INTERNATIONAL STANDARD

ISO 26262-4

Second edition
2018-12

Road vehicles — Functional safety —

Part 4: Product development at the system level

Véhicules routiers — Sécurité fonctionnelle —

Partie 4: Développement du produit au niveau du système

Reference number
ISO 26262-4:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 22, *Road vehicles Subcommittee*, SC 32, *Electrical and electronic components and general system aspects*.

This edition of ISO 26262 series of standards cancels and replaces the edition ISO 26262:2011 series of standards, which has been technically revised and includes the following main changes:

— requirements for trucks, buses, trailers and semi-trailers;

— extension of the vocabulary;

— more detailed objectives;

— objective oriented confirmation measures;

— management of safety anomalies;

— references to cyber security;

— updated target values for hardware architecture metrics;

— guidance on model based development and software safety analysis;

— evaluation of hardware elements;

— additional guidance on dependent failure analysis;

— guidance on fault tolerance, safety related special characteristics and software tools;

— guidance for semiconductors;

— requirements for motorcycles; and

— general restructuring of all parts for improved clarity.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

A list of all parts in the ISO 26262 series can be found on the ISO website.

Introduction

The ISO 26262 series of standards is the adaptation of IEC 61508 series of standards to address the sector specific needs of electrical and/or electronic (E/E) systems within road vehicles.

This adaptation applies to all activities during the safety lifecycle of safety-related systems comprised of electrical, electronic and software components.

Safety is one of the key issues in the development of road vehicles. Development and integration of automotive functionalities strengthen the need for functional safety and the need to provide evidence that functional safety objectives are satisfied.

With the trend of increasing technological complexity, software content and mechatronic implementation, there are increasing risks from systematic failures and random hardware failures, these being considered within the scope of functional safety. ISO 26262 series of standards includes guidance to mitigate these risks by providing appropriate requirements and processes.

To achieve functional safety, the ISO 26262 series of standards:

a) provides a reference for the automotive safety lifecycle and supports the tailoring of the activities to be performed during the lifecycle phases, i.e., development, production, operation, service and decommissioning;

b) provides an automotive-specific risk-based approach to determine integrity levels [Automotive Safety Integrity Levels (ASILs)];

c) uses ASILs to specify which of the requirements of ISO 26262 are applicable to avoid unreasonable residual risk;

d) provides requirements for functional safety management, design, implementation, verification, validation and confirmation measures; and

e) provides requirements for relations between customers and suppliers.

The ISO 26262 series of standards is concerned with functional safety of E/E systems that is achieved through safety measures including safety mechanisms. It also provides a framework within which safety-related systems based on other technologies (e.g. mechanical, hydraulic and pneumatic) can be considered.

The achievement of functional safety is influenced by the development process (including such activities as requirements specification, design, implementation, integration, verification, validation and configuration), the production and service processes and the management processes.

Safety is intertwined with common function-oriented and quality-oriented activities and work products. The ISO 26262 series of standards addresses the safety-related aspects of these activities and work products.

Figure 1 shows the overall structure of the ISO 26262 series of standards. The ISO 26262 series of standards is based upon a V-model as a reference process model for the different phases of product development. Within the figure:

— the shaded “V”s represent the interconnection among ISO 26262-3, ISO 26262-4, ISO 26262-5, ISO 26262-6 and ISO 26262-7;

— for motorcycles:

  — ISO 26262-12:2018, Clause 8 supports ISO 26262-3;

  — ISO 26262-12:2018, Clauses 9 and 10 support ISO 26262-4;

— the specific clauses are indicated in the following manner: “m-n”, where “m” represents the number of the particular part and “n” indicates the number of the clause within that part.

EXAMPLE “2-6” represents ISO 26262-2:2018, Clause 6.

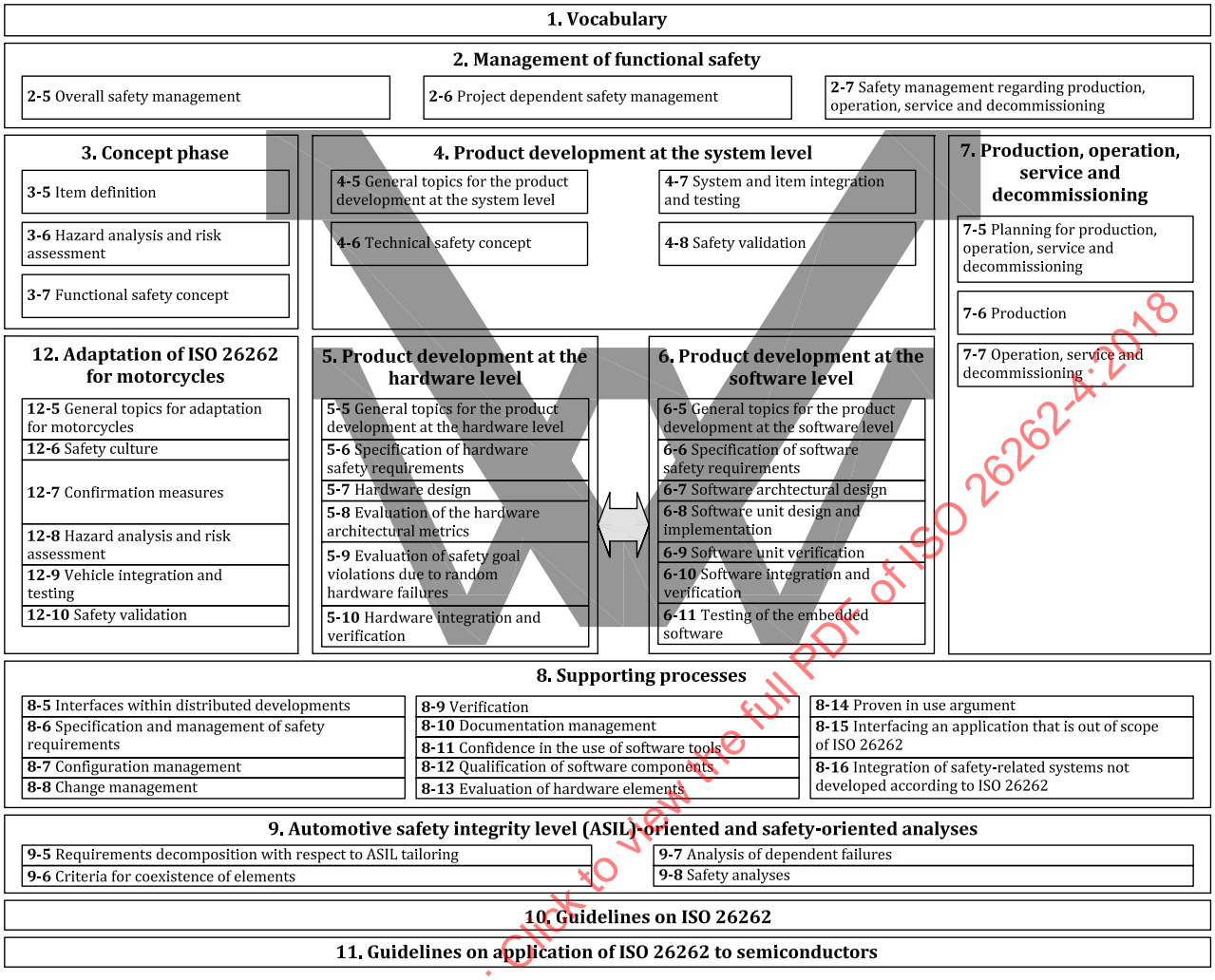

Figure 1 — Overview of the ISO 26262 series of standards

Road vehicles — Functional safety —

Part 4:
Product development at the system level

1 Scope

This document is intended to be applied to safety-related systems that include one or more electrical and/or electronic (E/E) systems and that are installed in series production road vehicles, excluding mopeds. This document does not address unique E/E systems in special vehicles such as E/E systems designed for drivers with disabilities.

NOTE Other dedicated application-specific safety standards exist and can complement the ISO 26262 series of standards or vice versa.

Systems and their components released for production, or systems and their components already under development prior to the publication date of this document, are exempted from the scope of this edition. This document addresses alterations to existing systems and their components released for production prior to the publication of this document by tailoring the safety lifecycle depending on the alteration. This document addresses integration of existing systems not developed according to this document and systems developed according to this document by tailoring the safety lifecycle.

