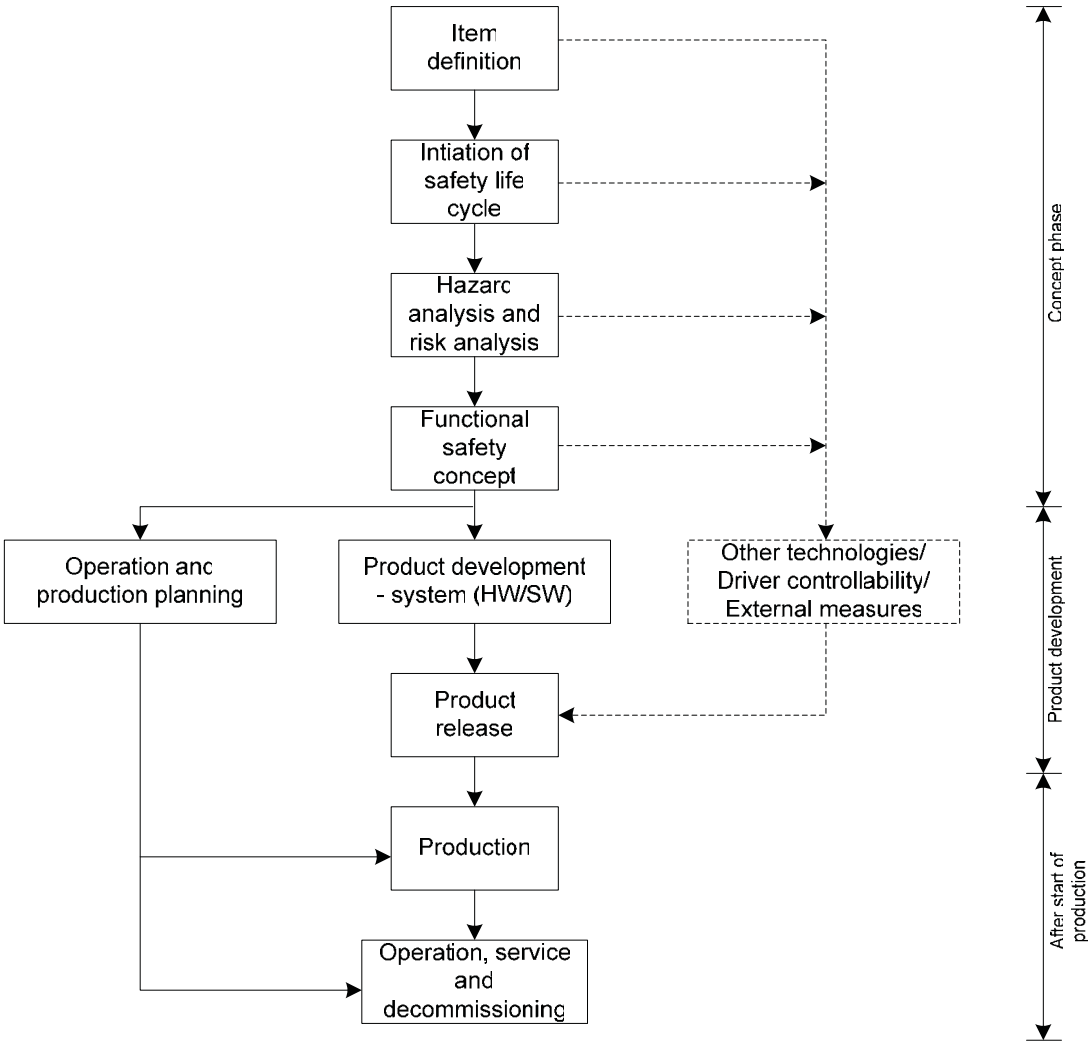
Safety requirements and validation methods for safety-related automotive electronics

Jan Jacobson SP, Andreas Söderberg SP,
Lars-Åke Johansson QRtech, Henrik Lönn Volvo Technology

SP Technical Research Institute of Sweden



SP Electronics
SP REPORT 2007:13

Safety requirements and validation methods for safety-related automotive electronics

Jan Jacobson, SP
Andreas Söderberg, SP
Lars-Åke Johansson, QRtech
Henrik Lönn, Volvo Technology

Abstract

Safety requirements and validation methods for safety-related automotive electronics

Functional safety for automotive embedded systems is a topic of increasing importance. The report describes the overall safety lifecycle, the functional safety requirements and validation and verification for safety related automotive systems. Simplified examples are given to illustrate the development activities.

References are made to standards for functional safety developed by the International Electrotechnical Commission (IEC) and the International Standardisation Organisation (ISO).

The report is based on the work of the AutoVal project of the IVSS (Intelligent Vehcle Safety Systems) research programme.

Key words: automotive electronics, dependable systems, safety, reliability, validation

SP Sveriges Tekniska Forskningsinstitut
SP Technical Research Institute of Sweden

SP Report 2007:13
ISBN 978-91-85533-83-1
ISSN 0284-5172
Borås 2007

Contents

Preface

New safety functions and the increased complexity of vehicle electronics enhance the need to demonstrate dependability. Vehicle manufacturers and suppliers must be able to present a safety argument for the dependability of the product, correct safety requirements and suitable development methodology.

The objective of the AutoVal project is to develop a methodology for safety validation of a safety-related function (or safety-related subsystem) of a vehicle. The validation shall produce results which can be used either as a basis for a whole vehicle type approval driven by legislation, or for supporting dependability claims according to the guidelines of the manufacturer.

The AutoVal project is a part of the IVSS (Intelligent Vehicle Safety Systems) research programme. IVSS aims at systems and smart technologies to reduce fatalities and severe injuries. This can be done by crash avoidance, injury prevention, mitigation and upgrading of handling, stability and crash-worthiness of cars and commercial vehicles enabled by modern IT. Both infrastructure dependent and vehicle autonomous systems are included as are systems for improved safety for unprotected road – users. The core technologies of IVSS are:

- Driver support & human – machine interface (HMI) systems
- Communication platforms – external / internal to the vehicles
- Sensor – rich embedded systems
- Intelligent road infrastructure & telematics
- Crashworthiness, bio-mechanics and design of vehicles for crash-avoidance and injury prevention.
- Dependable systems
- Vehicle dynamic safety systems

Partners of the AutoVal project are Haldex, QRtech, Saab Automobile, SP Technical Research Institute of Sweden and Volvo AB. The following researchers and engineers have participated in the AutoVal project:
Mr Henrik Aidnell, Saab Automobile
Mrs Sabine Alexandersson, Haldex Brake Products
Mr Joacim Bergman, QRtech
Mr Per-Olof Brandt, Volvo
Mr Robert Hammarström, SP
Mr Jan Jacobson, SP (project manager)
Dr Lars-Åke Johansson, QRtech
Dr Henrik Lönn, Volvo
Mr Carl Nalin, Volvo
Mr Anders Nilsson, Haldex Brake Products
Dr Magnus Gäfvert, Haldex Brake Products
Mr Josef Nilsson, SP
Mr Lars Strandén, SP
Mr Jan-Inge Svensson, Volvo
Mr Andreas Söderberg, SP

Summary

Increased functionality, increased complexity and dependability requirements must be combined for automotive electronics. Systematic work according to an overall safety life cycle will be essential for developing systems with adequate functional safety. The life cycle has to address the concept, the risk analysis, the system development, the hardware development and the software development.

This report is based on the work of the AutoVal project of the IVSS (Intelligent Vehicle Safety Systems) research programme. The aim is to show how safety requirements can be specified and how safety can be demonstrated. Practical experience and techniques are prioritised.

References are given to international standards in this report. The standard IEC 61508 is possible to obtain from the International Electrotechnical Commission (www.iec.ch). The working documents of the draft standard ISO 26262 are not yet publicly available. Information on the progress of the standardisation can be given by the International Standardisation organisation (www.iso.ch) or its national branches. Standards are subject to copyright.

Most terms and definitions used in this report correspond to the use in IEC standards. But the reader is advised to check with his application since differences in terminology exist between standards. Definitions in ISO automotive standards may be different.

This report is the first in a series of three reports. The other two reports are “Methods for Verification and Validation of Safety” and “Model Driven Software Verification and Validation”.

1 Automotive embedded systems

1.1 Increased functionality and authority

More functions will be developed for road vehicles. The number of safety critical and safety related automotive embedded systems is large and will grow even larger.

Many of the functions of a modern car or truck are realised using embedded programmable electronic systems. Nearly all the recently developed functions in vehicles would be impossible without software and electronics. Electronic control units (ECUs) are embedded all over the vehicle. Drivers and passengers are expecting the electronic systems to provide at least the same reliability and availability as the mechanical parts of the vehicle. Most drivers are not even aware of which vehicle functions depend on embedded systems.

There are safety-related parts of the vehicle where a failure of control would cause a hazardous situation. An example of such a system is engine control which must not deliver unexpected engine torque. An unexpected low torque could be hazardous e.g. in an overtaking situation. An unexpected high torque might cause sudden acceleration and loss of control of the vehicle. Other examples of safety-related control are door locks, electronic stability control (ESC) and battery management in hybrid vehicles.

Active safety systems require "intelligence" implemented by programmable electronic systems. The correct employment of an airbag system depends on the correct processing of sensor signals. An airbag is expected to blow when needed at a crash, but also expected not to blow when it should not. Any failure will be safety-related.

A large number of active safety systems can be expected in future vehicles. Advanced airbag systems, electronic stability systems, lane departure warning systems, brake assistance systems and adaptive cruise control systems are already commercially available. Future active safety systems may include collision avoidance by forced steering, night vision systems and brake-by-wire systems.

There are also embedded systems that implement functions with minor safety importance. Malfunction of infotainment systems in the vehicle will be a nuisance, but is unlikely to be safety-related. Failure of automatic climate control systems will reduce the comfort for the driver, but will not be safety-related in the short time aspect.

1.2 Increased complexity

Automotive systems are becoming more complex. This complexity has to be mastered.

More computing power and larger memory capacity enhance the capability in an automotive embedded system. Performance in computing systems has become affordable also for the cost-sensitive automotive industry. The increased performance makes it possible to develop more complex systems.

The different embedded systems of a road vehicle are not isolated. The ECUs are connected together in vehicle communication networks, and data is exchanged between them.

The design principle has been to use one ECU for every function in the vehicle, and then make the ECU work together in a federated way. This will no longer be feasible as the number of functions grow continuously. New functions will have to be distributed over existing ECUs. New hardware units cannot be introduced with every new function. This will lead to increasing complexity. The partitions of an ECU are to be sufficiently isolated otherwise the functions will interfere with each other. All the ECUs performing a part of a function have to be available as "members" of the function. This also increases the complexity.

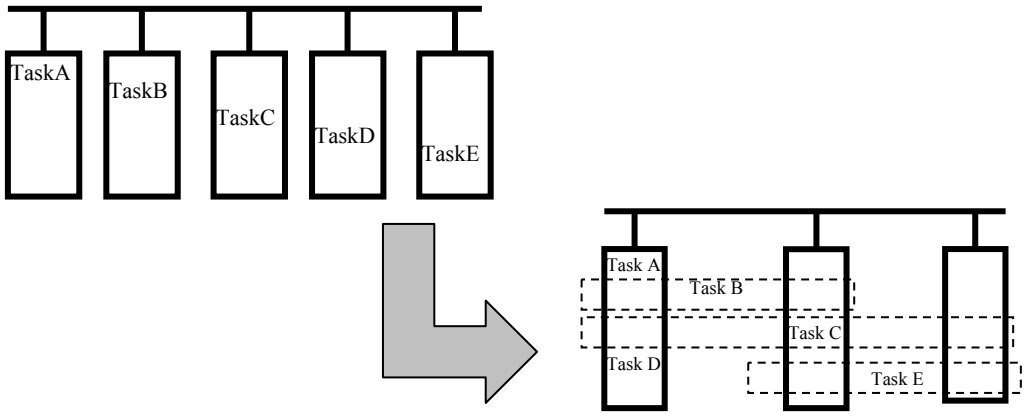


Figure 1. Complexity is increased by distribution of tasks over several nodes.

Future systems for communication vehicle-to-vehicle and vehicle-to-infrastructure will also increase the complexity. Transmission and reception of data to be used in the embedded functions of the vehicle must be handled with care. Open systems may in many situations be hard to combine with dependability.

