-3-031-76459-2_35.

Zhao L. et al. (2020). *Natural Language Processing (NLP) for Requirements Engineering: A Systematic Mapping Study*. http://arxiv.org/pdf/2004.01099v2.