This document addresses possible hazards caused by malfunctioning behaviour of safety-related E/E systems, including interaction of these systems. It does not address hazards related to electric shock, fire, smoke, heat, radiation, toxicity, flammability, reactivity, corrosion, release of energy and similar hazards, unless directly caused by malfunctioning behaviour of safety-related E/E systems.

This document describes a framework for functional safety to assist the development of safety-related E/E systems. This framework is intended to be used to integrate functional safety activities into a company-specific development framework. Some requirements have a clear technical focus to implement functional safety into a product; others address the development process and can therefore be seen as process requirements in order to demonstrate the capability of an organization with respect to functional safety.

This document does not address the nominal performance of E/E systems.

This document specifies the requirements for product development at the system level for automotive applications, including the following:

— general topics for the initiation of product development at the system level;

— specification of the technical safety requirements;

— the technical safety concept;

— system architectural design;

— item integration and testing; and

— safety validation.

Annex A provides an overview on objectives, prerequisites and work products of this document.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 26262-1:2018, *Road vehicles — Functional safety — Part 1: Vocabulary*

ISO 26262-2:2018, *Road vehicles — Functional safety — Part 2: Management of functional safety*

ISO 26262-3:2018, *Road vehicles — Functional safety — Part 3: Concept phase*

ISO 26262-5:2018, *Road vehicles — Functional safety — Part 5: Product development at the hardware level*

ISO 26262-6:2018, *Road vehicles — Functional safety — Part 6: Product development at the software level*

ISO 26262-7:2018, *Road vehicles — Functional safety — Part 7: Production, operation, service and decommissioning*

ISO 26262-8:2018, *Road vehicles — Functional safety — Part 8: Supporting processes*

ISO 26262-9:2018, *Road vehicles — Functional safety — Part 9: Automotive Safety Integrity Level (ASIL)-oriented and safety-oriented analyses*

3 Terms and definitions

For the purposes of this document, the terms, definitions and abbreviated terms given in ISO 26262-1:2018 apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at https://www.iso.org/obp

4 Requirements for compliance

4.1 Purpose

This clause describes how:

a) to achieve compliance with the ISO 26262 series of standards;

b) to interpret the tables used in the ISO 26262 series of standards; and

c) to interpret the applicability of each clause, depending on the relevant ASIL(s).

4.2 General requirements

When claiming compliance with the ISO 26262 series of standards, each requirement shall be met, unless one of the following applies:

a) tailoring of the safety activities in accordance with ISO 26262-2 has been performed that shows that the requirement does not apply; or

b) a rationale is available that the non-compliance is acceptable and the rationale has been evaluated in accordance with ISO 26262-2.

Informative content, including notes and examples, is only for guidance in understanding, or for clarification of the associated requirement, and shall not be interpreted as a requirement itself or as complete or exhaustive.

The results of safety activities are given as work products. "Prerequisites" are information which shall be available as work products of a previous phase. Given that certain requirements of a clause are ASIL-dependent or may be tailored, certain work products may not be needed as prerequisites.

"Further supporting information" is information that can be considered, but which in some cases is not required by the ISO 26262 series of standards as a work product of a previous phase and which may be made available by external sources that are different from the persons or organizations responsible for the functional safety activities.

4.3 Interpretations of tables

Tables are normative or informative depending on their context. The different methods listed in a table contribute to the level of confidence in achieving compliance with the corresponding requirement. Each method in a table is either:

a) a consecutive entry (marked by a sequence number in the leftmost column, e.g. 1, 2, 3), or

b) an alternative entry (marked by a number followed by a letter in the leftmost column, e.g. 2a, 2b, 2c).

For consecutive entries, all listed highly recommended and recommended methods in accordance with the ASIL apply. It is allowed to substitute a highly recommended or recommended method by others not listed in the table, in this case, a rationale shall be given describing why these comply with the corresponding requirement. If a rationale can be given to comply with the corresponding requirement without choosing all entries, a further rationale for omitted methods is not necessary.

For alternative entries, an appropriate combination of methods shall be applied in accordance with the ASIL indicated, independent of whether they are listed in the table or not. If methods are listed with different degrees of recommendation for an ASIL, the methods with the higher recommendation should be preferred. A rationale shall be given that the selected combination of methods or even a selected single method complies with the corresponding requirement.

NOTE A rationale based on the methods listed in the table is sufficient. However, this does not imply a bias for or against methods not listed in the table.

For each method, the degree of recommendation to use the corresponding method depends on the ASIL and is categorized as follows:

— "++" indicates that the method is highly recommended for the identified ASIL;

— "+" indicates that the method is recommended for the identified ASIL; and

— "o" indicates that the method has no recommendation for or against its usage for the identified ASIL.

4.4 ASIL-dependent requirements and recommendations

The requirements or recommendations of each sub-clause shall be met for ASIL A, B, C and D, if not stated otherwise. These requirements and recommendations refer to the ASIL of the safety goal. If ASIL decomposition has been performed at an earlier stage of development, in accordance with ISO 26262-9:2018, Clause 5, the ASIL resulting from the decomposition shall be met.

If an ASIL is given in parentheses in the ISO 26262 series of standards, the corresponding sub-clause shall be considered as a recommendation rather than a requirement for this ASIL. This has no link with the parenthesis notation related to ASIL decomposition.

4.5 Adaptation for motorcycles

For items or elements of motorcycles for which requirements of ISO 26262-12 are applicable, the requirements of ISO 26262-12 supersede the corresponding requirements in this document. Requirements of ISO 26262-2 that are superseded by ISO 26262-12 are defined in Part 12.

4.6 Adaptation for trucks, buses, trailers and semi-trailers

Content that is intended to be unique for trucks, buses, trailers and semi-trailers (T&B) is indicated as such.

5 General topics for the product development at the system level

5.1 Objectives

The objective of this clause is to provide an overview of product development at the system level.

5.2 General

The necessary activities during the development of a system are given in Figure 2. In an iterative process, the technical safety concept is developed, incorporating technical safety requirements and the system architectural design. The system architecture is established, the technical safety requirements are allocated to elements of the system, and, if applicable, on other technologies. In addition, the technical safety requirements are refined and requirements arising from the system architecture are added, including the hardware-software interface (HSI). Depending on the complexity of the architecture, the requirements for subsystems can be derived iteratively.

After their development, the hardware and software elements are integrated and tested to form an item that is then integrated into a vehicle. Once integrated at the vehicle level, safety validation is performed to provide evidence of functional safety with respect to the safety goals.

This document applies to the development of systems. ISO 26262-5 and ISO 26262-6 describe the development requirements for hardware and software, respectively. Figure 3 is an example of a system with multiple levels of integration, illustrating the application of this document, ISO 26262-5 and ISO 26262-6.

NOTE 1 Table A.1 provides an overview of objectives, prerequisites and work products of the particular sub-phases of product development at the system level.

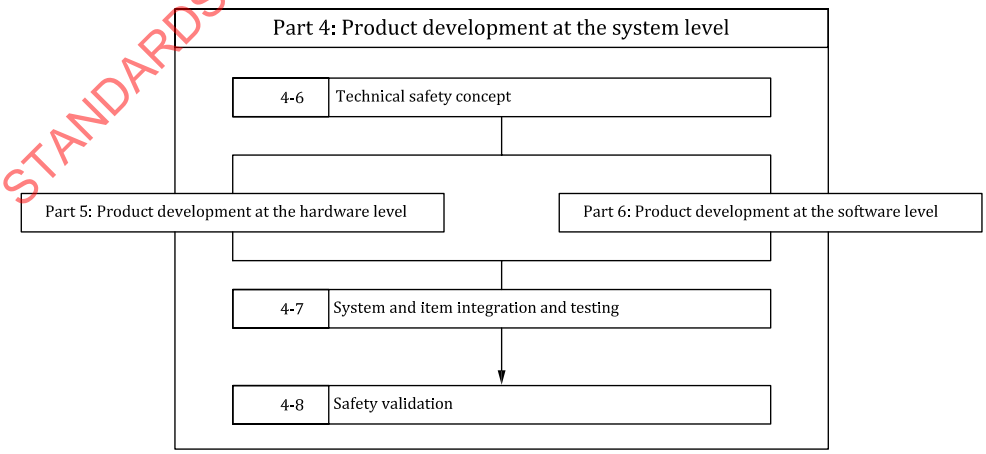

Figure 2 — Reference phase model for the development of a safety-related item

NOTE 2 Within the figures 2 and 3, the specific clauses of each part of ISO 26262 are indicated in the following manner: "m-n", where "m" represents the number of the part and "n" indicates the number of the clause, e.g. "4-6" represents ISO 26262-4:2018, Clause 6.