1.3 Dependability

The safety requirements must be unambiguous, and there must be methods to validate functional safety.

There are many different risks in technical systems, mechanical risks, chemical risks, electrical risks, explosive risks etc. When we consider a system, a device or a machine as safe we mean that all these risks are sufficiently low. Safety means that there are no unacceptable risks for physical damages or injuries on health neither directly nor indirectly as a result of damages on property or environment.

Functional safety is part of the overall safety that depends on a system or equipment operating correctly in response to its inputs. The term "safety validation" in this report means the activities to demonstrate that the needed functional safety has been achieved. Functional safety should not be mistaken for other kinds of safety aspects such as electrical safety or safety in explosive atmospheres. Neither safety nor functional safety can be determined without considering the system as a whole and the environment with which it interacts.

Dependability is a term which summarizes several attributes:
- availability
- reliability
- safety
- confidentiality
- integrity
- maintainability

The concept of "dependable automotive systems" indicates that the system should not only be safe, but also cover the other dependability aspects.

It is not trivial to show that a complex system actually is suitable for its intended use. Careful risk analysis must be made already from the beginning of the development. Every step of the realisation must be confirmed by verification. At the end of the development, there shall be an overall safety validation to demonstrate functional safety.

This report addresses validation of dependability and functional safety. Other safety aspects such as environmental stress, electromagnetic compatibility and electrical safety are not in focus. Additional validation methods should be applied to cover also those aspects.

1.4 Safety-related functions

Development of automotive embedded systems requires handling of safety as an integrated part of the control functions.

Traditional safety functions are designed with the dedicated purpose to reduce a risk in the system. A high-pressure relief valve of a boiler in process control is a good example of a safety function. The valve has the task to open if the pressure of the boiler exceeds the limit and the risk of an explosion is eminent. Controls in the process industry often combine sensors, programmable electronic systems and actuators to build safety functions. Temperature alarms, level gauging and flow control are examples where the safety function is easily distinguished from the basic process control system. (See figure 2.) This is often not the case in automotive control.

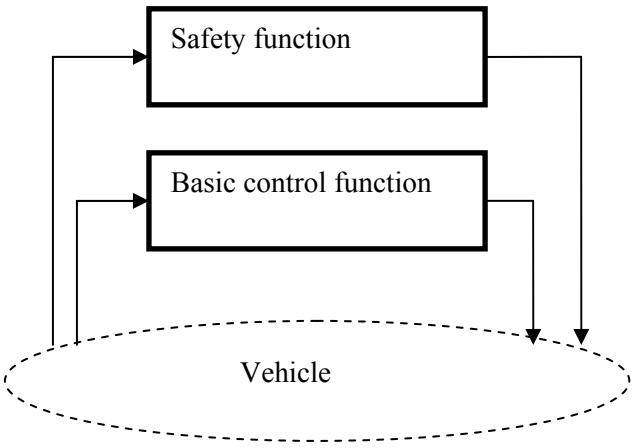


Figure 2. The safety function is well separated from the basic control functions

Many of the functions of a road vehicle are of minor importance to the safety of the driver and the passengers. Such “basic control functions” can be exemplified by the continuous charging of the battery, the engine temperature control and the illumination of the instruments on the dashboard. But it is possible to imagine situations when also these functions may affect safety.

What makes automotive electronics special is that almost every function of the car is to some extent safety-related. The switching on of the headlights is not of primary importance to safety. But what happens if both headlights are unintentionally switched off during night driving? The shifting of gear in reverse is uncritical in most driving situations, but will be safety-related when you have to back out of a dangerous area. Few functions are safety functions in the original meaning, i.e. functions specifically designed to handle a hazardous situation. Seat belt pretensioners and airbags are examples of safety functions.

The advanced driver assistance systems have the safety-related functions as integrated parts of the system. An example is the electronic stability control (ESC) which operates the individual brakes of the wheels to improve stability. Incorrect operation of the brakes is safety-critical. (See figure 3.)

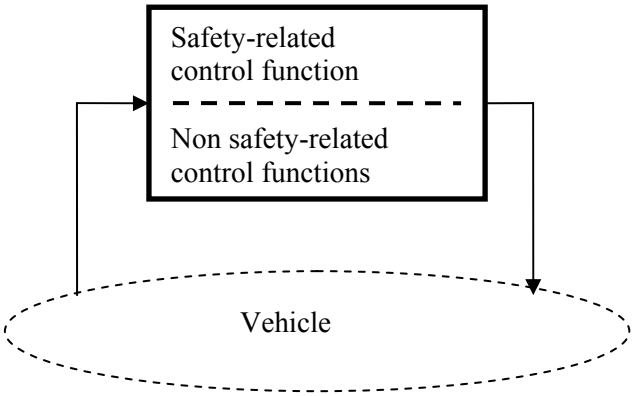


Figure 3. The safety aspects are integrated in the control function

2 Frameworks for functional safety

2.1 Standard IEC 61508

The standard IEC 61508 [IEC 61508] covers those aspects to be considered when electrical/electronic/programmable electronic systems (E/E/PESs) are used to carry out safety functions. It is comprehensive and aimed for use in many different sectors of industry. The objective is that it shall suit all contexts where electrical, electronic or programmable electronic systems are included. It is called a basic safety publication by the International Electrotechnical Commission (IEC) since IEC 61508 is also intended to be the base for sector specific standards. A sector specific standard is intended for a certain domain, for example the automotive, and can therefore be less comprehensive.

The standard is divided into seven parts:
Part 1: General requirements
Part 2: Requirements for E/E/PES safety-related systems
Part 3: Software requirements
Part 4: Definitions and abbreviations
Part 5: Examples of methods for the determination of safety integrity levels
Part 6: Guidelines on the application of part 2 & 3
Part 7: Overview of techniques and measures

The integrity of the safety function is described by four safety integrity levels; SIL 1, 2, 3 and 4. (See figure 4.) SIL 4 provides the most reliable risk reduction. All safety-related functions are expected to be assigned a SIL. The necessary risk reduction is decided depending on the severity of an accident, the exposure to the risk, the possibility of the driver to avoid the situation and the tolerable remaining risk.

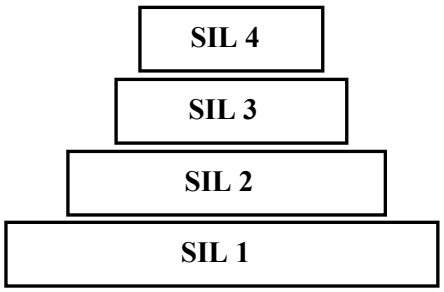


Figure 4. Safety Integrity Levels SIL

The international standard IEC 61508 is accepted as a European standard. It is then called EN 61508. The standard has also been accepted by national standardisation organisations (e.g. as DS-EN 61508 in Denmark). The content of the standard is the same in all three cases. The difference in denomination only shows that it has been accepted by different standardisation organisations.

Additional information on functional safety is given at the IEC web site. [IEC]

2.2 Draft standard ISO 26262

The International Standardisation Organisation (ISO) has decided to develop a standard for functional safety of road vehicles. Some aspects of the IEC 61508 basic safety publication are not regarded to be suitable for the automotive industry. The automotive standard ISO 26262 [ISO26262] is intended to support and facilitate the development of safe products in the automotive industry.

An overall safety lifecycle is specified in the standard. It describes phases such as system definition (concept), hazard analysis and risk assessment, safety requirements, design, realization and safety validation. The work for functional safety shall be carried out all through the life cycle. Activities for hardware verification, software verification, safety validation and functional safety assessment are specified.

The integrity of the safety function is described by four safety integrity levels; ASIL A, B, C and D. (See figure 5.) ASIL D provides the highest risk reduction. All safety-related functions are expected to be assigned an ASIL. The necessary risk reduction is decided depending on the severity of an accident, the exposure to the risk, the possibility of the driver to control the situation and the tolerable remaining risk. ASIL is not intended as a probabilistic target value of the item. This is a difference between ASIL of ISO 26262 and SIL of IEC 61508.

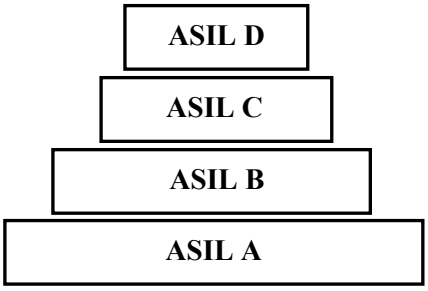


Figure 5. Safety Integrity Levels ASIL

Verification and validation activities are listed in the standard and may be used when considering a performance testing programme.

The standard is drafted in eight parts:
Part 1: Glossary
Part 2: Management of functional safety
Part 3: Concept phase
Part 4: Product development at system level
Part 5: Product development - Hardware
Part 6: Product development - Software
Part 7: Production and operation
Part 8: Supporting processes

The ISO 26262 standard will be an automotive derivate of the IEC 61508 standard, but is intended to be an independent publication without normative references to IEC 61508. The ISO work is in progress and no working documents are publicly available outside the standardisation organisations. A draft international standard is planned to be circulated in 2008.

2.3 MISRA Guidelines for Safety Analysis

The MISRA Safety Analysis Guidelines [MISRA] are the first attempt for defining an automotive-specific approach to safety analysis. It is a complement to the 1994 MISRA Development Guidelines for Vehicle Based Software. The Safety Analysis Guidelines are an interpretation of the IEC 61508 for automotive domain and cover the entire development life-cycle from early concept to final development. The MISRA Safety Analysis Guidelines have been released for external review, but the current version is not final and public at this moment.

The guidelines identify two phases, Preliminary Safety Snalysis (PSA) and Detailed Safety Analysis (DSA) and propose to use a safety case to collect evidence for the safety of the analyzed system. Focus in the current version is on the PSA, and the following activities are suggested for this phase: System Modeling, Hazard Identification, Hazard Classification, Risk Analysis and Safety Requirements Allocation. The Detailed Safety Analysis is carried out during the implementation of the system, and techniques such as FMEA and FTA are suggested. Figure 6 shows the overall steps in the suggested approach and the required data.

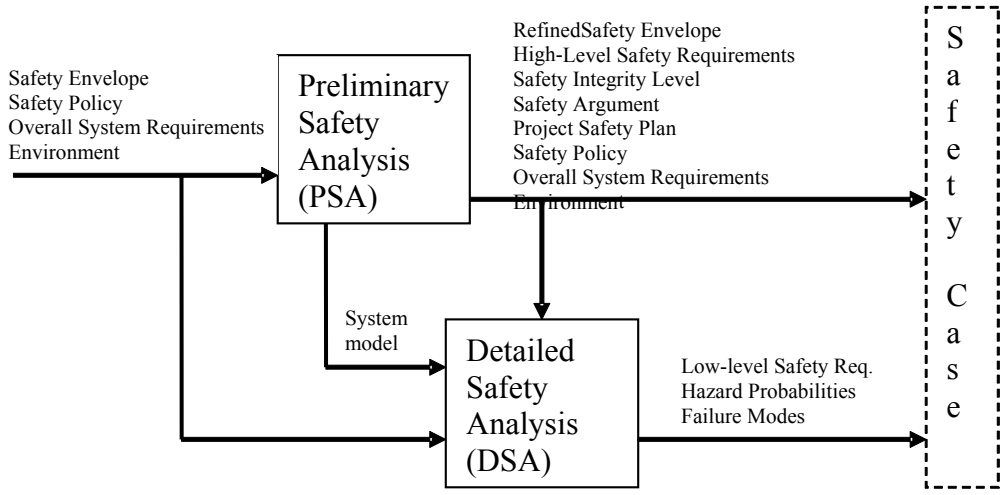


Figure 6. The MISRA Safety Analysis Guidelines: Information flow

Information on the progress of the development of the guidelines will be given at the website of MISRA. [MISRAweb]

2.4 Approval of vehicle control systems

2.4.1 Complex Electronic Vehicle Control Systems

Type approval of electronic braking is regulated in international agreement [ECE-R13]. Also electronic steering is regulated in the corresponding safety regulations. [ECE-R79] Type approvals are also given to anti-theft systems. There are several new systems of a vehicle which will influence the control of the vehicle: autonomous cruise control, enhanced stability control, lane keeping assistance, autonomous park assistance etc.

