Qt Quality Assurance

GUIDE

Functional Safety According to ISO 26262

ASIL-A, ASIL-B, ASIL-C and ASIL-D

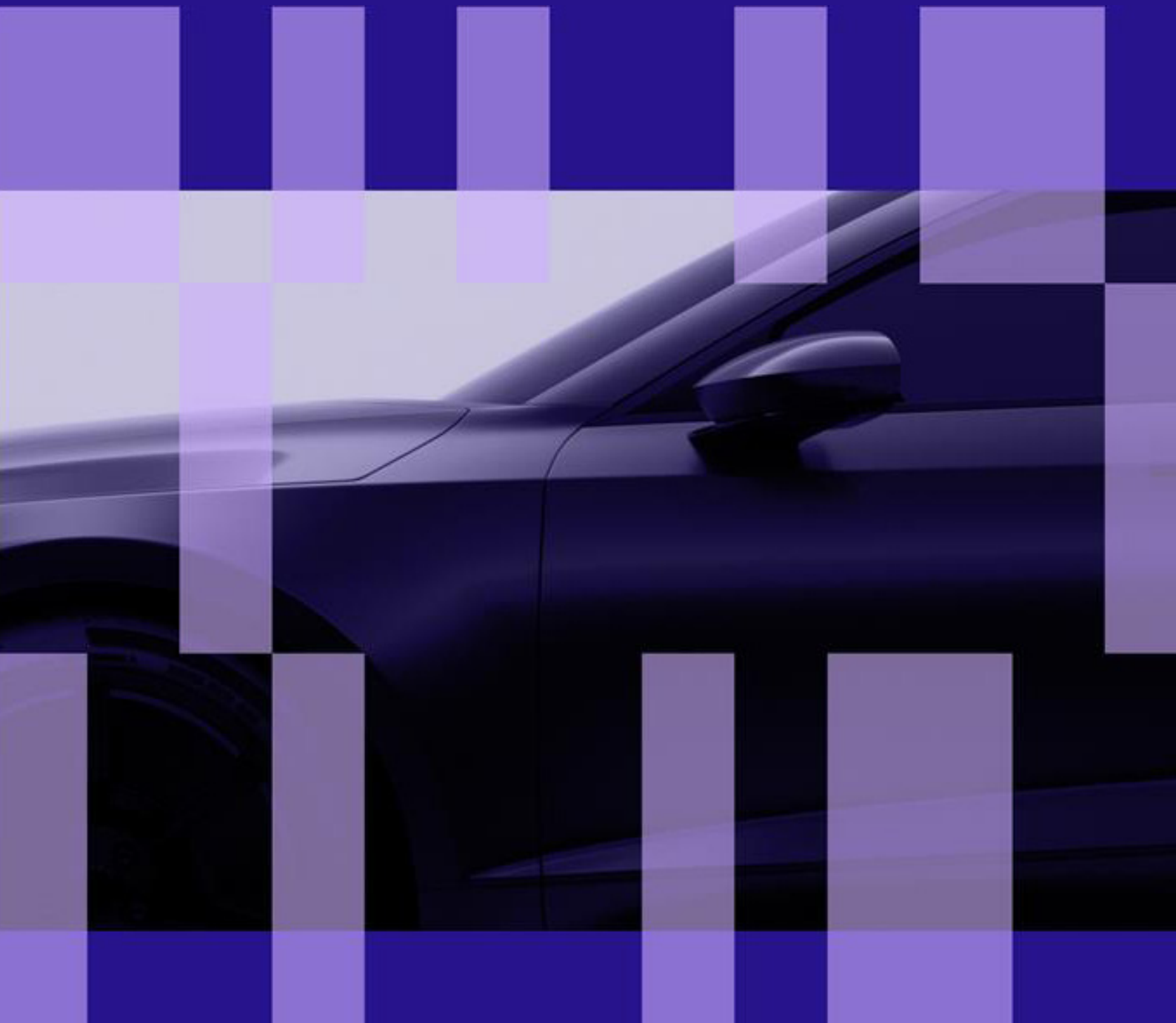


Table of Contents

Meet our Experts

Complying with ISO 26262 when developing safety-critical software can be challenging.

Our experts have years of experience in the automotive industry and can support you.

Get in touch to learn more about our tools and services or to request a demo.

Contact us
www.qt.io/axivion

Introduction

The ISO 26262 standard is the adaptation of IEC 61508 and ensures the functional safety of electrical and electronic (E/E) systems in road vehicles. The standard introduces Automotive Safety Integrity Levels (ASILs), ranging from A (least stringent) to D (most stringent). These levels determine how rigorously safety requirements must be verified.

Axivion Suite, the combination of Axivion Static Code Analysis (SCA) and Axivion Architecture Verification (AV), is the ideal tool to endure the software meets the ISO 26262 requirements up to ASIL-D. It covers the verification steps defined in ISO 26262:2018 with a focus on static code and design checks.