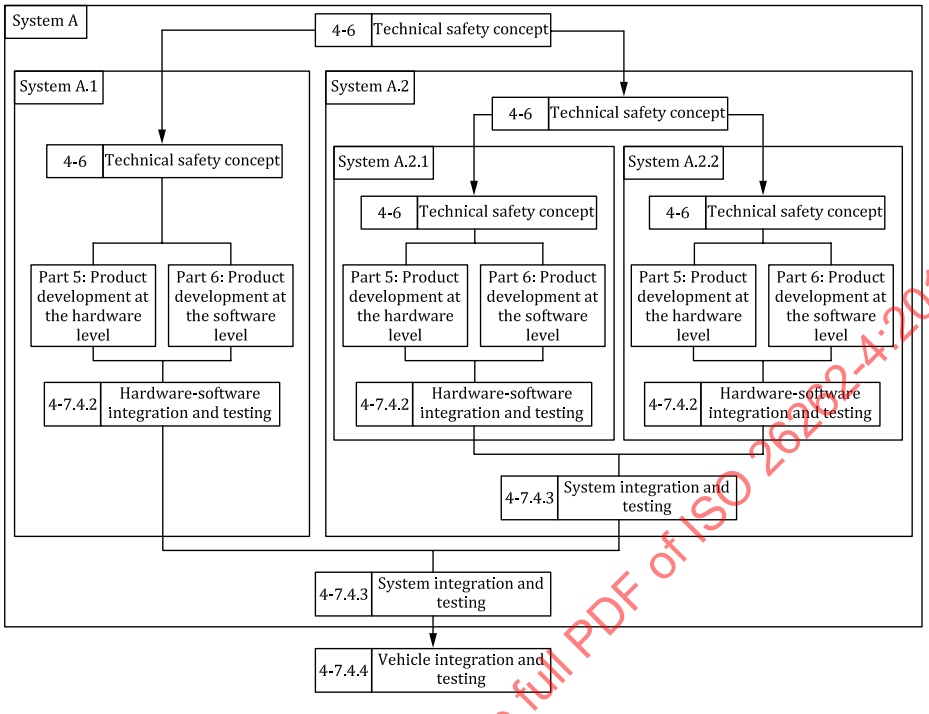

Figure 3 — Example of a product development at the system level

NOTE 3 Further information regarding product development at the system level can be found in References [1] and [2].

6 Technical safety concept

6.1 Objectives

The objectives of this clause are:

a) to specify technical safety requirements regarding the functionality, dependencies, constraints and properties of the system elements and interfaces needed for their implementation;

b) to specify technical safety requirements regarding the safety mechanisms to be implemented in the system elements and interfaces;

c) to specify requirements regarding the functional safety of the system and its elements during production, operation, service and decommissioning;

d) to verify that the technical safety requirements are suitable to achieve functional safety at the system level and are consistent with the functional safety requirements;

e) to develop a system architectural design and a technical safety concept that satisfy the safety requirements and that are not in conflict with the non-safety-related requirements;

f) to analyse the system architectural design in order to prevent faults and to derive the necessary safety-related special characteristics for production and service; and

g) to verify that the system architectural design and the technical safety concept are suitable to satisfy the safety requirements according to their respective ASIL.

6.2 General

The **technical safety concept** is an aggregation of the technical safety requirements and the corresponding system architectural design that provides rationale as to why the system architectural design is suitable to fulfil safety requirements resulting from activities described in ISO 26262-3 (with consideration of non-safety requirements) and design constraints.

The **technical safety requirements** specify the technical implementation of the functional safety requirements at their respective hierarchical level; considering both the item definition and the system architectural design, and addressing the detection of latent failures, fault avoidance, safety integrity and operation and service aspects.

The **system architectural design** is the selected system-level solution that is implemented by a technical system. The system architectural design aims to fulfil both, the allocated technical safety requirements and the non-safety requirements.

System development can be performed iteratively.

6.3 Inputs to this clause

6.3.1 Prerequisites

The following information shall be available:

— functional safety concept in accordance with ISO 26262-3:2018, 7.5.1;

— system architectural design (from an external source, see ISO 26262-3:2018, 7.3.1); and

— requirements to the item from other safety relevant items if applicable.

EXAMPLE Requirements from a park assist system to a brake system.

NOTE In a distributed development, a technical safety concept can be based on another technical safety concept realized by subsystems.

6.3.2 Further supporting information

The following information can be considered:

— hazard analysis and risk assessment report (see ISO 26262-3:2018, 6.5.1); and

— item definition (see ISO 26262-3:2018, 5.5.1).

6.4 Requirements and recommendations

6.4.1 Specification of the technical safety requirements

6.4.1.1 The technical safety requirements shall be specified in accordance with the functional safety concept and the system architectural design of the item considering the following:

a) the safety-related dependencies and constraints of items, systems and their elements;

b) the external interfaces of the system, if applicable; and

c) the configurability of the system.

NOTE 1 Design constraints can result from: environmental conditions, the installation space, the implementation itself (e.g. available performance, thermal capacity, thermal dissipation), and other functional or non-functional requirements (e.g. security, physical limits of used technology).

NOTE 2 The configurability of systems is determined by variants in the system elements, by configuration data or by calibration data and is often used as part of the strategy to reuse existing systems for different applications.

6.4.1.2 The technical safety requirements shall specify the stimulus response of the system that affects the achievement of safety requirements. This includes the combinations of relevant stimuli and failures with each relevant operating mode and defined system state.

EXAMPLE The Brake System Electronic Control Unit (ECU) disables Adaptive Cruise Control (ACC) braking if a received ACC command message fails error detection code checks.

6.4.1.3 If other functions or requirements are implemented by the system or its elements, in addition to those functions for which technical safety requirements are specified, then these functions or requirements shall be specified or their specification referenced.

EXAMPLE Other requirements can come from Economic Commission for Europe (ECE) rules, Federal Motor Vehicle Safety Standard (FMVSS), company platform strategies, functional concepts or other concepts such as cybersecurity concept.

6.4.1.4 Technical safety and non-safety requirements shall not contradict.

6.4.2 Safety mechanisms

6.4.2.1 The technical safety requirements shall specify the safety mechanisms that detect faults and prevent or mitigate failures present at the output of the system that violate the functional safety requirements (see ISO 26262-3:2018, Clause 7) including:

a) the safety mechanisms related to the detection, indication and control of faults in the system itself;

NOTE 1 This includes the system self-monitoring to detect random hardware faults and, if appropriate, to detect systematic faults.

NOTE 2 This includes safety mechanisms for the detection and control of communication channel failures (e.g. data interfaces, communication buses, wireless radio link).

NOTE 3 Safety mechanisms can be specified with respect to the appropriate level within the system architecture.

b) the safety mechanisms related to the detection, indication and control of faults in other external elements that interact with the system;

EXAMPLE External devices include other electronic control units, power supplies or communication devices.

c) the safety mechanisms that contribute to the system achieving or maintaining the safe state of the item;

NOTE 4 This includes arbitration in the case of multiple control requests from safety mechanisms.

d) the safety mechanisms to define and implement the warning and degradation strategy; and

e) the safety mechanisms that prevent faults from being latent.

NOTE 5 These safety mechanisms are usually related to self-tests that take place during power up (pre-drive checks), as in the case of measures a) to d), during operation, during power-down (post-drive checks), and as part of maintenance.

6.4.2.2 For each safety mechanism that enables an item to achieve or maintain a safe state, the following shall be specified:

a) the transition between states;

NOTE 1 This includes the requirements to control the actuators.

b) the fault handling time interval with respect to the timing requirements apportioned from the appropriate architectural level; and

NOTE 2 This sub-requirement aims to achieve a consistent timing within the boundary of the fault handling time interval) FTTI which is specified for each Safety Goal.

c) the emergency operation tolerance time interval, see ISO 26262-1:2018, 3.45, if the safe state of the item cannot be reached within the FTTI.

NOTE 3 In-vehicle testing and experimentation can be used to determine the emergency operation tolerance time interval.

EXAMPLE 1 Duration of the degraded operation prior to the safe state.

EXAMPLE 2 A safety mechanism for a brake-by-wire application, which depends on the power supply, can include the specification of a secondary power supply or storage device (capacity, time to activate and operate, etc.).

6.4.2.3 This requirement applies to ASILs (A), (B), C, and D. If applicable, safety mechanisms shall be specified to prevent faults from being latent.