A discussion is ongoing at the World Forum for Harmonization of Vehicle Regulations concerning approval of a complex electronic control system [GRRF/23003/27]. No decisions have been taken yet, but the proposed regulation demonstrates a way of thinking which may be used as input when considering validation. The definitions may be considered as an established vocabulary and may be used in the vocabulary.

A Complex Electronic Vehicle Control System (CEVCS) is an electronic control system which is subject to a hierarchy of control in which a controlled function may be over-ridden by a higher level electronic control system/function.

An Electrical Sub-Assembly (ESA) is an electronic device or set of devices, intended to be part of a vehicle together with any associate connections and interactions, which performs one or more specialised functions.

Three different alternative procedures are suggested for type approval. One alternative is to type approve the vehicle directly. If this procedure is chosen by a vehicle manufacturer, no separate testing of CEVCSs as ESAs is required.

Another alternative for type approval is to test individual ESAs. The approval for the vehicle is obtained by demonstrating that all the relevant CEVCSs have been approved. The CEVCSs must also be installed in accordance with the instructions of the manufacturer.

The third procedure for type approval is to type approve an ESA to be fitted either to any vehicle type or to a specific vehicle type requested by the manufacturer. ESAs involved in the direct control of vehicles will normally receive type approval by agreement with the vehicle manufacturer.

2.4.2 Documentation required

It was suggested that the manufacturer should provide a documentation package to describe the basic design of the system and the means by which it is linked to other vehicle systems or by which it directly controls output variables. The Safety Concept and the functions of the system shall then be explained. The safety concept is a description of the measures designed into the system in order to address system integrity and thereby ensure safe operation even in the event of an electrical failure.

The description of functions of the system shall
- list all inputs and sensed variables (including definitions of their working range)
- list all output variables which are controlled by the system (including definitions of the range of control exercised)
- include limits defining the boundaries of functional operation (i.e. the boundaries of the external physical limits within which the system is able to maintain control)

System layout and schematics shall include
- inventory of all units of the system
- description of the functions of the units
- interconnections
- signal flow and priorities
- identification of units

The safety concept shall be documented by
- a statement by the manufacturer that the safety objectives will not, under fault conditions, prejudice the safe operation
- the software architecture, the design methods and tools
- the design provisions to generate safe operation under fault conditions
- description of warning signals
- an analysis which shows how the system will behave on the occurrence of faults

2.4.3 Verification and test

The type approval testing is expected to cover both the functional tests and the verification of the safety concept. The function of the system shall be verified under non-fault conditions. It should be tested against the manufacturer's benchmark specification. Verification of the safety concept is verified by checking the reaction of the system under the influence of a failure. A failure of any individual unit is simulated by applying corresponding output signals to electrical units or mechanical elements in order to simulate the effects of internal faults.

3 Overall Safety Lifecycle

The work to create good functional safety in automotive systems is a continuous process starting from the concept phase, all through the development activities and continuing after start of production. Correction of previous mistakes will be time consuming and expensive. It is important to design for safety from the beginning.

The safety life cycle starts in the concept phase when the first ideas of a new control system or a new component start to form. The work with risks continues with the risk analysis, the functional safety requirements and the technical safety concept before the detailed design starts.

The realisation phase is the largest part of the work with a new control system. Parallel to the design of electronics and software the safety life cycle indicates that it is possible to work with safety based on other technologies (for example mechanical protections) and external risk reduction measures in order to reduce the risks.

IEC 61508 defines 16 different phases of a safety life cycle for safety functions (see figure 7):

1 Concept
2 Overall scope definition
3 Hazard and risk analysis
4 Overall safety requirements
5 Safety requirements allocation
6 Overall operation & maintenance planning
7 Overall safety validation planning
8 Overall installation & commissioning planning
9 Realisation (electrical/electronic/programmable electronic control systems)
10 Realisation (other technology)
11 Realisation (external risk reduction)
12 Overall installation & commissioning
13 Overall safety validation
14 Overall operation, maintenance & repair
15 Overall modification & retrofit
16 Decommissioning or disposal

Management responsibilities for functional safety or technical activities are not specified as phases of the lifecycle, but have nevertheless to be covered. Safety policies, safety strategies and responsibilities of departments, persons and organisations are important for the safety management.

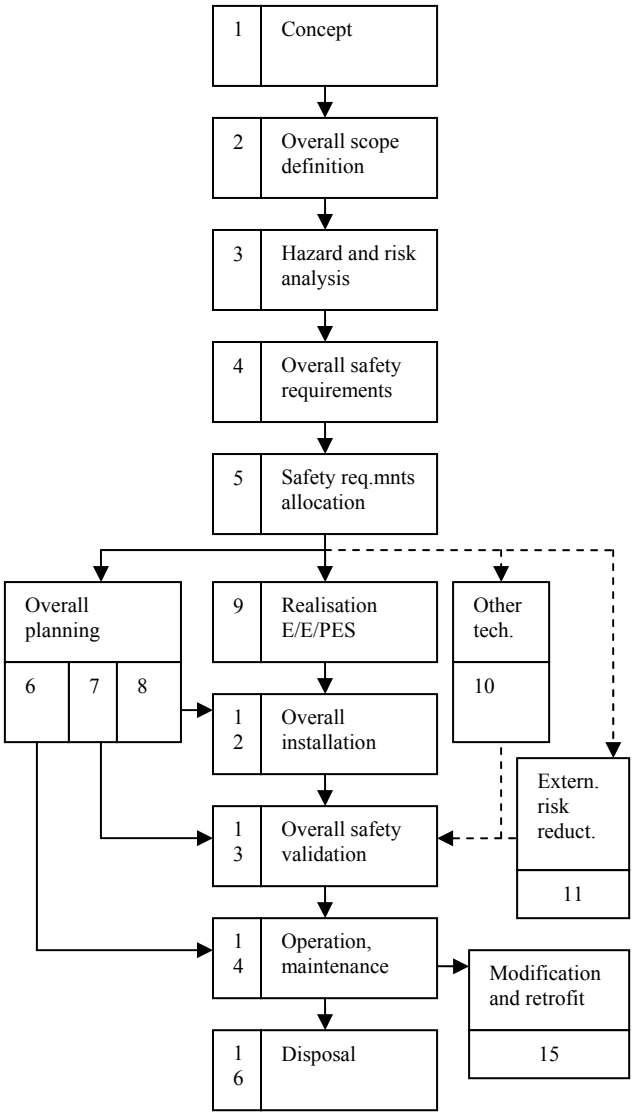


Figure 7. The overall safety lifecycle [IEC61508]

The overall safety lifecycle introduced by the IEC 61508 standard is not regarded as optimal by the automotive industry. The draft standard ISO 26262 introduces an overall safety life cycle intended for automotive electronics. (See figure 8.) Automotive components are developed and then put into serial production according to the lifecycle model.

ISO 26262 defines eleven different activities spread over the concept phase, the product development and after start of production (SOP):

- Item definition
- Initiation of the safety lifecycle
- Hazard analysis and risk assessment
- Functional safety concept
- Product development at system level
- Product development – Hardware
- Product development – Software
- Operation planning
- Production planning
- Production
- Operation, service and decommissioning

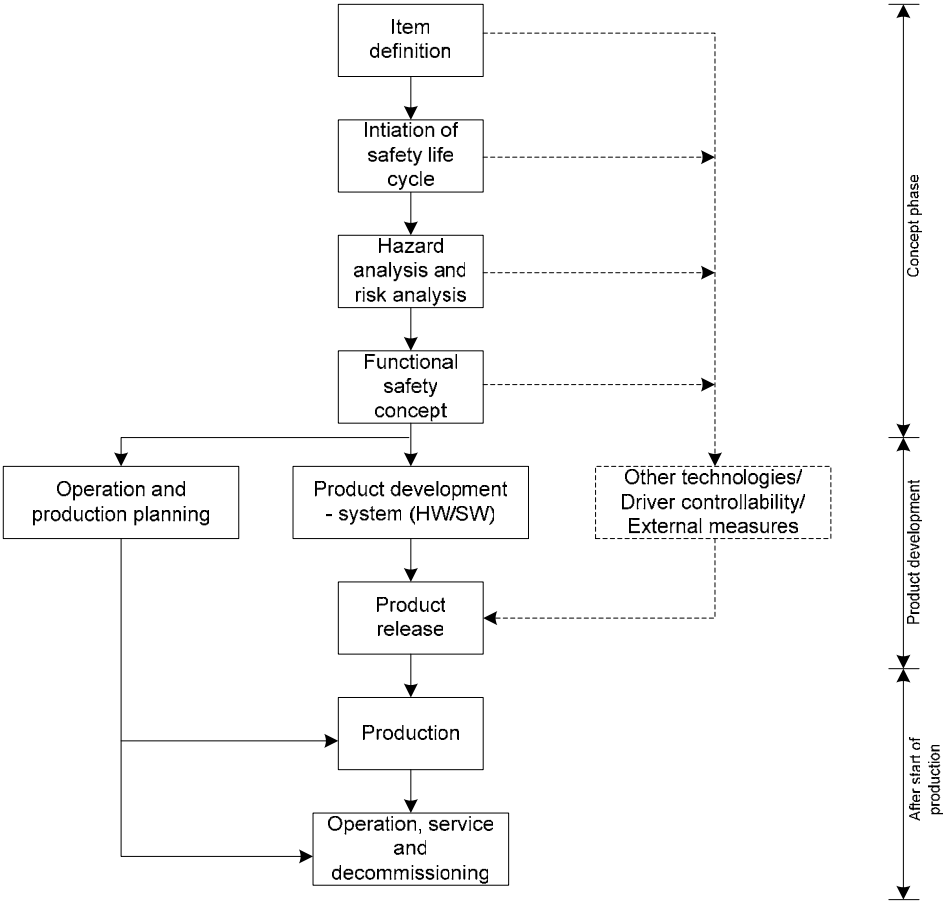


Figure 8. The expected safety lifecycle for automotive electronics [ISO26262]

The overall life cycle phases can also be divided into parts as shown in figure 9.

This report focuses on seven phases of the overall safety lifecycle. (See figure 10.) A preliminary safety analysis comprises the item definition, the hazard and risk analysis and the functional safety concept. A detailed safety analysis is described as development at system level, development at hardware level, development at software level and safety validation. A complete overall safety lifecycle must be supplemented with several phases as described in the standards IEC 61508 and ISO 26262.

