

# ISO26262 Automotive Safety (DE0204)

## Automotive ISO26262: Road Vehicles Functional Safety

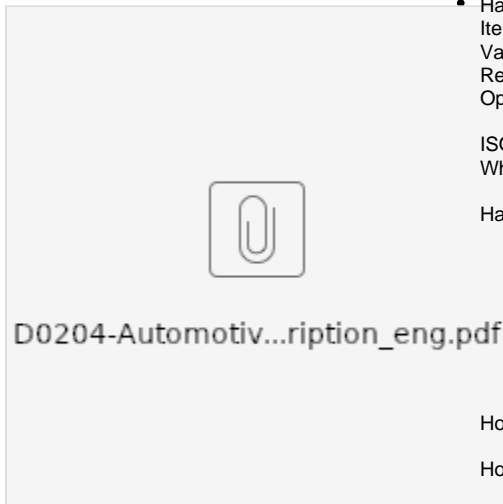
Who should attend:

- Safety Managers
- Development Engineers (System, Hardware and Software)
- Product Managers
- Project Leaders of safety related development projects
- Managers responsible for establishment of work processes
- Quality managers

**Duration:** 4 days

**Language:** German or English, training material will be in English.

### Brochure



Course topics:

- Functional Safety Overview / General Introduction
  - Why is Functional Safety needed in Automotive Industry?
- FS Management & Safety Life Cycle Requirements (ISO 26262 - Part 2)
  - Activities for Functional Safety Management
  - Safety Plan
  - Safety Life Cycle Management
- Risk Based Safety Requirements Engineering (ISO 26262 - Part 3)
- Item Definition
- Hazard Analysis & Risk Assessment
  - Safety Goal Definition with ASIL with examples and exercises
- Functional Safety Concept
  - Warning and degradation concept
  - System Development
- Technical Safety Concept (ISO 26262 - Part 4)
  - Safety Architecture and Architectural Elements
  - Safety Functions and Safety Integrity Function
  - Exemplification: typical solutions detailed in a technical safety concept, requirements allocation to system and ASIL Decomposition example
  - Verification: System Safety FMEA and FTA
- Hardware-Software-Interface Specification: The ISO solution for an old problem
  - Item integration and testing
  - Validation
  - Release for Production
  - Optional:
    - Dependant failure analysis (DFA)
  - ISO 26262 lifecycle approach: Product Lifecycle and process requirements
  - Where is hardware development in the process model?
    - What are inputs to hardware development?
  - Hardware Development (ISO 26262 - Part 5)
    - HW Safety Requirements
    - HW Architecture and Design with examples
    - HW Architecture Evaluation
      - Introduction into Fault Tree Analysis with exercise
      - Requirements for the Evaluation: Metrics for Safety Goal Violation
      - Fault models, failure rates and target values
      - Presentation of the probabilistic approach
      - qualitative approach with a semi probabilistic argumentation
  - How to evaluate the metric for "Safety Goal Violation"
    - exemplification: calculation via FTA based on the results of the quantitative FMEDA
  - How to evaluate the metrics SPFM and LFM
    - exemplification: exida FMEDA approach for metric calculation
- HSI - Cooperation with the software team
  - Hardware-Software-Interface Specification HSI
- Qualification of HW components (ISO26262 - Part 8 § 13)
- Optional:
  - ASICs in the scope of the ISO 26262
  - Communication channels and their evaluation
  - Dependant failure analysis (DFA)
- Software Development Process (ISO 26262 - Part 6)
  - Content of the Software Safety Process, how to do initiation and tailoring
- Software Safety Requirements Specification:
  - Sources of Software Safety Requirements and interfaces to System Level
  - Interpretation of properties and attributes required by ISO26262
  - Practical methods how to derive and detail requirements for the software
  - Methods for the verification of Software Safety Requirements
- Software Architecture:
  - How to develop a Software Architecture acc. to ISO26262
  - Semi-formal architecture development - interpretation of ISO26262
  - Measures to be considered
  - ASIL Decomposition at the software level
  - How to implement Freedom from Interference
  - Safety requirements allocation to software architectural components
  - Methods for the verification of the Software Architecture
- Software Unit Design
  - Content of a semi-formal Software Unit Design Specification
  - Interpretation of design requirements recommended by ISO26262
  - How to deal with OO programming languages (C++)?
  - Methods for the verification of the software unit design
- Software Safety Verification
  - Software Analysis Techniques
    - Software Criticality Analysis

Scheduled courses -

[Register here:](#)

- Software Dependent Failure Analysis
- Software Testing Techniques (Unit and Integration Testing)
  - Requirements based (Equivalence Classes, Boundary Values, etc.)
  - Structure based (Statement Coverage, MCDC, Call Coverage, etc.)
- Tool classification and qualification
- Software qualification: How to deal with existing components?