NOTE 1 Only random hardware faults which are multiple-point faults have the potential to be latent.

EXAMPLE Self-tests are safety mechanisms which verify the status of components during the different operation modes (e.g. power-up, power-down, during operation or in an additional self-test mode) to detect multiple-point faults. Valve, relay or lamp function tests that take place during power up routines are examples of self-tests.

NOTE 2 Evaluation criteria identifying the need for safety mechanisms preventing faults from being latent are derived in accordance with good engineering practice. The latent fault metric, given in ISO 26262-5:2018, Clause 8, provides evaluation criteria.

6.4.2.4 This requirement applies to ASILs (A), (B), C, and D. To avoid multiple-point failures, the diagnostic test strategy shall be specified for each safety mechanism implemented to detect multiple-point faults, considering:

a) the reliability requirements of the hardware components with consideration given to their role in the architecture and their contribution to a multiple-point failure;

b) the specified quantitative target values for the maximum probability of violation of each safety goal due to random hardware failures (see ISO 26262-5:2018, Clause 9);

c) the assigned ASIL derived from the related safety goal, the related functional safety requirement or technical safety requirement at a higher hierarchical level; and

d) the multiple-point fault detection time interval.

NOTE 1 The diagnostic test strategy can be time driven (e.g. using the diagnostic test time interval) or event driven (e.g. a start-up test).

NOTE 2 A second-order multiple-point failure comprises two faults, separated by the multiple-point fault detection time interval.

NOTE 3 The use of the following measures depends on the time constraints:

— periodic testing of the system or elements during operation;

— self-tests of elements during power-up or power-down; and

— testing the system or elements during maintenance.

6.4.2.5 This requirement applies to ASILs (A), (B), C, and D. The development of safety mechanisms that are implemented only to prevent dual point faults from being latent shall at least comply with:

a) ASIL B for technical safety requirements assigned ASIL D;

b) ASIL A for technical safety requirements assigned ASIL B and ASIL C; and

c) QM for technical safety requirements assigned ASIL A.

NOTE If ASIL decomposition is applied to a requirement, then this clause is applied to the decomposed requirement.

EXAMPLE A memory has a parity as its safety mechanism, with requirements rated ASIL B. The requirement for the self-test that tests the capability of the parity to detect and signal memory faults can be rated ASIL A.

6.4.3 System architectural design specification and technical safety concept

6.4.3.1 The system architectural design in this sub-phase and the technical safety concept shall be based on the item definition, functional safety concept and the prior system architectural design.

6.4.3.2 The consistency of the system architectural design in ISO 26262-3:2018, 7.3.1 and the system architectural design in this sub-phase shall be checked. If discrepancies are identified, an iteration of the activities described in ISO 26262-3:2018 may be necessary.

6.4.3.3 The system architectural design shall implement the technical safety requirements.

6.4.3.4 With regard to the implementation of the technical safety requirements, the following shall be considered in the system architectural design:

a) the ability to verify the system architectural design;

b) the technical capability of the intended hardware and software elements with regard to the achievement of functional safety; and

c) the ability to execute tests during system integration.

6.4.3.5 The internal and external interfaces of safety-related elements shall be defined such that other elements shall not have adverse safety-related effects on the safety-related elements.

6.4.3.6 If ASIL decomposition is applied to the safety requirements during system architectural design, it shall be applied in accordance with ISO 26262-9:2018, Clause 5.

6.4.4 Safety Analyses and avoidance of systematic failures

6.4.4.1 Safety analyses on the system architectural design shall be performed in accordance with Table 1 and ISO 26262-9:2018, Clause 8 in order to:

— provide evidence for the suitability of the system design to provide the specified safety-related functions and properties with respect to the ASIL;

— identify the causes of failures and the effects of faults;

— identify or confirm the safety-related system elements and interfaces; and

— support the design specification and verify the effectiveness of the safety mechanisms based on identified causes of faults and the effects of failures.

Table 1 — System architectural design analysis

Methods		ASIL			
		A	B	C	D
1	Deductive analysis	o	+	++	++
2	Inductive analysis	++	++	++	++

NOTE 1 Safety-related properties include independency and freedom from interference requirements.

NOTE 2 The purpose of these analyses is to assist in the design. Therefore at this stage, qualitative analysis is sufficient. Quantitative analysis can be performed if necessary.

NOTE 3 The analysis is conducted at the level of detail necessary to identify causes and effects of random hardware failures and systematic failures.

NOTE 4 The aim of using a combination of deductive and inductive methods is to provide complementary approaches to analysis, see also ISO 26262-9:2018, 8.2.

6.4.4.2 Identified internal causes of failure shall be eliminated, or their effects mitigated where necessary, to comply with the safety goals or requirements.

6.4.4.3 Identified external causes of failure shall be eliminated, or their effects mitigated where necessary, to comply with the safety goals or requirements.

6.4.4.4 To reduce the likelihood of systematic failures, well-trusted systems design principles should be applied where applicable. These may include the following:

a) re-use of well-trusted technical safety concepts;

b) re-use of well-trusted designs for elements, including hardware and software components;

c) re-use of well-trusted mechanisms for the detection and control of failures; and

d) re-use of well-trusted or standardized interfaces.

6.4.4.5 An analysis of the suitability of well-trusted design principles shall be performed and documented to ensure consistency and suitability to the product's application.

6.4.4.6 In order to avoid systematic faults, the system architectural design shall exhibit the following properties:

a) modularity;

b) adequate level of granularity; and

c) simplicity.

NOTE Aforementioned properties can be achieved by the use of design principles such as hierarchical design, precisely defined interfaces, avoidance of unnecessary complexity of components and interfaces, maintainability, and verifiability.

6.4.4.7 Hazards newly identified during safety analyses or during the system architectural design that are not already covered by a safety goal shall be included in an updated hazard analysis and risk assessment (HARA) in accordance with ISO 26262-3.

NOTE Hazards not already covered by a safety goal may be non-functional hazards. Non-functional hazards are outside the scope of ISO 26262, but they can be annotated in the hazard analysis and risk assessment; e.g. by annotating the hazard with the following statement "No ASIL is assigned to this hazard as it is not within the scope of ISO 26262".

6.4.5 Measures for control of random hardware failures during operation

6.4.5.1 Measures for the detection, control or mitigation of random hardware failures shall be specified with respect to the system architectural design given in 6.4.3.

EXAMPLE 1 Such measures can be hardware diagnostic features and their usage by the software to detect random hardware failures.

EXAMPLE 2 A hardware design having random hardware failures that always result in the safe state being entered without detection (i.e. a fail-safe hardware design).

NOTE A quantitative approximation of the inductive and deductive analyses in 6.4.4.1 is helpful to decide if further safety measures are necessary. A final decision may be necessary after hardware analysis according to ISO 26262-5.

6.4.5.2 This requirement applies to ASILs (B), C, and D of the safety goal. One of the alternative procedures for the evaluation of violation of the safety goal due to random hardware failures (see ISO 26262-5:2018, Clause 9) shall be chosen and the target values shall be specified for final evaluation at the item level.

6.4.5.3 This requirement applies to ASILs (B), C, and D of the safety goal. Appropriate target values for failure rates and diagnostic coverage should be specified at the element level in order to comply with:

a) the target values of the metrics in ISO 26262-5:2018, Clause 8; and

b) the procedures in ISO 26262-5:2018, Clause 9.

6.4.5.4 This requirement applies to ASILs (B), C, and D. For distributed developments (see ISO 26262-8:2018, Clause 5) the derived target values shall be communicated to each relevant party.

NOTE 1 Architectural constraints described in ISO 26262-5:2018, Clauses 8 and 9, are not necessarily applicable to COTS parts and components. This is because suppliers usually cannot foresee the usage of their products in the end-item and the potential safety implications. In such a case, basic data such as failure rate, failure modes, failure rate distribution per failure modes, built-in diagnostics, etc. are made available by the supplier in order to allow the estimation of architectural constraints at overall hardware architecture level.

6.4.6 Allocation to hardware and software

6.4.6.1 The technical safety requirements shall be allocated to the system architectural design elements with system, hardware or software as the implementing technology.

NOTE If the requirements are allocated to system as implementing technology, ISO 26262-4 is used again for further development of these requirements until they can be allocated to hardware and software.

6.4.6.2 The allocation and partitioning decisions shall comply with the system architectural design.

NOTE To achieve independence and to avoid propagation of failures, the system architectural design can implement the partitioning of functions and components.