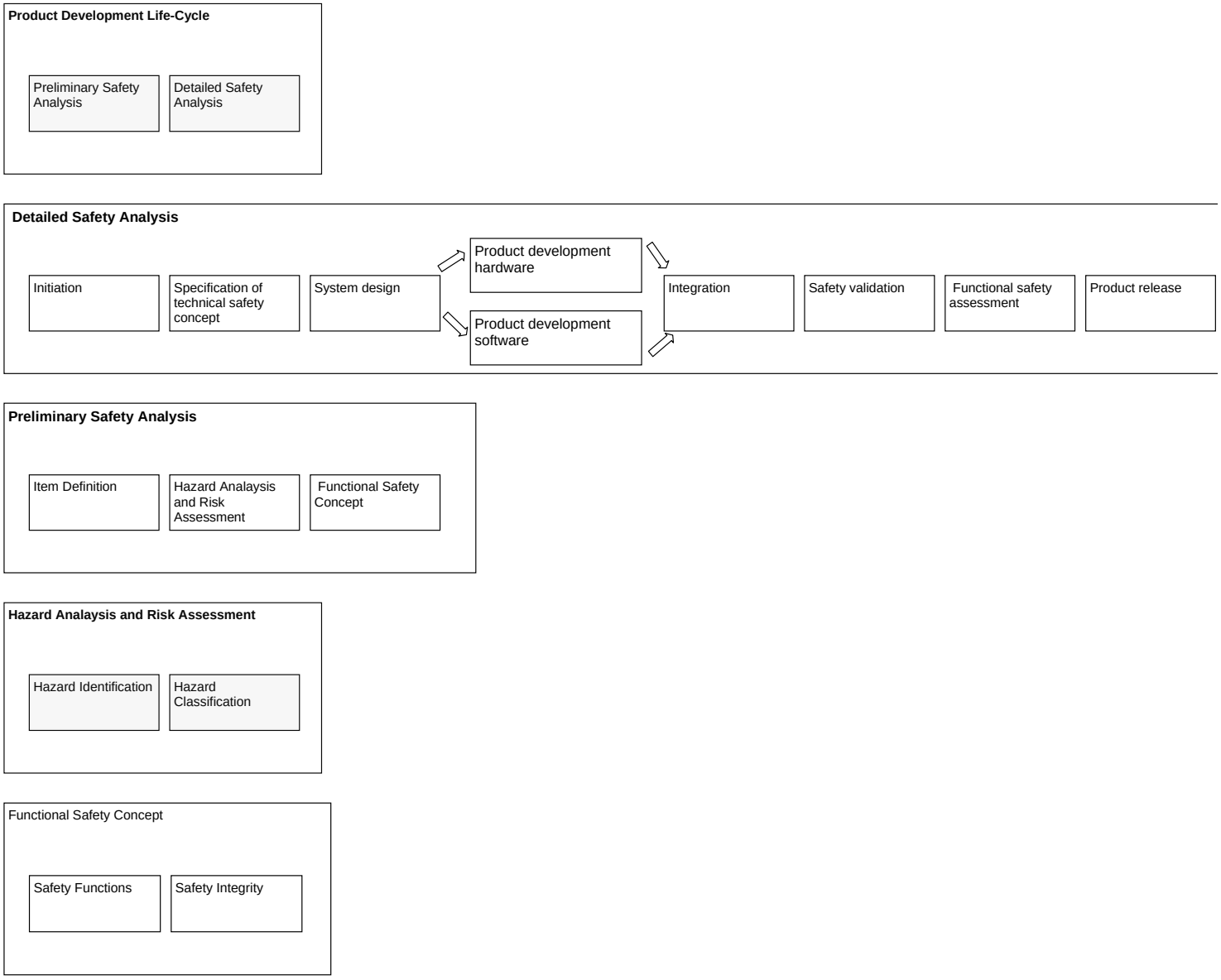


Figure 9. Divisions of the phases of the overall safety lifecycle

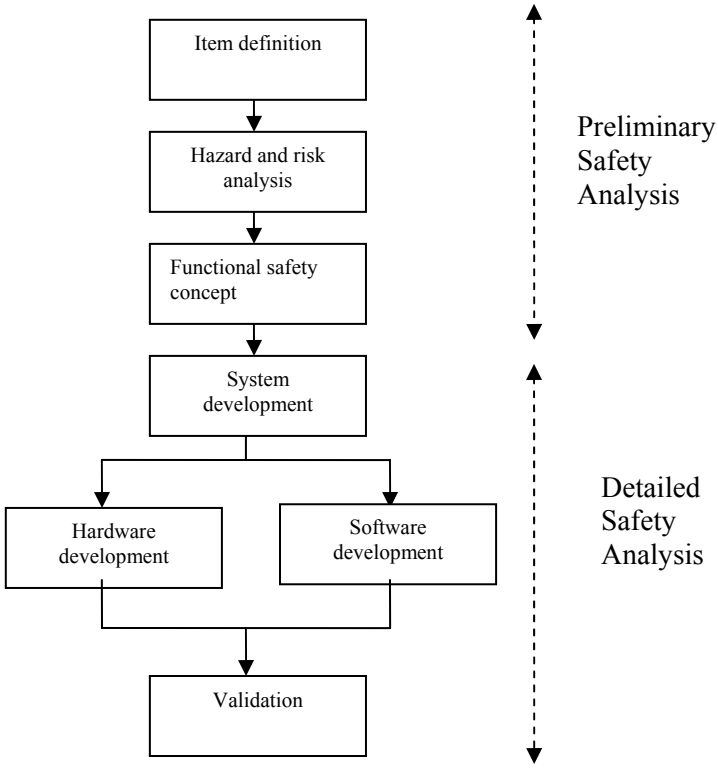


Figure 10. The primary phases used in the AutoVal report

3.1 Support in standards

Descriptions of the overall safety lifecycle are given in the standards IEC 61508-1, ISO 26262-2

4 Preliminary Safety Analysis

The concept phase of the overall safety lifecycle can be seen as a preliminary hazard analysis. This section summarises the activities as item definition, hazard and risk analysis and the development of a functional safety concept.

4.1 Item definition

The objective of this phase is to define the item and to develop an adequate understanding of it. Safety analyses are carried out on the basis of an item description, and the safety goals are derived from it.

The purpose of item definition is to collect and produce sufficient material about the analysis object (item) to adequately define and understand it. The input is existing documentation and information that is relevant for the item. The output should cover all relevant aspects of item, and be sufficiently detailed to perform further design and safety analysis. Examples include:

- Item objectives
  The purpose of item in its current context
- Item Realization
  A representation of the item that corresponds to a preliminary or actual realization. Important aspects are structure and behaviour of item and the external interface.
- Environment of the item
  A description of surrounding systems, assumptions regarding physical environment, vehicle dynamics relevant for item
- Requirements
  Functional and non-functional requirements including known safety requirements
- Safety constraints
  Safety policy of the company, expected use of the vehicle relevant for item, known hazards.

In general, the information should correspond to known and fixed properties of the Item. Implementation decisions that can be revoked should be avoided, while known constraints and decisions should not be hidden.

In the MISRA Safety Guidelines [MISRA], the item definition corresponds to the System Modeling activity.

4.1.1 Support in standards

Descriptions of the item definition are given in the standards
IEC 61508-1 clauses 7.2 and 7.3, ISO 26262-3

4.2 Hazard and risk analysis

The concept of risk is mainly based on the severity of an event and on the probability of the occurrence. Hazard analysis can be made both quantitatively and qualitatively. Severity, probability of occurrence, probability of avoidance etc. have to be judged either by calculating numbers or by qualitative statements (e.g. "high", "medium", "low").

The object to be considered in a hazard and risk analysis is here called the equipment under control (EUC). In this report the EUC often means the complete vehicle.

The objective of the hazard and risk analysis is
- to determine the hazards and hazardous events of the EUC and the EUC control system (in all modes of operation), for all reasonably foreseeable circumstances including fault conditions and misuse;
- to determine the event sequences leading to the hazardous events determined;
- to determine the EUC risks associated with the hazardous events determined.
[IEC 61508, clause 7.4.1]

A hazardous event is an event that causes a person to be exposed to the hazard which results in harm.

4.2.1 Hazard identification

The most common and traditional method of performing hazard identification is to start with defining all different operational situations that may be foreseen with the target object (EUC). With regard to each defined situation potential sources of harm (i.e. hazards) are identified. The main purpose of the hazard identification is to point out those “dangers” that people have to be protected from.

Example: The design of a headlight control of a car. One obvious hazard that has to be considered for this control is the high speed (kinetic energy) of the vehicle. Even though the headlight control is of minor importance during daylight driving, at low speed and when the car is parked the kinetic energy is vital during night driving and high speed is therefore an important hazard. Another hazard for this control may be fire due to e.g. heat from the cablings to the headlight lamps in case of a short circuit.

4.2.2 Risk assessment

The risk related to a hazard is a measure of how efficiently people are protected from that hazard with or without any protective measures. The risk is defined as the probability of occurrence of harm and the consequence of that harm. This can be expressed as:

$R = f \times C$

where f is the frequency of the hazardous event and C is the consequence of the hazardous event.

The risk cannot be directly measured and is the result of a judgement based on experience of the hazards and the hazardous situations related to the target EUC. The methods for evaluating the risk are usually systematic and qualitative and based on the expression $R = f \times C$. The procedure usually is to consider the EUC (e.g. vehicle dynamic function) disregarding any protective (safety) measures and then divide the consequence parameter and the probability of exposure parameter into more fine grained parameters. Examples of

such parameters are described in the following three sections of this report. Different standards for functional safety may use different parameters.

The refined risk associated with an automotive control function may then be described as

R = f x C = [(Exposure) x (Controllability)] x (Severity)

Another method to assess the risk is to calculate the quantitative risk measure using e.g. event trees. If this method is used the above described parameters are not relevant.

The result of a risk assessment may be qualitative or quantitative and is used for determining the amount of protective measures that have to be added in order to reduce the risk sufficiently.

4.2.2.1 Severity

This parameter denotes the consequence of a hazardous event. The more injury the hazardous event causes a person, the higher severity level should be selected. The severity is usually classified using the following parameters or similar:
S0 – No injuries
S1 – Minor injuries
S2 – Severe and non recoverable injuries
S3 – Fatal injuries

4.2.2.2 Exposure

This parameter denotes how often a person is situated in a hazardous situation and is usually expressed as a frequency. This measure mitigates the total risk, even though the severity may be very high for a hazardous event the risk becomes fairly low if it is very improbable that any person is present in the hazardous situation. For example: If the risk of failures related to the steering wheel is considered it is more likely that a driver becomes exposed (and thus injured) to the hazardous event if the failure occurs when the car has a higher speed than if the speed is lower.

The parameter “exposure” may be qualitatively estimated by the probability or by the frequency of exposure:

The probability of exposure to the hazardous situation can be classified as:
E1 - Not very probable
E2 - Probable
E3 - Very probable
E4 - Likely

The frequency of exposure to the hazardous situation can be classified as:
F1 - Rare to often exposure to the hazardous situation
F2 - Often to permanent exposure to the hazardous situation

Depending on what type of equipment is under consideration one of the above parameters will be more applicable than the other.

4.2.2.3 Controllability

MISRA argues that the hazard classification suggested in the IEC 61508 standard is unsuitable for automotive applications [MISRAcontr]. MISRA suggests the concept of "controllability" to describe the ability of the driver, another vehicle occupant, or another person interacting with the system to control the safety of the situation following a failure. (See figure 11.) "Controllability" is recommended to be used for in-vehicle systems, roadside systems and integrated traffic control systems.

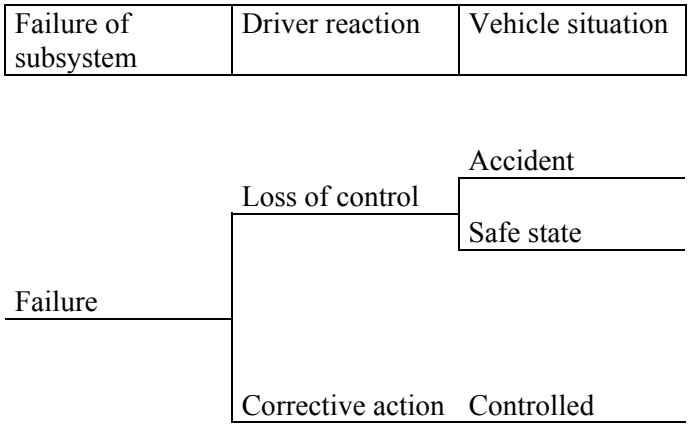


Figure 11. Event tree describing how a failure may result in an accident

Hazards are classified by allocating them to one of five controllability categories; uncontrollable, difficult to control, debilitating, distracting and nuisance only.

Table 1. Definition of the controllability categories [MISRAcontr]

Controllability category	**Definition**
Uncontrollable	This relates to failures whose effects are not controllable by the road user, or vehicle occupants, and which are most likely to lead to extremely severe outcomes. The outcome cannot be influenced by a human response.
Difficult to control	This relates to failures whose effects are not controllable by the road user, or vehicle occupants but could, under favourable circumstances, be influenced by a mature human response. They are likely to lead to very severe outcomes.
Debilitating	This relates to failures whose effects are usually controllable by a sensible human response and, whilst there is a reduction in the safety margin, can usually be expected to lead to outcomes which are at worst severe.
Distracting	This relates to failures which produce operational limitations, but a normal human response will limit the outcome to no worse than minor.
Nuisance only	This relates to failures where safety is not normally considered to be affected, and where customer satisfaction is the main consideration.

Four parameters are considered when a controllability category is assigned to a hazard:
a/ Level of system inter-dependency
b/ Loss of authority or control due to the hazard
c/ Provision of backup or mitigation
d/ Reaction time

The level of system inter-dependency describes how much other systems are relying on the correct functioning of the subsystem. Full functional interdependency means that other systems are operating on data provided by the faulty system. Autonomous system means that no inter-dependency exists.