In an increasingly complex automotive software landscape, where software-defined vehicles are becoming the norm, tools like Axivion are essential for maintaining safety, traceability, and compliance. Its integration into modern development pipelines ensures that safety is not an afterthought but a continuous, verifiable process throughout the software lifecycle.

Axivion SCA has been certified by SGS-TÜV Saar GmbH for use in the development of safety systems according to ISO 26262 up to ASIL D.

Covering the Safety Case

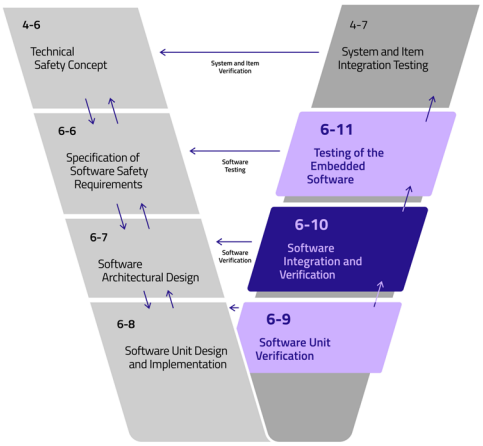
In ISO 26262, the safety case is a structured, evidence-based argument that an E/E system is acceptably safe for its intended use. It encompasses all phases of the development lifecycle, including requirements engineering, design, implementation, verification, validation, and configuration management. The development process directly influences whether the required functional safety is achieved.

Axivion contributes directly to the safety case by providing traceable, repeatable, and certifiable analyses. Its integration into the development workflow ensures that static code analysis and architecture verification are consistently applied, supporting the demonstration of compliance with safety goals specific to each project.

By embedding Axivion into the safety case, organizations can streamline audits and assessments by certification bodies. The tool's ability to generate detailed reports and maintain historical analysis data supports transparency and accountability, which are critical for demonstrating due diligence in safety-critical development.

Position in the V-Model

Axivion aligns with ISO 26262-6, which governs software-level product development. Within the V-Model framework, Axivion supports activities across software unit verification, software integration, and embedded software testing.



Its applicability spans all ASIL levels, from QM (Quality Management) through ASIL A to ASIL D, making it a versatile tool for projects with varying safety requirements. This flexibility allows development teams to maintain consistent tool usage across different safety-critical components.

Furthermore, Axivion supports traceability across the V-Model by linking architectural elements and code artifacts to requirements and test cases. This traceability is essential for impact analysis, change management, and ensuring that all safety requirements are adequately addressed throughout the development lifecycle.

Support for Modeling and Coding Guidelines

By automating the enforcement of these guidelines, Axivion Suite reduces the risk of human error and ensures consistent application of safety principles across teams and projects. This not only improves code quality but also accelerates compliance efforts by embedding best practices directly into the development workflow.

Axivion enforces key software development principles that are critical for safety compliance, including:

- Enforcing low complexity
- Use of well trusted design principles
- Use of unambiguous graphical representation
- Use of language subset
- Enforcement of strong typing
- Use of defensive implementation techniques
- Use of style guides
- Use of naming conventions
- Concurrency aspects

		ASIL				Axivion Suite							
	Topics	A	B	C	D	Architecture Verification	Clone Detection aâ Management	Cycle Detection	Dead Code Analysis	Metrics Monitoring	Coding Guidelines	Static / Semantic Code Analysis	Data Races
1a	Enforcement of low complexity	++	++	++	++	•	•	•	•	•	•		
1b	Use of well-trusted design principles	+	+	++	++	•	•	•		•			
1c	Use of unambiguous graphical representation	+	++	++	++	•							
1d	Use of language subset	++	++	++	++						•		
1e	Enforcement of strong typing	++	++	++	++						•	•	
1f	Use of defensive implementation techniques	+	+	++	++		•				•	•	
1g	Use of style guides	+	++	++	++						•	•	
1h	Use of naming conventions	++	++	++	++	•					•		
1i	Concurrency aspects	+	+	+	+							•	•

Topics to be covered by modeling and coding guidelines according to ISO 26262-6:2018

These principles are supported through features such as architecture analysis, clone detection, cycle detection, unreachable code identification, and metrics computation. The suite also enforces coding standards including MISRA C, MISRA C++, AUTOSAR C++14, and CERT, ensuring alignment with industry best practices.

Analyses at System Level and for Day-to-Day Use

Axivion integrates seamlessly with continuous integration (CI) systems, enabling automated, repeatable analyses that enhance traceability and consistency. Its delta analysis mechanism allows developers to monitor changes over time, facilitating regression detection and impact analysis.

For day-to-day development, Axivion supports local analyses within integrated development environments (IDEs), providing immediate feedback and reducing turnaround times. This dual capability ensures both strategic system-level insights and tactical developer-level support.

This combination of system-wide and developer-centric analysis empowers teams to maintain high software quality without sacrificing agility. Developers receive actionable insights early in the development cycle, while project leads and safety managers benefit from comprehensive overviews of system health and compliance status.