6.4.6.3 Each system architectural design element shall inherit the highest ASIL from the technical safety requirements that it implements.

6.4.6.4 If a system architectural design element is comprised of sub-elements with different ASILs assigned, or of safety-related and non-safety-related sub-elements, then each of these shall be treated in accordance with the highest ASIL, unless the criteria for coexistence (in accordance with ISO 26262-9:2018, Clause 6) are met.

6.4.6.5 If technical safety requirements are allocated to custom hardware elements that incorporate programmable behaviour (such as ASICs, FPGA or other forms of digital hardware) an adequate development process, combining requirements from ISO 26262-5 and ISO 26262-6, shall be defined and implemented.

NOTE 1 The evidence of compliance with an allocated safety requirement for some of those hardware elements can be provided through evaluation methods in accordance with ISO 26262-8:2018, Clause 13, if the criteria for applying this clause are met.

NOTE 2 Guidance can be found in ISO 26262-11:2018.

6.4.7 Hardware-software interface (HSI) specification

6.4.7.1 The HSI specification shall specify the hardware and software interaction and be consistent with the technical safety concept. The HSI specification shall include the component's hardware parts that are controlled by software and hardware resources that support the execution of the software.

NOTE The aspects and characteristics detailed in the HSI are given in Annex B.

6.4.7.2 The HSI specification shall include the following characteristics:

a) the relevant operating modes of the hardware devices and the relevant configuration parameters;

EXAMPLE 1 Operating modes of hardware devices such as default, initialization, test or advanced modes.

EXAMPLE 2 Configuration parameters such as gain control, band pass frequency or clock pre-scaler.

b) the hardware features that ensure the independence between elements or that support software partitioning;

c) shared and exclusive use of hardware resources;

EXAMPLE 3 Memory mapping, allocation of registers, timers, interrupts, I/O ports.

d) the access mechanism to hardware devices; and

EXAMPLE 4 Serial, parallel, slave, master/slave.

e) the timing constraints derived from the technical safety concept.

6.4.7.3 The relevant diagnostic capabilities of the hardware, and their use by the software, shall be specified in the HSI specification:

a) the hardware diagnostic features shall be defined; and

EXAMPLE Detection of over-current, short-circuit or over-temperature.

b) the diagnostic features concerning the hardware, to be implemented in software, shall be defined.

6.4.7.4 The HSI shall be specified during the system architectural design.

NOTE The HSI is refined during hardware development (see ISO 26262-5:2018, Clause 6) and during software development (see ISO 26262-6:2018, Clause 6).

6.4.8 Production, operation, service and decommissioning

6.4.8.1 The requirements addressed in ISO 26262-7:2018 for production, operation, service and decommissioning, identified during the system architectural design, shall be specified. These include:

a) measures required to achieve, maintain or restore the safety-related functions and properties of the item and its elements during production, service or decommissioning;

b) the safety-related special characteristics;

c) the requirements that ensure proper identification of systems or elements;

d) the verification measures for production;

e) the service requirements including diagnostic data and service notes; and

f) measures for decommissioning.

EXAMPLE Assembly or disassembly instructions, service notes, instructions regarding permitted repair for system elements, decommissioning instructions, labelling of elements.

NOTE There are two main aspects that ensure functional safety during production, operation, service and decommissioning. The first aspect relates to those activities that ensure an adequate system architectural design and the specification of suitable safety-related special characteristics during the development phase, which are given in requirement 6.4.8.1, while the second aspect relates to those activities that ensure the achievement or maintenance of functional safety during the production and operation phase (e.g. based on specified safety-related special characteristics), which are addressed in ISO 26262-7:2018.

6.4.8.2 Diagnostic features shall be specified in order to provide the required data that enables field monitoring for the item or its elements according to ISO 26262-2:2018, Clause 7, with consideration being given to the results of safety analyses and the implemented safety mechanisms.

6.4.8.3 To restore or maintain functional safety, diagnostic features shall be specified that allow fault identification and the effectiveness of maintenance or repair to be checked during servicing.

6.4.9 Verification

6.4.9.1 The technical safety requirements shall be verified in accordance with ISO 26262-8:2018, Clauses 6 and 9, to provide evidence for their correctness, completeness, and consistency with respect to the given boundary conditions of the system.

6.4.9.2 The system architectural design, the hardware-software interface (HSI) specification and the specification of requirements for production, operation, service and decommissioning and the technical safety concept shall be verified using the verification methods listed in Table 2 to provide evidence that the following objectives are achieved:

a) they are suitable and adequate to achieve the required level of functional safety according to the relevant ASIL;

b) there is consistency between the system architectural design and the technical safety concept; and

c) validity of and compliance with system architectural designs of prior development steps.

NOTE Safety anomalies and incompleteness identified will be reported in accordance with ISO 26262-2:2018, 5.4.3.

Table 2 — Verification

Methods		ASIL			
		A	B	C	D
1a	Inspection[a]	+	++	++	++
1b	Walkthrough[a]	++	+	o	o
2a	Simulation[b]	+	+	++	++
2b	System prototyping and vehicle tests[b]	+	+	++	++
3	System architectural design analyses[c]	see Table 1			

[a] Methods 1a and 1b serve as a check of complete and correct implementation of the requirements.

[b] Methods 2a and 2b can be used advantageously as a fault injection test to support the argumentation of completeness and correctness of a system architectural design with respect to faults.

[c] For conducting safety analyses, see ISO 26262-9:2018, Clause 8.

6.5 Work products

6.5.1 Technical safety requirements specification resulting from requirements in 6.4.1 and 6.4.2.

6.5.2 Technical safety concept resulting from requirements in 6.4.3 to 6.4.6.

6.5.3 System architectural design specification resulting from requirements in 6.4.3 to 6.4.6.

6.5.4 Hardware-software interface (HSI) specification resulting from requirements in 6.4.7.

6.5.5 Specification of requirements for production, operation, service and decommissioning resulting from requirements in 6.4.8.

6.5.6 Verification report for system architectural design, the hardware-software interface (HSI) specification, the specification of requirements for production, operation, service and decommissioning, and the technical safety concept resulting from requirements in 6.4.9.

6.5.7 Safety analyses report resulting from requirements in 6.4.4.

7 System and item integration and testing

7.1 Objectives

The integration and testing phase comprises three sub-phases and three objectives as described below. The first sub-phase is the integration of the hardware and software of each element. The second sub-phase is the integration of the elements that comprise a system to form a complete item. The third sub-phase is the integration of the item with other systems within a vehicle. The objectives of this clause are:

a) to define the integration steps and to integrate the system elements until the system is fully integrated;

b) to verify that the defined safety measures, resulting from safety analyses at the system architectural level, are properly implemented; and

c) to provide evidence that the integrated system elements fulfil their safety requirements according to the system architectural design.

7.2 General

The integration of the item's elements is carried out in a systematic way starting from software-hardware integration and verification through system integration and verification to vehicle integration and verification. Specified integration tests are performed at each integration stage to provide evidence that the integrated elements interact correctly.

After sufficient development of hardware and software in accordance with ISO 26262-5 and ISO 26262-6, system integration can be started in accordance with this clause.

7.3 Inputs to this clause

7.3.1 Prerequisites

The following information shall be available:

— safety goals from the hazard analysis and risk assessment report in accordance with ISO 26262-3:2018, 6.5.1;

— functional safety concept in accordance with ISO 26262-3:2018, 7.5.1;

— technical safety concept in accordance with 6.5.2;

— system architectural design specification in accordance with 6.5.3; and

— HSI specification in accordance with 6.5.4, ISO 26262-5:2018, 6.5.2 and ISO 26262-6:2018, 6.5.2.

7.3.2 Further supporting information

The following information can be considered:

— vehicle architecture (from an external source);

— technical safety concepts of other vehicle systems (from an external source); and

— safety analyses report (see 6.5.7).

7.4 Requirements and recommendations

7.4.1 Specification of integration and test strategy

7.4.1.1 To provide evidence that the system architectural design is compliant with the functional safety and technical safety requirements, integration testing activities shall be performed in accordance with ISO 26262-8:2018, Clause 9 to check:

a) the correct implementation of functional safety and technical safety requirements;

b) the correct functional performance, accuracy and timing of safety mechanisms;

c) the consistent and correct implementation of interfaces; and

d) adequate robustness.