The loss of authority or control due to the hazard relates to the faulty system. The effect ranges from fully lost to no effect on authority/control.

The provision of backup or mitigation relates to other functions outside the faulty system. Full redundancy or diversity means that other functions are available outside the boundaries of the faulty system. But if no other functions are available there is no mitigation or backup.

The reaction time is the speed with which the driver must be able to recognize that a change has occurred, work out what can be done, and apply corrective actions. In worst case, the reaction time required will be much faster than humanly possible.

The controllability may be classified as:
C1 – Simply controllable
C2 – Normally controllable
C3 – Difficult to control or uncontrollable

4.2.3 Safety integrity level

When starting to talk about the quality or the robustness of a safety function it is soon realised that the requirements placed on each different safety function will vary. High risks within, for example, the aircraft industry raise high demands on the safety function. Reasonable risks within, for example, household devices raise reasonable demands on the safety function. There is a need of being able to grade the quality of the safety functions.

The standard IEC 61508 defines SIL 1 up to SIL 4, where SIL 4 corresponds to the most severe demands. A safety function that fulfils SIL 4 has a very low probability of not working correctly and is developed with very great care. In situations with lower risks it is acceptable to choose a safety function of lower SIL i.e. that is more economic to use.

Therefore, the same device, machine or vehicle can have safety functions with different SIL demands. If all safety functions are controlled by the same control system the highest SIL requirement will be the guiding one, i.e. the control system must be designed for the highest SIL. In certain cases, however, it can be good economy not to design the safety functions better than they need to be.

The specification of SIL can be based on the assessment of risk parameters [IEC 61508]:
– consequence of the hazardous event (C);
– frequency of, and exposure time in, the hazardous zone (F);
– possibility of failing to avoid the hazardous event (P);
– probability of the unwanted occurrence (W).

A risk graph can be used as a qualitative way to specify SIL. (See figure 12.)

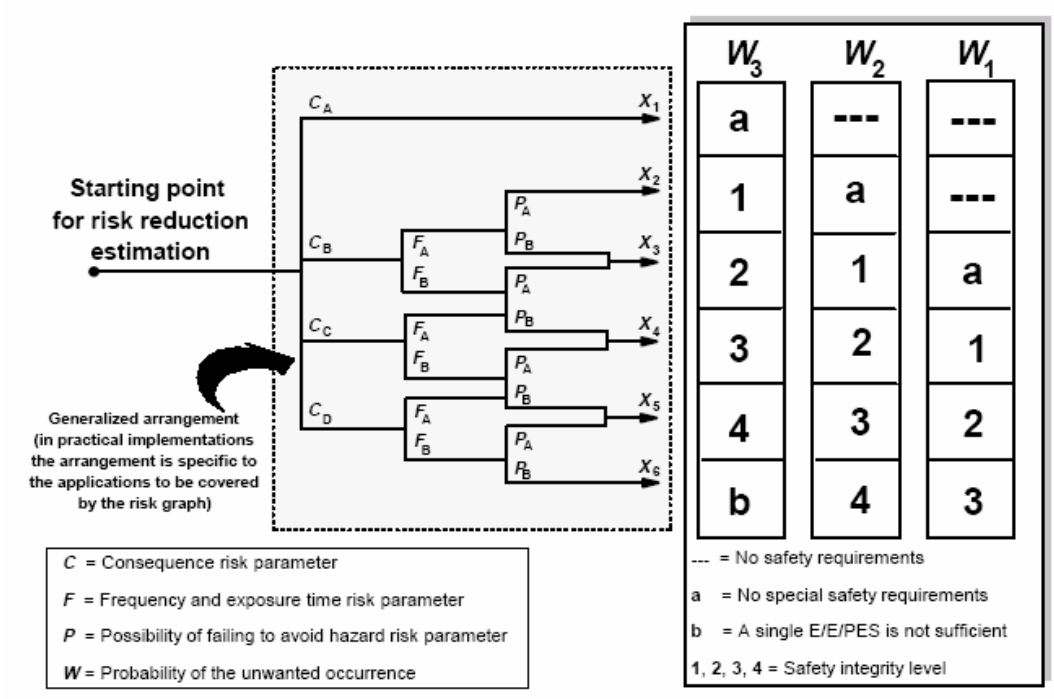


Figure 12. An example of a risk graph implementation [IEC 61508]

The safety levels are defined by means of the probability of a dangerous failure of the safety function (see table 2) but this is only part of the content of the standard. Great importance is also attached to methods in order to avoid design faults and methods in order to deal with faults that occur during operation. The table differs between continuous use of safety functions and functions that are seldom used.

Table 2. Safety Integrity Level, SIL [IEC61508]

SIL	**Low demand mode of operation** **Average probability of failure to perform its design function on demand**	**High demand or continuous mode of operation** **Probability of a dangerous failure per hour**
4	$\geq 10^{-5}$ till $<10^{-4}$	$\geq 10^{-9}$ till $<10^{-8}$
3	$\geq 10^{-4}$ till $<10^{-3}$	$\geq 10^{-8}$ till $<10^{-7}$
2	$\geq 10^{-3}$ till $<10^{-2}$	$\geq 10^{-7}$ till $<10^{-6}$
1	$\geq 10^{-2}$ till $<10^{-1}$	$\geq 10^{-6}$ till $<10^{-5}$

The draft ISO 26262 expresses the target values for random hardware failures of different automotive safety integrity levels (ASIL). (See table 3.)

Table 3. Expected random hardware failure target values [ISO26262]

ASIL	Random hardware failure target values
D	$<10^{-8}$ /h
C	$<10^{-7}$ /h
B	No requirement
A	No requirement

4.2.4 Support in standards

Descriptions of the hazard and risk analysis are given in the standards IEC 61508-1 clause 7.4, IEC 61508-5 and ISO 26262-3.

4.3 Functional safety concept

4.3.1 Scope and limitations

This chapter deals with functional safety concepts. Safety concepts are here used as a common name for all methods used to increase safety in a control system at run time.

When designing a system there is always to some extent possible to make a choice between run time concepts and design time methods. If, for example, it was possible to get error free software, no run time error detection for software errors would be needed. However, software designed following a good design process will most likely have less design errors and will therefore require less run time checking in order to fulfil the requirements of a certain safety integrity level. Methods dealing with minimizing errors at design time are not further discussed in this chapter.

Functional safety concepts then means that the safety concepts are intended to assure the safety of a function. However, a function can be small and limited like the measurement of a sensor value or large and covering a very complex function, like for instance the complete control of petrol engine.

Normally error detection and some kind of management of errors are used to implement the functional safety concepts. Fault tolerance for transient or permanent errors might be necessary. In order to detect errors different levels of independence might be required. Error management might also imply different levels of independence. A special class of independence is redundancy.

It is normally a very important design decision to decide the granularity of the functional safety concepts. Shall they cover small items or large? How shall the functionality of a system be organised in order to simplify the allocation and design of the safety concepts? Techniques and guidelines for doing a good design with respect to safety concepts implementation are not treated in this report.

Further a limitation in this chapter is that we only deal with safety concepts for use in electronic control systems for vehicles i.e. typically a microcontroller, some electronics, sensors and actuators several ECUs connected with networks etc.

It is also assumed that a function oriented design method for safety critical systems like proposed in the IEC 61508 standard and the ISO 26262 coming standard is used. It is assumed that a Preliminary Hazard Analysis is made and that different safety integrity levels have been allocated to functions with some granularity. IEC 61508 uses SILx and ISO 26262 uses ASILx to denote different safety integrity levels. The opposite to this is to first design a system and it´s functions and then afterwards use FMEA or other methods to analyse the safety of the system.

4.3.2 Systematic fault tolerance

Systematic fault tolerance meaning methods to assure that a computing platform is safe is often proposed as an alternative to functional safety. The principle is that functions executed on a fault tolerant and safe platform then automatically are safe.

There are several problems with this approach. One is that a fault tolerant platform can be very complex and expensive. Another is that it can never be made completely fault tolerant to all types of errors. What about a software design error in a function or a requirement error?

Of cause all computing platform must have a certain level of robustness towards errors but it will still be necessary and more cost efficient to deal with many types of errors at the functional level. Then only the most safety critical functions have to use the most costly error detection and handling mechanisms.

4.3.3 Fault models

Many control systems still are designed with ad hoc error detection and handling and afterwards analysed and judged regarding safety levels. This is a method much too inefficient for development projects with short designs loops and where high levels of integrity have to be reached.

The right design process is to always start with a defined fault models coupled to different integrity levels. Errors can always be detected at different levels in a control system and it has to be decided where to detect the errors i.e. define the fault model.

A goal for a good fault model is that the likelihood to detect all types of relevant errors that might lead to a functional failure, is high. A fault model on the other hand must not be too complicated. Then the implementation of safety concepts will be difficult.

4.3.3.1 Faults at the functional level

A fault model for functional levels has to make sense for typical functional level items. For example a bit flip in a program memory cell does not make sense to use in this kind of fault model. It is hard to make any coupling to how this bit flip affects a function. The full scale from no effect at all to that the function becomes erroneous and cannot be used is possible.

Examples of errors that can be useful at the functional level are:

- errors in data from sensors
- errors in variables within the software
- errors in the calculation of variables
- lack of execution of functions
- errors in communication

It might have to be mentioned that the original cause of an error in principle is irrelevant. If all essential errors can be detected at the functional level, errors occurring at lower levels are irrelevant. If they do not affect the safety critical functions it is not necessary to pay any attention to them.

In the same way it is often irrelevant whether the root cause is an electronics hardware error or a software error. Furthermore, when carefully analysed electronics hardware errors and software design errors have much more in common that can be expected. Design errors in well tested software, tends to manifest much like transient errors in electronics hardware. They occur in very rare use cases and occur very seldom. Transient errors in electronics hardware, often depends on design errors and occur seldom in rare driving situation (for example RFI). Today's microcontrollers are very complex and there is most likely no existing microcontroller without a lot of remaining design errors. These errors, behaves just like software design errors.

4.3.3.2 Different models for different integrity levels

A Safety Integrity Level whether it is a SILx as in IEC 61508 or an ASILx as in ISO 26262 can be associated with a certain level of failure probability. For example, in order to fulfil the integrity level requirements of ASILx for a certain function it has to be guaranteed that the probability of failure of that function is lower than 10^{-y} per driving hour.

It is well known that for different parts in a typical vehicle control system different failure probability levels can be achieved. For example simple pushbutton sensors have a much higher expected failure probability than for example a MOSFET transistor. There will be differences between sensors, actuators, ECU electronics, processing within a microcontroller, data storage, communication in vehicle networks etc.

Therefore different integrity levels will have different failure models. The lowest SIL or ASIL will have smaller failure models than higher SIL and ASIL levels.

4.3.4 Safety concepts

A functional safety concept is intended to detect errors and to prevent that a safety critical failure occurs. Detection and handling of errors thus are the two parts of a safety concept.

Another important aspect is transient and permanent errors. In most control systems transient errors are much more common than permanent errors. In automotive applications it is of high importance to recover from transient errors. It will be very costly for a vehicle manufacturer if transient failure recovery does not work properly. A lot of vehicles will then be inoperable without any reason.

4.3.4.1 Error detection

The ways to detect errors in electronic control systems for vehicles are numerous. Some methods especially useful to detect errors at functional levels are:

- comparison to static limit values
- comparison to predicted values
- parallel execution of another control algorithm
- check sums on variables
- counters connected to variables
- feed back from actuator behaviour
- feed back from vehicle behaviour

Error detection might also require a certain level of independence as explained below.

4.3.4.2 Error handling

4.3.4.2.1 Transient errors

The importance of transient error recovery makes it necessary to have specific treatment of transient errors. Software is often the most efficient way to handle transient errors. A certain level of independence might be required as explained below.

Common methods to handle transient errors in vehicle applications are:

- use old control parameter values
- use the erroneous control parameter value
- use a predicted control parameter value

It is all a matter of the nature of the safety critical function and the characteristics of the control system. If the control system operates with a much higher control periodicity, than the reaction time of the mechanics to be controlled, it is often easy to design an efficient transient handling. Otherwise it might be more complicated.

An important part is to decide when the transient error is to be treated as a permanent error. A commonly used methods is transient counters. When a transient counter reaches a certain number of errors or a certain frequency of errors the error is treated as permanent.