Architectural Analyses

Axivion's architecture verification capabilities allow for the validation of implementation against design specifications, whether expressed informally (e.g., box-and-arrow diagrams) or formally (e.g., UML models). The hypothesis-driven architecture recovery feature enables reconstruction of architecture from legacy systems or undocumented codebases.

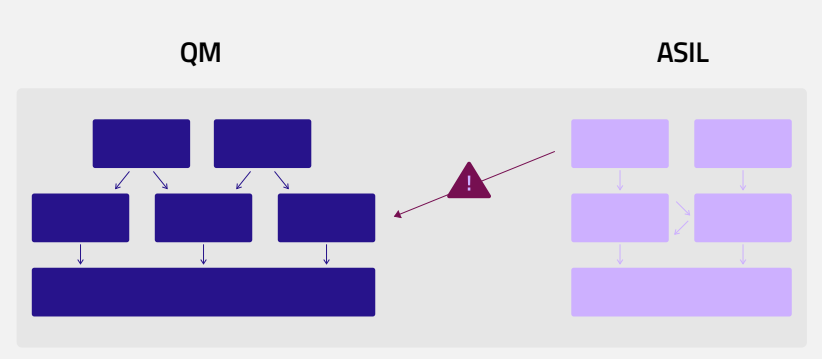
For AUTOSAR-based projects, Axivion supports arxml import and validation against architectural models created in tools like IBM Rhapsody or Enterprise Architect. This ensures consistency between design intent and implementation, even in complex, model-driven environments.

By bridging the gap between design and implementation, Axivion Architecture Verification helps prevent architectural erosion—a common issue in long-lived automotive software projects. This ensures that the system remains maintainable, scalable, and aligned with safety and performance goals throughout its lifecycle.

Freedom from Interference

Freedom from interference is an important goal in safety projects and is defined in ISO 26262-1:2018. It is essential for ensuring that components of differing ASIL levels do not adversely affect each other. Axivion identifies unintended interactions at the source code level, such as misdirected function calls between components of mixed criticality. This is done by detecting misdirected calls between components with mixed criticality ASIL (Automotive Safety Integrity Level) classification or between ASIL and QM partitions. In this way, the frequency of MMU/MPU exceptions in the later development phases with dynamic execution are reduced and, thus saving development time.

In addition, Axivion's visualization tools help developers and architects understand the interdependencies between software modules, making it easier to design and verify safe partitioning strategies. This is particularly valuable in multicore and mixed-criticality systems where interference risks are elevated.



By detecting these issues early, Axivion helps prevent runtime exceptions and reduces the need for costly debugging in later development stages. This proactive approach supports robust partitioning and isolation strategies, contributing to overall system safety.

Requirements Decomposition

ASIL decomposition, as described in ISO 26262-9:2018, allows for the distribution of safety requirements across independent architectural elements. Axivion verifies that the software architecture supports the required independence, ensuring that decomposed requirements are correctly implemented and isolated.

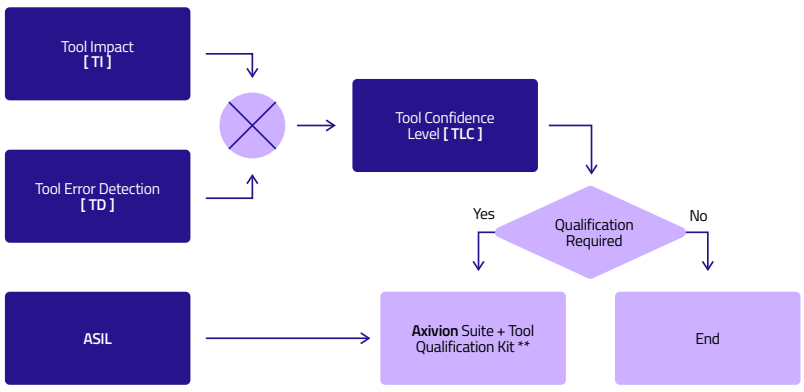
This capability enables cost-effective safety strategies while maintaining compliance, particularly in systems where full ASIL D implementation may be impractical or unnecessary.

Axivion's ability to validate architectural independence at the code level ensures that decomposition strategies are not only theoretically sound but also practically enforced. This reduces the risk of latent design flaws and supports the creation of modular, reusable components across projects.

Tool-Qualification

Tool qualification is essential when using software tools in safety-critical development. Depending on the required Tool Confidence Level (TCL), Axivion must be qualified to ensure its outputs can be trusted without additional verification.

Axivion Static Code Analysis provides a comprehensive Tool Qualification Kit, which includes predefined code violations and test cases for MISRA, CERT, and AUTOSAR C++14. This kit supports qualification for any ASIL level and ensures that the toolchain meets the necessary confidence requirements. The qualification process is supported by detailed documentation and test artifacts, simplifying the certification process for development teams. This reduces the overhead typically associated with tool qualification and accelerates project timelines without compromising safety or compliance.



** Quality tool with appropriate degree of vigour

Qt Quality Assurance

www.qt.io/axivion