7.4.1.2 An integration and test strategy shall be defined that considers the system architectural design specification, the functional safety concept and the technical safety concept. It shall address:

a) the test goals suitable to provide evidence for functional safety; and

b) the integration and testing of the item and its elements that contribute to the safety concepts.

NOTE This includes elements of other technologies that contribute to the safety concepts.

7.4.1.3 To enable the item integration sub-phase, the following shall be performed based on the integration and test strategy:

a) the item integration and test strategy shall be defined for the hardware-software integration and testing;

b) the item integration and test strategy shall be defined to include the specification of integration tests for the system and vehicle-levels. It shall ensure that open issues from hardware-software verifications are addressed;

c) the item integration and test strategy shall consider interfaces between vehicle systems (both internal and external to the item) and the environment; and

d) the item integration and test strategy shall consider if systems or elements are being integrated that were developed as safety element out of context (SEooC) and if the assumptions made during that development need to be verified.

NOTE The specification of the integration and the verification carried out at the hardware-software integration level and the item level considers the interface and the interaction between hardware and software.

7.4.1.4 If the system is configurable (e.g. by variance of elements or calibration data), then the verification at the system or vehicle level shall provide evidence of compliance with safety requirements for the configurations at implementation-level intended for series production.

NOTE Testing a justified subset of configurations may be sufficient.

7.4.1.5 The fulfilment of each functional safety and technical safety requirement shall be verified (if applicable by testing) at least once in the complete integration sub-phase.

NOTE 1 A common practice is to verify a safety requirement at the next higher level of integration to which it has been specified.

NOTE 2 When a SEooC is integrated in a safety-related system, validity of assumptions used for its development is also verified.

NOTE 3 Safety anomalies identified during integration testing are reported in accordance with ISO 26262-2:2018, 5.4.3.

7.4.1.6 To enable the appropriate specification of test cases for the integration tests, test cases shall be derived using an appropriate combination of methods, as listed in Table 3, and by considering the integration level.

Table 3 — Methods for deriving test cases for integration testing

Methods		ASIL			
		A	B	C	D
1a	Analysis of requirements	++	++	++	++
1b	Analysis of external and internal interfaces	+	++	++	++
1c	Generation and analysis of equivalence classes for hardware-software integration	+	+	++	++
1d	Analysis of boundary values	+	+	++	++
1e	Error guessing based on knowledge or experience	+	+	++	++
1f	Analysis of functional dependencies	+	+	++	++
1g	Analysis of common limit conditions, sequences, and sources of dependent failures, see ISO 26262-9:2018, Clause 7	+	+	++	++
1h	Analysis of environmental conditions and operational use cases	+	++	++	++
1i	Analysis of field experience	+	++	++	++

7.4.2 Hardware-software integration and testing

7.4.2.1 Hardware-software integration

7.4.2.1.1 The hardware developed in accordance with ISO 26262-5 and the software developed in accordance with ISO 26262-6 shall be integrated and used as the subject of the test activities in Table 4 to Table 8.

7.4.2.1.2 The integrated hardware and software shall be tested for compliance with the requirements addressing the HSI specification.

NOTE The use of production-intent hardware and software is preferred. Modified hardware or software might be used where necessary for particular test techniques.

7.4.2.2 Test goals and test methods during hardware-software testing

7.4.2.2.1 The test goals resulting from the requirements 7.4.2.2.2 to 7.4.2.2.6 shall be addressed by the application of adequate test methods, as given in the corresponding tables.

NOTE 1 These will support the detection of systematic faults in the system architectural design.

NOTE 2 Depending on the implemented functionality, its complexity or the distributed nature of the system, it may be feasible to perform tests in other integration sub-phases, provided adequate rationale is given.

7.4.2.2.2 Evidence for the correct implementation of the safety-related functions and behaviour according to the technical safety requirements at the hardware-software level shall be provided by using test methods listed in Table 4.

Table 4 — Correct implementation of technical safety requirements at the hardware-software level

Methods		ASIL			
		A	B	C	D
1a	Requirements-based test[a]	++	++	++	++
1b	Fault injection test[b]	+	++	++	++
1c	Back-to-back test[c]	+	+	++	++

[a] A requirements-based test denotes a test against functional and non-functional requirements.

[b] A fault injection test uses special means to introduce faults into the test object during runtime. This can be done within the software via a special test interface or specially prepared hardware. The method is often used to improve the test coverage of the safety requirements, because during normal operation safety mechanisms are not invoked.

[c] A back-to-back test compares the responses of the test object with the responses of a simulation model to the same stimuli, to detect differences between the behaviour of the model and its implementation.

NOTE The differences in the level of effort applied for Method 1b in Table 4 and Table 9 result from the amount of effort needed to conduct fault injection tests at the system level.

7.4.2.2.3 This requirement applies to ASIL (A), B, C, and D. The correct functional performance, accuracy and timing of the safety mechanisms at the hardware-software level shall be demonstrated using test methods listed in Table 5.

Table 5 — Correct functional performance, accuracy and timing of safety mechanisms at the hardware-software level

Methods		ASIL			
		A	B	C	D
1a	Back-to-back test[a]	+	+	++	++
1b	Performance test[b]	+	++	++	++

[a] A back-to-back test compares the responses of the test object with the responses of a simulation model to the same stimuli, to detect differences between the behaviour of the model and its implementation.

[b] A performance test can verify the performance (e.g. task scheduling, timing, power output) in the context of the whole test object, and can verify the ability of the intended control software to run with the hardware.

7.4.2.2.4 This requirement applies to ASIL (A), B, C, and D. Evidence for the consistent and correct implementation of the external and internal interfaces at the hardware-software level shall be provided by using test methods listed in Table 6.

Table 6 — Consistent and correct implementation of external and internal interfaces at the hardware-software level

Methods		ASIL			
		A	B	C	D
1a	Test of external interfaces[a]	+	++	++	++
1b	Test of internal interfaces[a]	+	++	++	++
1c	Interface consistency check[a]	+	++	++	++

[a] Interface tests of the test object include tests of analogue and digital inputs and outputs, boundary tests and equivalence-class tests, to test the compatibility, timings and other specified ratings. Internal interfaces of an ECU can be tested by static tests for the compatibility of software and hardware as well as dynamic tests of Serial Peripheral Interface (SPI) or Integrated Circuit (IC) communications or any other interface between the elements of an ECU.

7.4.2.2.5 This requirement applies to ASIL (A), (B), C, and D. The effectiveness of the hardware fault detection mechanisms at the hardware-software level, with respect to the fault models, shall be demonstrated using test methods listed in Table 7.

NOTE For references to fault models, see ISO 26262-5:2018, Annex D.

Table 7 — Effectiveness of a safety mechanisms at the hardware-software level

Methods		ASIL			
		A	B	C	D
1a	Fault injection test[a]	+	+	++	++
1b	Error guessing test[b]	+	+	++	++

[a] A fault injection test uses special means to introduce faults into the test object during runtime. This can be done within the software via a special test interface or specially prepared hardware. The method is often used to improve the test coverage of the safety requirements, because during normal operation safety mechanisms are not invoked.

[b] An error guessing test uses expert knowledge and data collected through lessons learned to anticipate errors in the test object. Then a set of tests along with adequate test facilities is designed to check for these errors. Error guessing is an effective method given a tester who has previous experience with similar test objects.

7.4.2.2.6 This requirement applies to ASIL (A), (B), (C), and D. The level of robustness of the elements at the hardware-software level shall be demonstrated using test methods listed in Table 8.

Table 8 — Level of robustness at the hardware-software level

Methods		ASIL			
		A	B	C	D
1a	Resource usage test[a]	+	+	+	++
1b	Stress test[b]	+	+	+	++

[a] A resources usage test can be done statically (e.g. by checking for code sizes or analysing the code regarding interrupt usage, in order to verify that worst-case scenarios do not run out of resources), or dynamically by runtime monitoring.

[b] A stress test verifies the test object for correct operation under high operational loads or high demands from the environment. Therefore, tests under high loads on the test object, or with exceptional interface loads, or values (bus loads, electrical shocks, etc.), as well as tests with extreme temperatures, humidity or mechanical shocks, can be applied.

7.4.3 System integration and testing

7.4.3.1 System integration

7.4.3.1.1 The individual elements of the system shall be integrated in accordance with the system architectural design, and tested in accordance with the system integration test specification.

NOTE The tests are intended to provide evidence that each system element interacts correctly, complies with the technical and functional safety requirements, and gives an adequate level of confidence that unintended behaviours, that could violate a safety goal, are absent.