It is also necessary to recover from an error that has been treated as permanent but disappears. Again counters can be used together with number of errors or error frequency.

4.3.4.2.2 Permanent errors

The required handling of permanent errors depends on the function to be handled. Often the function can be shut down. It might be possible to operate the vehicle without it. Another common situation is that limited functionality can be used. For example a braking system with ABS can enter a mode where only braking without ABS is possible. Full functional performance despite the presence of errors and requirements for redundancy in the control system is rarely necessary in vehicle applications. If possible it is avoided due to cost. Examples might be that electrical power for electrical braking systems needs a redundant solution.
Also handling of permanent errors might require independence as discussed below.

4.3.4.3 Independence

Both error detection and error handling might require a certain level of independence at the concept level. It is a design choice per each SIL or ASIL level what level of independence that is required.

Examples of independent error detection are:

- different sensors of the same type or of another type
- different methods to read sensors like for instance A/D conversion or PWM
- different software algorithms executed in parallel
- execution of the same or a different software algorithm on another microcontroller of the same type or of another type
- getting actuator feed back by different electronics
- getting actuator feed back by different principles
- getting actuator feed back to different microcontrollers of the same type or of different types

Examples of independent error handling are:

- different software algorithms executed in parallel
- execution of the same or a different software algorithm on another microcontroller of the same type or of another type
- control of actuators by different electronics controlled by the same or a different microcontroller of the same or of another type.

Many types of independence can also be achieved using diversity during the design process but as explained earlier this is beyond the scope of this report.

4.3.5 Functional requirements

It is important to understand all the basic functions needed to achieve the safety goal. The functionality of the vehicle will be complex and the functions can be expected to interact. The functions can also be expected to have different safety integrity.

The aim of the functional safety concept is to specify functionality and safety integrity. It is not aiming at technical design details.

This specification of the basic functionalities is called the overall safety requirements by standard IEC 61508.

The objective of the overall safety requirements specification can be summarised as
- to specify the safety functions necessary to ensure the required functional safety
- to determine the necessary risk reduction for each hazardous event
[IEC61508-1, clause 7.5.1]

The functional safety concept may often be specified in natural language. Mathematical or formal expressions may also be used. Experience shows that clear and unambiguous specifications are important. A simple and easy-to-understand specification of the functions important to safety should be useful for all personnel working in the development project.

4.3.6 Support in standards

Descriptions of the functional safety concept and the safety requirements allocation are given in the standards IEC 61508-1 clause 7.5 and ISO 26262-3.

5 Detailed Safety Analysis

The development, integration and safety validation phases of the overall safety lifecycle can be seen as a detailed hazard analysis. This report summarises the activities as system development, hardware development, software development and verification and validation.

5.1 System development

The system development can be described as the specification of the technical safety concept and system design.

5.1.1 Specification of the technical safety concept

The technical safety concept develops the conclusions on functionality and safety integrity into technical requirements. It will be the result of developing the functional safety concept into technical safety requirements.

Experience shows that it may be difficult to realize when the functional safety concept turns into the technical safety concept. The main reason for this is that the development engineer has prior knowledge of the implementation of similar systems. Too many technical details seem obvious already at the concept phase.

But technical details should be left to the development phase. The technical safety concept uses the functional requirements to describe how the technical architecture and the parts of the system will fulfil the safety requirements.

This mapping of the functionalities to technical systems is called safety requirements allocation in the standard IEC 61508.

The objective of the Safety requirements allocation phase is
- to allocate the safety functions, contained in the specification for the overall safety requirements (both the safety functions requirements and the safety integrity requirements), to the designated E/E/PE safety related systems, other technology safety-related systems and external risk reduction facilities;
- to allocate a safety integrity level to each safety function.
[IEC 61508, clause 7.6.1]

5.1.2 System design

The design starts at system level, and will then be continued with detailed development of hardware and software. System design means developing the technical safety concept into electronic hardware and software. It may also be that certain safety-related functions are trusted to other technologies than electric/electronic/programmable electronic systems.

The work will result in a system design specification.

5.1.3 Support in standards

The product development at system level is described in ISO2626-4 and in IEC 61508-2. Tables of recommended techniques and measures are given in annex B IEC 61508-2.

5.2 Hardware development

5.2.1 Avoiding failures

During hardware development of the safety-related control system it is important to apply methods for avoiding that failures are introduced unintentionally. Such failures are called systematic failures and are introduced during the specification, design or the production of the Programmable Electronic control System (PES).

These failures may be difficult to detect and it is therefore important to start to apply preventive methods very early in the development in order to reduce the probability of their occurrence. The methods for handling systematic failures may according to IEC 61508 be categorized in:

Methods for the requirement specification
The complete safety lifecycle concept is introduced in order to avoid failures in the specification. In addition the following methods are recommended for avoiding failures during the specification of requirements:
-Requirements specification in natural language
-Formal specification tools
-Semiformal specification tools
-Computer-aided specification tools
-Documentation

Examples of failures that may be introduced during the requirement specification may be misunderstanding of the result from the hazard and risk analysis or of the intended use of the target item (vehicle).

Methods for the development and design
The following methods are recommended for avoiding residual failures after the PES design:
-Use the principles recommended for the control of failures during operation.
-Protection from environmental stress or influences such as: Consideration of over- and under voltage, voltage variations, separation between power lines and information carrying lines. Sufficient de-rating of components and use of well-tried components is also important.
-Documentation

Examples of failures that may be introduced during the development are logical design mistakes in the circuit diagram or mistakes in the layout of the printed circuit board.

Methods for the production of the programmable electronic control system (PES)
The production of the PES is an advanced process related to many potential sources of failure. The same methods as for the integration (see below) are applied in order to avoid the introduction of failures.

Examples of failures that may be introduced during the production are that the mounting machine turns a certain passive component in the wrong direction, the soldering on a batch of PES is poor and the etching of the PCB is erroneous.

Methods for the integration of the PES into the vehicle
The concept of integration includes the integration of software into hardware, the integration of a PES with other PESs and the integration of the hardware into the EUC. The following methods are recommended for all sorts of PES integration:
-Functional testing
-Statistical testing
-Black-box testing
-Documentation

Examples of failures introduced during integration of PES may be that an older version of the PES is installed in the vehicle instead of the intended PES, the installed PES is supplied from the wrong power line (assuming more than one) or a safety related sensor is connected to the wrong analogue input of the PES.

Methods for the operation and service
The following methods are recommended in order to avoid failures when operating or maintaining the system:
-Limited operation possibilities
-Protection against operator mistakes
-User friendliness

One example of a failure that may be caused by the driver is to take advantage of the vehicle safety systems in order to drive faster. Thus the protection will be lowered. Another example is if the incorrect version of embedded software is downloaded into the ECU during service at the garage.

5.2.2 Control of failures during operation

A fault in the hardware or in the control logic may affect a safety-related function in several ways. The worst case would be when the fault causes a failure of a safety-related function and the failure is undetected by the user of the system. The safety-related system will then be in a hazardous state. An example of such a failure is a short-circuit of an output transistor making the system output stage incapable to switch off its output current.

Fault detection
Programmable Electronic Systems (PES) have the potential of detecting faults in the hardware before a fault is manifested as a failure of the system. The techniques and measures used focus on different parts of the electronic hardware and may require different amount of system effort. It is regarded as state-of-the-art to implement techniques for fault detection in PES used in safety-related applications. The dependability of a system increases by detection and handling of faults.

The best case is if a fault does not affect any safety-related function, and is detected by internal self-checks of the system. An example of such a "safe fault" would be if memory cells containing texts for displays are unintentionally changed, but detected by internal background checksumming of the memory.

The techniques and measures implemented to find permanent faults in parts of the hardware will also be efficient to find transient faults. Transient faults due to environmental disturbances or software failures often have greater failure intensity than the permanent hardware faults. It will be of secondary importance to state if the measures are implemented for detection of permanent or transient faults.

Control of failures
In certain applications is it enough to detect a permanent hardware fault or a transient fault and then issuing an alarm. More critical applications require the fault to be handled before it causes a dangerous failure. This can only be achieved by applying redundancy. There are different kinds of redundancy:
-Time redundancy (e.g. repeated data processing or reading of an analogue input)
-Information redundancy (e.g. checksums)
-Physical redundancy (e.g. additional hardware)

There are also two different ways of applying redundancy:
Active redundancy – when all redundant functions operate continuously
Stand-by redundancy – when the redundant function will be active after a fault in the main function having been detected.

The selection of redundancy measures for parts of the control system is usually referred to as architectural constraints. The architectural constraints of the system depend on the result of the hazard and risk analysis (or on requirements of availability). They are usually divided into two main issues to determine the level of hardware fault tolerance and the behaviour at fault.

The level of hardware fault tolerance (N)
Defines the amount of independent random hardware faults the system shall withstand without dangerous operation. For example, if the level of hardware fault tolerance = 1 the control system shall handle any single-point hardware fault and continue to operate safely (but not necessarily with the same functionality). If an additional single-point fault occurs (after a while), the control system may operate dangerously.

Behaviour at fault
Defines how the control system shall react when different faults are detected. Usually there are one or several different so called safe-states defined to be entered when different types of faults occur. A safe-state is a special operational state in which the system performs a pre-defined action which is defined as safe and which the system cannot exit until the fault is removed. Depending on the hazard and risk analysis in conjunction with the intended basic functionality the most suitable behaviour at fault may be determined, e.g.
-The safety related control function shuts-down and becomes disengaged.
Such a control system has no availability but usually requires a lesser amount of redundancy.
-The safety related control function forces the outputs to a predefined value and maintains this value.
Such a control system has a low level of availability although it usually requires fairly much redundancy. The operation of the system becomes highly degraded when entering the safe-state.
-The safety related control function reduces the functionality strongly, only allowing a very small set of “safe” output combinations.
Such a control system has a medium level of availability and usually requires a lot of redundancy. The system operation becomes medium degraded when entering the safe-state.
-The safety related control function disengages the faulty part of the control system and continues to operate according to the specification.
Such a control system has a high level of availability and usually requires very much redundancy. The system operation becomes completely or almost non-degraded when entering the safe-state.

When developing redundant systems is it important to include robustness to transient faults which are very common (such as spurious glitches from relays or contacts). Therefore some redundant function should be equipped with a device that can evaluate certain faults before forcing the control system to the safe-state.

Considerations on faults and failures

The most common fault model used starts with a fault which is the actual defect itself and which cannot be measured nor predicted. The consequence of this fault leads to an error (e.g. an erroneous signal state or current) which is measurable. The consequence of this error is a new functionality (or malfunction) which was not intended or specified. This new function is called a failure (i.e. the system fails to perform the specified function). This fault model is recursive so that a failure of a subsystem may be considered as a fault in a higher system perspective.

Examples of faults:
- the short circuit of a resistor (at a detailed level)
- the omission of a CAN-message (at a sub-system level)
- the loss of one out of two hydraulic brake circuits (at a system level)

The above described fault model is theoretically correct but may in a real development of a PES be infeasible in order to ensure that enough integrity has been achieved and has to be complemented with other measures.

For example, consider a microcontroller (or any other complex integrated circuit). Even though a redundant architecture is used, by a parallel microcontroller, is it impossible to prove that these two microcontrollers truly are redundant for any internal fault that may occur: This is due to the complexity of the resulting failure which in some cases is very difficult to foresee, especially if the function of the software is also regarded. Another aspect is that the fault model does not regard the quality or the de-rating of the selected components.

Therefore is it common in standards for functional safety to introduce the measure of probability of dangerous failure, or the reliability of the safety related functionality, as a complement to the architectural constraints. Even though the safe-states are very well defined and implemented is it only a matter of probability that a fault manifests itself in such a way that the fault detecting mechanisms actually detects it and that the safe-state is entered. The next chapter in this report focuses therefore solely on the reliability analysis of safety related parts of the control system.