7.4.3.2 Test goals and test methods during system testing

7.4.3.2.1 The test goals resulting from the requirements 7.4.3.2.2 to 7.4.3.2.5 shall be addressed by the application of adequate test methods, as given in the corresponding tables.

NOTE 1 These will support the detection of systematic faults during system integration and testing.

NOTE 2 Depending on the implemented functionality, its complexity, or the distributed nature of the system, it may be feasible to perform tests in other integration sub-phases provided adequate rationale is given.

7.4.3.2.2 Evidence for the correct implementation of functional safety and technical safety requirements at the system level shall be provided by using test methods as listed in Table 9.

Table 9 — Correct implementation of functional safety and technical safety requirements at the system level

Methods		ASIL			
		A	B	C	D
1a	Requirement-based test[a]	++	++	++	++
1b	Fault injection test[b]	+	+	++	++
1c	Back-to-back test[c]	o	+	+	++

[a] A requirements-based test denotes a test against functional and non-functional requirements.

[b] A fault injection test uses special means to introduce faults into the system. This can be done within the system via a special test interface or specially prepared elements or communication devices. The method is often used to improve the test coverage of the safety requirements, because during normal operation safety mechanisms are not invoked.

[c] A back-to-back test compares the responses of the test object with the responses of a simulation model to the same stimuli, to detect differences between the behaviour of the model and its implementation.

7.4.3.2.3 This requirement applies to ASIL (A), (B), (C), and D. The correct functional performance, accuracy, coverage of failure modes at the system level, and timing of the safety mechanisms at the system level shall be demonstrated using test methods listed in Table 10.

Table 10 — Correct functional performance, accuracy and timing of safety mechanisms at the system level

Methods		ASIL			
		A	B	C	D
1a	Back-to-back test[a]	o	+	+	++
1b	Fault injection test[b]	+	+	++	++
1c	Performance test[c]	o	+	+	++
1d	Error guessing test[d]	+	+	++	++
1e	Test derived from field experience[e]	o	+	++	++

[a] A back-to-back test compares the responses of the test object with the responses of a simulation model to the same stimuli, to detect differences between the behaviour of the model and its implementation.

[b] In the context of demonstrating the effectiveness of the safety mechanisms' failure mode coverage at the system level, fault injection method-based test means to introduce faults into the test object during runtime. This can be done within the software via a special test interface or specially prepared hardware. This approach is valid for a limited set of fault models, i.e. the simple ones that can be realistically injected at system level (like reproducing a stuck-at in a component pin). For fault models at semiconductor level (like soft errors or transistor stuck-at), the fault injection method is applied at a more detailed level as described in ISO 26262-11:2018, 4.8.

[c] A performance test can verify the performance (e.g. actuator speed or strength, whole system response times) of the safety mechanisms of the system.

[d] An error guessing test uses expert knowledge and data collected through lessons learned to anticipate errors in the system. Then a set of tests along with adequate test facilities is designed to check for these errors. Error guessing is an effective method given a tester who has previous experience with similar systems.

[e] A test derived from field experience and data gathered from the field

7.4.3.2.4 Evidence for the consistent and correct implementation of the external and internal interfaces at the system level shall be provided by using test methods listed in Table 11.

Table 11 — Consistent and correct implementation of external and internal interfaces at the system level

Methods		ASIL			
		A	B	C	D
1a	Test of external interfaces[a]	+	++	++	++
1b	Test of internal interfaces[a]	+	++	++	++
1c	Interface consistency check[a]	+	+	++	++
1d	Test of interaction/communication[b]	++	++	++	++

[a] An interface test of the system includes tests of analogue and digital inputs and outputs, boundary tests, and equivalence-class tests, to completely test the specified interfaces, compatibility, timings, and other specified characteristics of the system. Internal interfaces of the system can be tested by static tests (e.g. match of plug connectors) as well as by dynamic tests concerning bus communications or any other interface between system elements.

[b] A communication and interaction test includes tests of the communication between the system elements, as well as between the system under test and other vehicle systems during runtime, against the functional and non-functional requirements.

7.4.3.2.5 The level of robustness at the system level shall be demonstrated using test methods listed in Table 12.

Table 12 — Level of robustness at the system level

Methods		ASIL			
		A	B	C	D
1a	Resource usage test[a]	o	+	++	++
1b	Stress test[b]	o	+	++	++
1c	Test for interference resistance and robustness under certain environmental conditions[c]	++	++	++	++

[a] At the system level, resource usage testing is usually performed in dynamic environments (e.g. lab cars or prototypes). Issues to test include power consumption and bus load.

[b] A stress test verifies the correct operation of the system under high operational loads or high demands from the environment. Therefore, tests under high loads on the system, or with extreme user inputs or requests from other systems, as well as tests with extreme temperatures, humidity or mechanical shocks, can be applied.

[c] A test for interference resistance and robustness, under certain environmental conditions, is a special case of stress testing. This includes EMC and ESD tests (e.g. see [4], [5], [6], [7]).

7.4.4 Vehicle integration and testing

7.4.4.1 Vehicle integration

7.4.4.1.1 The item shall be integrated into the vehicle and the vehicle integration tests shall be carried out.

NOTE When planning the vehicle level integration and verification, the correct vehicle behaviour under typical and extreme vehicle conditions and environments can be considered, but with a subset being sufficient (see Table 3).

7.4.4.1.2 The verification of the interface specification of the item with the in-vehicle communication network and the in-vehicle power supply network shall be performed.

7.4.4.2 Test goals and test methods during vehicle testing

7.4.4.2.1 Test goals resulting from the requirements 7.4.4.2.2 to 7.4.4.2.5 shall be addressed by the application of adequate test methods as listed in the corresponding tables.

NOTE 1 These will support the detection of systematic faults during vehicle integration.

NOTE 2 Depending on the implemented functionality, its complexity or the distributed nature of the system, it may be feasible to perform tests in other integration sub-phases provided adequate rationale is given.

7.4.4.2.2 The correct implementation of the functional safety requirements at the vehicle level shall be demonstrated using test methods listed in Table 13.

Table 13 — Correct implementation of the functional safety requirements at the vehicle level

Methods		ASIL			
		A	B	C	D
1a	Requirement-based test[a]	++	++	++	++
1b	Fault injection test[b]	++	++	++	++
1c	Long-term test[c]	++	++	++	++
1d	User test under real-life conditions[c]	++	++	++	++

[a] A requirements-based test denotes a test against functional and non-functional requirements.

[b] A fault injection test uses special means to introduce faults into the item. This can be done within the item via a special test interface or specially prepared elements or communication devices. The method is often used to improve the test coverage of the safety requirements, because during normal operation safety mechanisms are not invoked.

[c] A long-term test and a user test under real-life conditions are similar to tests derived from field experience but use a larger sample size, normal users as testers, and are not bound to prior specified test scenarios, but performed under real-life conditions during everyday life. These tests can have limitations, if necessary, to ensure the safety of the testers, e.g. with additional safety measures or disabled actuators.

7.4.4.2.3 This requirement applies to ASIL (A), (B), C, and D. The correct functional performance, accuracy and timing of the safety mechanisms at the vehicle level shall be demonstrated using test methods listed in Table 14.

Table 14 — Correct functional performance, accuracy and timing of safety mechanisms at the vehicle level

Methods		ASIL			
		A	B	C	D
1a	Performance test[a]	+	+	++	++
1b	Long-term test[b]	+	+	++	++
1c	User test under real-life conditions[b]	+	+	++	++

[a] A performance test can verify the performance (e.g. fault tolerant time intervals on vehicle level and vehicle controllability in the presence of faults) of the safety mechanisms concerning the item.

[b] A long-term test and a user test under real-life conditions are similar to tests derived from field experience but use a larger sample size, normal users as testers, and are not bound to prior specified test scenarios, but performed under real-life conditions during everyday life. These tests can have limitations, if necessary, to ensure the safety of the testers, e.g. with additional safety measures or disabled actuators.

[c] A fault injection test uses special means to introduce faults into the item. This can be done within the item via a special test interface or specially prepared elements or communication devices. The method is often used to improve the test coverage of the safety requirements, because during normal operation safety mechanisms are not invoked.

[d] An error guessing test uses expert knowledge and data collected through lessons learned to anticipate errors in the system. Then a set of tests along with adequate test facilities is designed to check for these errors. Error guessing is an effective method given a tester who has previous experience with similar systems.

[e] A test derived from field experience and data gathered from the field.

Table 14 *(continued)*

Methods		ASIL			
		A	B	C	D
1d	Fault injection test[c]	o	+	++	++
1e	Error guessing test[d]	o	+	++	++
1f	Test derived from field experience[e]	o	+	++	++

[a] A performance test can verify the performance (e.g. fault tolerant time intervals on vehicle level and vehicle controllability in the presence of faults) of the safety mechanisms concerning the item.

[b] A long-term test and a user test under real-life conditions are similar to tests derived from field experience but use a larger sample size, normal users as testers, and are not bound to prior specified test scenarios, but performed under real-life conditions during everyday life. These tests can have limitations, if necessary, to ensure the safety of the testers, e.g. with additional safety measures or disabled actuators.

[c] A fault injection test uses special means to introduce faults into the item. This can be done within the item via a special test interface or specially prepared elements or communication devices. The method is often used to improve the test coverage of the safety requirements, because during normal operation safety mechanisms are not invoked.

[d] An error guessing test uses expert knowledge and data collected through lessons learned to anticipate errors in the system. Then a set of tests along with adequate test facilities is designed to check for these errors. Error guessing is an effective method given a tester who has previous experience with similar systems.

[e] A test derived from field experience and data gathered from the field.

7.4.4.2.4 This requirement applies to ASIL (A), (B), C, and D. The consistency and correctness of the implementation of the interfaces internal and external to the vehicle shall be demonstrated using test methods listed in Table 15.

NOTE Internal interfaces are between items or between systems. External interfaces are between an item and the vehicle environment.

Table 15 — Correct implementation of internal and external interfaces at the vehicle level

Methods		ASIL			
		A	B	C	D
1a	Test of internal interfaces[a]	+	+	++	++
1b	Test of external interfaces[a]	+	+	++	++
1c	Test of interaction/communication[b]	+	+	++	++

[a] An interface test at the vehicle level tests the interfaces of the vehicle systems for compatibility. This can be done statically by validating value ranges, ratings, or geometries as well as dynamically during operation of the whole vehicle.

[b] A communication and interaction test includes tests of the communication between the systems of the vehicle during runtime against functional and non-functional requirements.

7.4.4.2.5 This requirement applies to ASIL (A), (B), C, and D. The level of robustness at the vehicle level shall be demonstrated using test methods listed in Table 16.

Table 16 — Level of robustness at the vehicle level

Methods		ASIL			
		A	B	C	D
1a	Resource usage test[a]	+	+	++	++
1b	Stress test[b]	+	+	++	++
1c	Test for interference resistance and robustness under certain environmental conditions[c]	+	+	++	++
1d	Long-term test[d]	+	+	++	++

[a] At the vehicle level, resource usage testing is usually performed in dynamic environments (e.g. electronic control unit network environments, prototypes or whole vehicles). Issues to test include item internal resources, power consumption, or limited resources of other vehicle systems.

[b] A stress test verifies the correct operation of the vehicle under high operational loads or high demands from the environment. Therefore tests under high loads on the vehicle or with extreme user inputs or requests from other systems as well as tests with extreme temperatures, humidity, or mechanical shocks can be applied.

[c] A test for interference resistance and robustness, under certain environmental conditions, is a special case of stress testing. This includes EMC and ESD tests (e.g. see References [4], [5], [6], [7]).

[d] A long-term test and a user test under real-life conditions are similar to tests derived from field experience but use a larger sample size, normal users as testers, and are not bound to prior specified test scenarios, but performed under real-life conditions during everyday life.

7.5 Work products

7.5.1 Integration and test strategy resulting from requirements in 7.4.1.

7.5.2 Integration and test report resulting from requirements in 7.4.2, 7.4.3 and 7.4.4.

8 Safety validation

8.1 Objectives

The objectives of this clause are:

a) to provide evidence that the safety goals are achieved by the item when being integrated into the respective vehicle(s), and

b) to provide evidence that the functional safety concept and the technical safety concept are appropriate for achieving functional safety for the item.

8.2 General

The purpose of the preceding verification activities (e.g. design verification, safety analyses, hardware, software, and item integration and test) is to provide evidence that the results of each particular activity comply with the specified requirements.

The safety validation of the integrated item in representative vehicle(s) aims to provide evidence of appropriateness for the intended use and aims to confirm the adequacy of the safety measures for a class or set of vehicles. Safety validation provides assurance that the safety goals have been achieved, based on examination and test.

8.3 Inputs to this clause

8.3.1 Prerequisites

The following information shall be available:

— hazard analysis and risk assessment report in accordance with ISO 26262-3:2018, 6.5.1; and

— functional safety concept in accordance with ISO 26262-3:2018, 7.5.1.

8.3.2 Further supporting information

The following information can be considered:

— technical safety concept (see 6.5.2);

— item definition (see ISO 26262-3:2018, 5.5.1); and

— safety analyses report (see 6.5.7).

8.4 Requirements and recommendations

8.4.1 Safety validation environment

8.4.1.1 The safety goals shall be validated for the item in a representative context at vehicle level.

NOTE 1 This integrated item includes, where applicable: system, software, hardware, elements of other technologies, external measures.

NOTE 2 This is especially important for T&B where different base vehicle types could be the subject of a safety validation.

8.4.1.2 For the definition of a representative context, representative vehicles based on vehicle types and vehicle configurations shall be considered.

NOTE A relevant input for the choice of representative vehicles might be the hazard analysis and risk assessment report (see ISO 26262-3:2018, 6.5.1).

8.4.1.3 Safety goals shall be validated giving consideration to variance in operation that impacts the technical characteristics, which have been considered in the hazard analysis and risk assessment.

8.4.2 Specification of safety validation

8.4.2.1 The safety validation specification shall be defined, including:

a) the configuration of the item subjected to safety validation including its calibration data in accordance with ISO 26262-6:2018, Annex C;

NOTE If a complete safety validation of each item configuration is not feasible, then a reasonable subset can be selected.

b) the specification of safety validation procedures, test cases, driving manoeuvres, and acceptance criteria; and

c) the equipment and the required environmental conditions.

8.4.3 Execution of safety validation

8.4.3.1 If testing is used for safety validation, then the same requirements as provided for verification testing (see ISO 26262-8:2018, 9.4.2 and 9.4.3) may be applied.

8.4.3.2 The achievement of functional safety for the item when being integrated into the vehicle shall be validated by evaluating the following aspects:

a) the controllability;

NOTE 1 Controllability can be validated using operating scenarios, including intended use and foreseeable misuse.

NOTE 2 One acceptance criteria for the safety validation might be a sufficient controllability in a safe state defined in ISO 26262-3:2018, 7.4.2.5.

b) the effectiveness of the external measures;

c) the effectiveness of the elements of other technologies; and

d) assumptions that influence the ASIL in the hazard analysis and risk assessment (see ISO 26262-3:2018, 6.4.4.4) that can be checked only in the final vehicle.

EXAMPLE If a mechanical component is assumed to prevent or mitigate a specific hazard potentially caused by a malfunction of an E/E system, the effectiveness of this component to prevent or mitigate that hazard is validated at the vehicle level.

8.4.3.3 The safety validation at the vehicle level, based on the safety goals, the functional safety requirements and the intended use, shall be executed as planned using:

a) the safety validation procedures and test cases for each safety goal including detailed pass/fail criteria; and

b) the scope of application. This may include issues such as configuration, environmental conditions, driving situations, operational use cases, etc.

NOTE Operational use cases can be created to help focus the safety validation at the vehicle level.

8.4.3.4 An appropriate set of the following methods shall be applied:

a) repeatable tests with specified test procedures, test cases, and pass/fail criteria;

EXAMPLE 1 Positive tests of functions and safety requirements, black box testing, simulation, tests under boundary conditions, fault injection, durability tests, stress tests, highly accelerated life testing (HALT), simulation of external influences.

b) analyses;

EXAMPLE 2 FMEA, FTA, ETA, simulation.

c) long-term tests, such as vehicle driving schedules and captured test fleets;

d) operational use cases under real-life conditions, panel or blind tests, or expert panels; and

e) reviews.

8.4.4 Evaluation

8.4.4.1 The results of the safety validation shall be evaluated to provide evidence that the implemented safety goals achieve functional safety for the item.

8.5 Work products

8.5.1 Safety validation specification including safety validation environment description resulting from requirements in 8.4.1 and 8.4.2.

8.5.2 Safety validation report resulting from requirements in 8.4.3 and 8.4.4.