5.2.3 Basic reliability relations and definitions for safety analysis of electronic hardware

Reliability, availability and maintainability (RAM) analysis is widely used in industry for e.g. optimization, maintenance planning and quality assurance of manufacturing processes or products. However, the reliability theory and the application of the methods for reliability described in this report are solely intended for the analysis of specific safety-related parts of electronic or programmable electronic control systems which are only a small partition of a general RAM analysis. The main purpose of this chapter is to introduce the reliability concept in order to guide the reader on how to apply methods for safety analysis and to understand the examples described in this project. This report also clarifies the different parameters that are commonly addressed as quantitative safety requirements in standards for functional safety. The reader should at least consult the references in conjunction with reading this chapter in order to get more detailed and in-depth information about reliability theory.

For in-depth information about basic reliability theory, please refer to [MIL-HDBK-338B] and [Goble].

Basic concepts of reliability analysis
The probability that a component/function success in operation during a time interval from zero to t is defined as the reliability R(t) where $0 \leq R(t) \leq 1$. The probability that a component/function fails during a time interval from zero to t is defined as the unreliability F(t) so that

$$R(t) + F(t) = 1.$$

The mean time to repair MTTR [h] (which sometimes is called the mean down time MDT) is the expected value of the random variable “repair time” and includes both the time to detect the failure and the actual repair time required to restore the system. The probability that the system becomes repaired after a failure is defined as

$$\mu = 1/MTTR.$$

The probability that a component/function will be operational at a given point in time taking into account both the reliability and the repair rates (MDT) is defined as the system availability A(t). The availability can also be used to define the unavailability

$$U(t) = 1 – A(t).$$

The availability function reaches a steady-state level as long as the component/function is repaired in contrary to the reliability function that always reaches zero when enough time has passed.

Probability distributions
The Probability Density Function (PDF) relates the value of a random variable to the probability of getting that value or value range.

Considering a large population of electronic components (e.g. transistors or lamps) some of them will in time fail in one way or another. The PDF may be derived by registering the frequency of occurrence of these failures in a graph for the whole population as a function of time and normalize it so that the sum of all occurrences equals 1 (thus obtaining he probabilities of occurrence). The sum of all probabilities in a time period (a to b) in this graph will provide the probability that a component has failed during this time interval.

If the PDF is represented as a continuous function the integral of this function is the Cumulative Distribution Function (CDF). More generally, the CDF shows how probabilities are distributed on events, not necessarily due to time. The mathematical relationship between PDF and CDF is:

$CDF(x) = F(x) = \int_{-\infty}^{x} PDF(x)dx$ where x is a continuous random variable (e.g. time t for reliability analysis of electronics)

Different components, products or properties relate to different distribution functions, for example:

The normal distribution may be applied for e.g. measurements of product strength and external stress. For example: To determine the probability of deviation from an expected measure. This distribution provides a non-constant failure rate.

$$PDF(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t-\mu}{\sigma})^2}$$

The lognormal distribution may be applied for modelling the completion of human activities such as repair rates or to model uncertainty in failure rate information. This distribution provides a non-constant failure rate.

$$PDF(t) = \frac{1}{\sigma \cdot t\sqrt{2\pi}} e^{\frac{1}{2}(\frac{\ln(t)-\mu}{\sigma})^2}$$

The Weibull distribution is a more general distribution due to its parameters which may be used for estimating other distributions. An example is when an electronic component is exposed to e.g. wearing causing a non-constant failure rate.

$$PDF(t) = \frac{\beta}{\eta}(\frac{t-\gamma}{\eta})^{\beta-1} e^{-[(\frac{t-\gamma}{\eta})^{\beta}]}$$

The exponential distribution is the simplest distribution which has the following probability density function:

$$PDF(t) = \lambda e^{-\lambda t}$$

and the CDF(t) = the unreliability function $F(t) = 1 - e^{-\lambda t}$ where λ is the failure rate (which also is constant, see 5.2.4 below). The reliability function for the exponential function is $R(t) = e^{-\lambda t}$. It can be shown that $F(t) = 1 - e^{-\lambda t}$ can be approximated with $F(t) \approx \lambda t$ when λt is small. This approximation should be used with caution because it is only valid when λt is small. However, it may be used in most cases for electronic components/circuits since their related failure rates often are very small. This may not be the case for e.g. mechanic or hydraulic systems because their failure rates often are much larger than for electronic components.

5.2.4 Failure rate

The failure rate can be used to describe the control system reliability. The failure rate is expressed as the number of failures per hour and is designated by the letter λ (”lambda”). Normal failure intensities for electronic components are very low, typically in the magnitude of 10^{-5} to 10^{-9} failures/h.

Generally the instantaneous failure rate is λ(t) (which is also known as the hazard rate function h(t)) and calculated as: $\lambda(t) = \dfrac{\frac{d}{dt}F(t)}{R(t)}$

The life of an electronic equipment or system can be shown as a failure rate versus time curve (which should not be confused with the PDF) which for this type of technology usually shows the characteristics of a bath-tub as shown in figure 13. Some types of systems or technologies are characterized by other curves such as the roller-coaster curve, but the bath-tub curve has been generally accepted for electronics over the years.

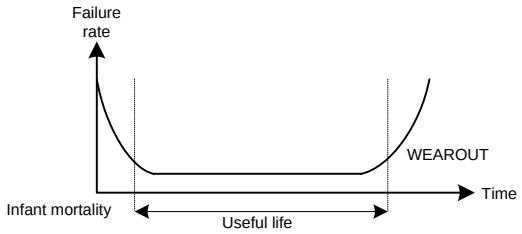


Figure 13: Bath-tub curve

The bath-tub curve describes the average failure rate of a population of items in time and is divided into three phases where the first denotes the items failure rate in the early product life. In this phase several products suffer from e.g. defects introduced by the manufacturing process. Therefore the failure rate is initially high but decreases rapidly. In the next phase most of these defects are removed and the failure rate becomes more or less constant for the population. In the last phase the population has aged by use and the failure rate increases.

The different phases in the bath-tub curve (or any other failure rate versus time curve) may be characterized by different probability distributions such as:
Infant mortality phase: e.g. Weibull-distributed failure rates
Useful life phase: e.g. Exponentially distributed failure rates
Wear-out phase: e.g. Normal distributed failure rates.

Depending on the engineering practice of the target system/equipment and the source of failure rate data the different phases are more or less dominant. For electronic components/circuits the period of more or less constant failure rate is the longest period and therefore the exponential distribution is most commonly used. For certain components other distributions may be more suitable, but this is not further considered in this report.

For most electronic components there is a statistically determined period of time called useful life (or θ - life expectancy), see figure 13, during which the failure rate is assumed to be constant. When this time period elapsed the component should be disposed because of wear-out.

When the wear-out period begins depends on several factors such as how the equipment/component is used due to its ratings. In a larger system where different components have different lengths of useful life the different components will cause the wear-out failures to occur at random times. This results in an overall constant failure rate and an exponential behaviour.

The proof test interval (T) is the time interval for which the developer ensures that the estimated failure rate for the product is valid. The proof test is intended for detection of dangerous failures and in that case to restore the system in an “as new” condition or as close as practical to this condition. Such a test may however never be exhaustive since it is not possible to test the reliability of single electronic components. Even though proof tests are carried out with fairly high frequency the reliability graph of the actual system will decrease during the mission time as in figure 14.

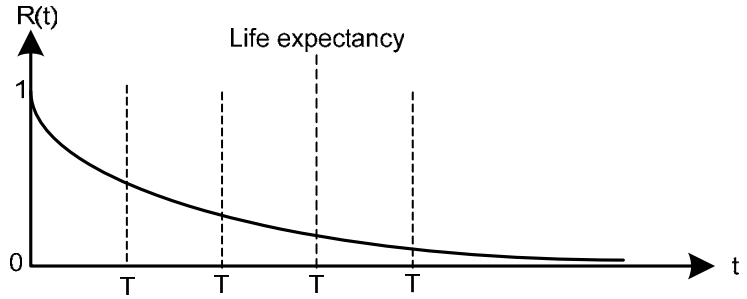


Figure 14: System reliability with proof tests performed without finding dangerous failures

Therefore it is in practice common to consider the proof test as the actual mission time and to replace the system (as many parts as practical) when the proof test interval has elapsed as shown in the second graph in figure 15.

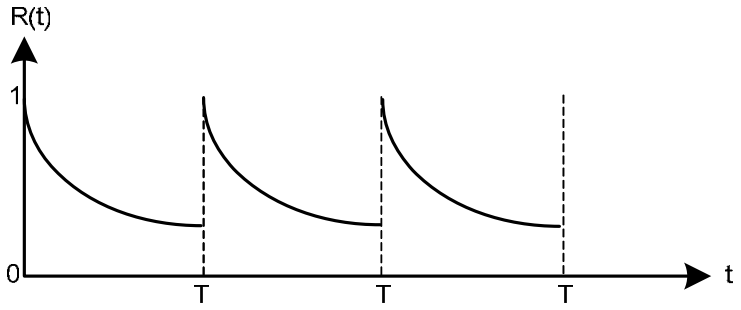


Figure 15: System with periodic (ideal) system exchange

During the development of a safety related control system the most usual means for obtaining failure rate data for electronic components are to use a reliability prediction handbook. An example of such a handbook is the [MIL-HDBK-217F]. In time, when the product has existed on the market, some of the predicted failure rates may be refined due to statistical testing. It is although important to keep in mind that a major part of any used reliability data source for a product is predicted (i.e. not measured or observed) and therefore is related with some uncertainty.

The reliability data prediction handbooks do not guide on how to determine the useful life time for all categories of component families, so the life time expectancy has in most cases to be determined by engineering experience or from statistical tests of products.

When the failure rate (λ) of an electronic component is determined the system failures (effects) caused by the components different fault modes are categorized as safe or dangerous according to:
- faults leading to a safe state
- faults leading to a dangerous state

The corresponding failure rates may be described as λ_S ("safe failure rate") and λ_D ("dangerous failure rate"). (See figure 16.)

The total failure rate may be expressed as

$$\lambda_{TOTAL} = \lambda_S + \lambda_D$$

Figure 16. The total failure rate divided as “safe” or “dangerous”.

It is necessary to categorize the faults into different groups depending on how they affect the safety function and if they can be detected by self-checks:
- dangerous undetected faults
- dangerous but detected faults
- safe undetected faults
- safe and detected faults

The failure intensities of the different types of failures (see figure 17) are called
- λ_{sd} ("safe detected")
- λ_{su} ("safe undetected")
- λ_{dd} ("dangerous detected")
- λ_{du} ("dangerous undetected")

The total failure can then be expressed as

$$\lambda_{TOTAL} = \lambda_S + \lambda_D = \lambda_{SD} + \lambda_{SU} + \lambda_{DD} + \lambda_{DU}$$

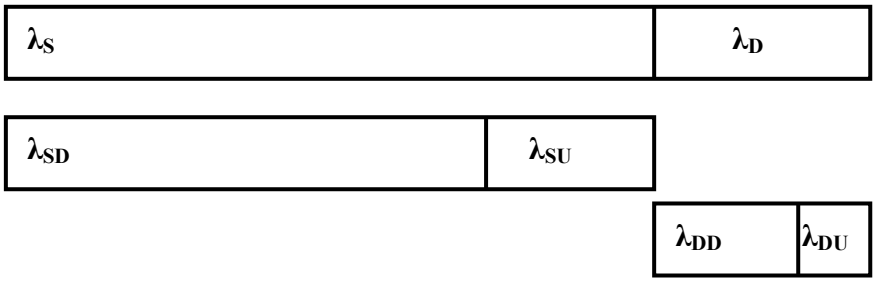


Figure 17. The total failure rate divided as “safe”, “dangerous”, “detected” or “undetected”.

The most cumbersome faults are the faults which are dangerous but undetected.

Failure rate is often expressed in the unit FIT (Failure in Time). One FIT equals 10^{-9} failures/hour.

5.2.5 Failure rate and MTTF

Elapsed time between failures is often used instead of failure rate. It will usually be expressed as mean time since it will be impossible to calculate the exact time. The MTTF (Mean Time To Failure) value specifies the expected mean time to failure. The MTTR (Mean Time To Restoration) value specifies how much time is expected for repairs etc. before the system can be restored after a failure.

The mean time to failure MTTF is defined as: $MTTF = \int_0^\infty R(t)dt$

The MTTF is a statistical measure which should be handled carefully and shall not be confused with the useful life as described in the following example:
A linear circuit has a predicted failure rate of 50 FIT which gives MTTF = 2300 years. The MTTF should be read as if 2300 such linear circuits operated continuously for one year is it likely that at least one of them has failed. The MTTF is not a minimum guaranteed life time of a single component.

The value MTBF (Mean Time Between Failures) is used to describe the elapsed time between failures.

MTBF = MTTF + MTTR [h]

when MTTF >> MTTR (the usual case) will make it possible to approximate

MTBF ≈ MTTF

Calculations regarding dangerous failures often use the value $MTTF_d$ ("Mean Time To Dangerous Failure"). This value describes the mean time to dangerous failure of a subsystem.

If the failure rate is assumed to be constant the relationship between failure rate and MTTF is:

$$\lambda = \frac{1}{MTTF} \text{ [failures/hour]}$$

Since MTTF >> MTTR (usually), the failure rate may be expressed as

$$\text{failure rate } \lambda = \frac{1}{MTTF} = \frac{1}{MTBF + MTTR} \approx \frac{1}{MTBF} \text{ [failures/hour]}$$

5.2.6 PFH and PFD

The probability of dangerous failure per hour (PFH_D) is a very common quantitative safety target for safety-related systems with continuous operation and is the average probability of dangerous failure per hour. It is a function of the failure rate (for dangerous failures) and the MTTR ($PFH_D = f(\lambda_d, MTTR)$) and is a measure of the safety related unavailability of the control system. For a single channelled system, where the system needs to immediately enter a safe state for any detected fault, the PFH_D may be calculated as $PFH_D = \lambda_{DU}$. However, most electronic control systems are redundant which provides a non-constant failure rate. For these systems the average probability of dangerous failure per hour $PFH_{avg,d}$ is of primary interest. Generally the $PFH_{avg,\,d}$ calculation is always based on the definition of the instantaneous failure rate. For a complex system the $PFH_{avg,\,d}$ calculation may be carried out by calculating the probability that a transition to a specific (dangerous) type of state occurs per hour. A coarse approximation may be used by consider the whole system as a component with constant failure rate. By calculating the MTTF for this component the average failure rate per hour may be approximated by $PFH_D = 1/MTTF_D$.

The probability of dangerous failure per demand (PFD_d) is another commonly used quantitative measure which applies for systems/functions with non-continuous operation. Such systems are not so common in road vehicles where most safety-critical systems are required to be continuous in operation. As for the PFH_d the PFD_d is a function of the dangerous failure rate and the mean down time $PFD_d = f(\lambda_d, MTTR)$ and is calculated by similar methods as PFH_d. Generally the PFD_d is the average of the total system cumulative distribution function over a limited time period. The PFD_d calculation may be carried out using a Markov model for calculating the average probability that a transition will occur to a specific (dangerous) state e.g. n times during the proof test interval.

5.2.7 Diagnostic Coverage

It is easy to imagine faults that can cause unexpected behaviour of the equipment under control (EUC). A bit in a memory cell may be stuck at ”0” or ”1”. The output circuits may be stuck at ”ON”. A software fault may cause a task to enter an ”eternal loop”. Perhaps interruptions in the power supply, or variations in the voltage level, may influence the execution of the software. Data transferred on serial communication lines may be distorted by interference. An internal CPU fault might cause incorrect execution. There are techniques and measures to automatically detect such faults before the EUC gets out of control.

Self tests are built into systems and operate automatically without being noticed by the user before a fault is detected. Simple tests will not detect all hardware faults. Elaborate tests will detect many hardware faults at the cost of much processing effort spent. The diagnostic coverage, DC, for safety-related applications is defined as the fractional decrease of the probability of dangerous hardware failure resulting from the operation of the automatic diagnostic tests. [IEC 61508-4, clause 3.8.6] If the test detects all faults, the coverage will be 100%. If no faults are detectable, the coverage will be 0%.

$$\text{diagnostic coverage DC} = \frac{\textit{the probability of detected dangerous failures}}{\textit{the probabilty of total dangerous failures}}$$

The diagnostic coverage is a measure of the efficiency of the self tests. It can be expressed either for a complete control system or for a subsystem. It may also be calculated as

$$\text{diagnostic coverage DC} = \frac{\sum \lambda_{DD}}{\sum \lambda_{D}} \quad [\%]$$

where λ_{DD} is the failure rate for dangerous detected faults and λ_D is the failure rate for dangerous failures. A DC=91% means that 91% of the possible dangerous faults will be detected.

The DC may be expressed for the whole safety-related system, or for parts of the system. The DC may typically be determined for a sensor, an ECU or for the final control elements.

It may be hard to find numerical values for the probabilities of different faults. It is possible to make a numerical calculation of the coverage of some methods. The coverage of other methods may have to be expressed in qualitative ways such as ”high/medium/low”. An estimation of the diagnostic coverage will be needed to be able to compare two diagnostic test methods.

A translation from the qualitative definition “low/medium/high”, to a quantitative measure expressed as a percentage will be needed. This report has chosen to follow the definitions suggested by the IEC 61508 standard. (See figure 18.) High coverage is used for techniques and measures with a probability higher than 99% to detect a fault. Medium coverage means a probability less than 99%, but higher than 90%. Low coverage will correspond to a diagnostic coverage greater than 60%, but lower than 90%. Techniques and measures offering less than 60% probability to detect faults are to be avoided in safety-related parts of control systems.

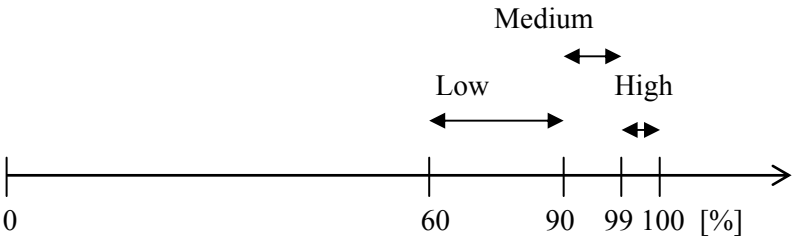


Figure 18. Diagnostic coverage defined as low, medium and high.

When numerical values are needed in calculations, 60% is used for “low” coverage, 90% is used for “medium” coverage and 99% is used for “high” coverage. A number of assumptions are used at the quantification of the diagnostic coverage for different methods of fault detection. It is not possible to state a probability that will be valid in all cases for all hardware components. The lack of data concerning the various types of memory chips, and the assumption that the potential faults are equally distributed introduce a number of uncertainties. The probability of different faults in the processing unit will depend on the type of processor, the manufacturer, the production process, the design etc. Faults in the programme sequence will have different probabilities depending on the programming language, the experience of the programmer, the testing effort etc.

The most valid estimation of diagnostic coverage for a fault detecting method should at this stage be limited to one of the three levels referred earlier in this section; low, medium or high. The level chosen may be different if probability, or numbers of errors, is used for the definition of diagnostic coverage. However, the level will be the same if all faults are equally probable.

5.2.8 Safe Failure Fraction

A fault in a safety-related function should not cause a hazardous situation. It is necessary to have a value to be able to compare the part of safe failures for different designs. (See figure 19.) That value is the Safe Failure Fraction (SFF).

SFF = (safe failures + dangerous detected failures) / (dangerous failures + safe failures)

$$\text{SFF} = \frac{\lambda_s + \lambda_{dd}}{\lambda_d + \lambda_s} \quad [\%]$$

The safe failure fraction is taken into account when determining the architectural constraints on hardware safety integrity according to standard IEC 61508-2.

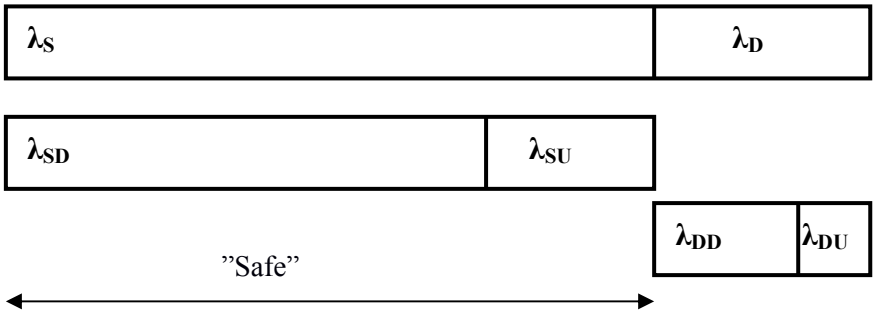


Figure 19 . Safe Failure Fraction (SFF)

The SFF=100% for an ideal system. All faults are regarded as “safe” or as being detected by the automatic self tests. Such a system hardly exists in reality. But the intention is always that the majority of faults must not cause a failure of the system.

5.2.9 Selection of techniques and measures to control failures

Techniques and measures to control failures should be implemented both to control random failures during operation and to control undetected systematic faults.

Recommendations are given [IEC 61508-2, annex A] to control failures in
- processing units
- invariable memory ranges
- variable memory ranges
- I/O units and interface
- data paths
- power supply
- program sequence
- ventilation and heating
- clock
- communication and mass-storage
- sensors
- actuators

The techniques and measures will have different coverage. Systems with an expected high level of risk reduction are expected to use techniques with high coverage, while systems with an expected moderate risk reduction are expected to use techniques with low or medium coverage.

5.2.10 Support in standards

The product development at hardware level is described in IEC 61508-2 and ISO 26262-5. Tables of techniques and measures are given in Annexes A and B of IEC 61508-2.

5.3 Software development

The hazards must be considered early in the safety lifecycle with the complete vehicle in mind. Functions important for safety must be defined. The safety goal and design principles shall be decided at system level. The design at system level will then be elaborated at the software level. (See figure 20.)

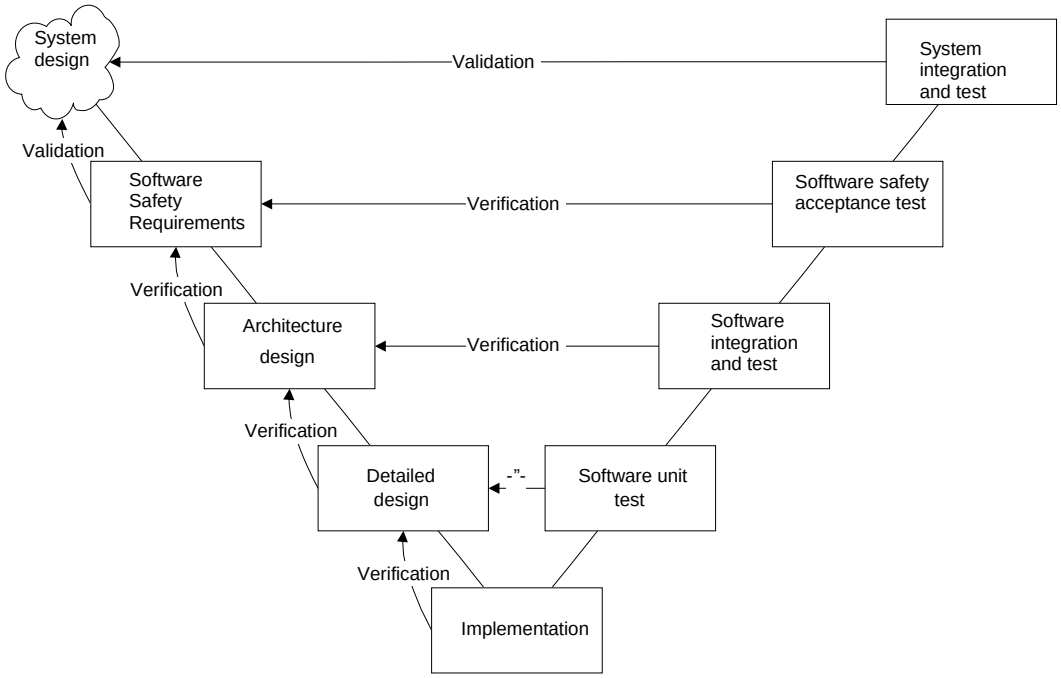


Figure 20. The "V model" describing software development

It is difficult to calculate a number for the reliability of a software package. The use of numerical values to describe the software failure rate should be avoided. The basic principle used to develop safety-related software is to choose sufficient development techniques and development methods. The techniques and methods are used to avoid the introduction of systematic faults in the software. They will also introduce mechanisms to detect and handle faults during operation of the system. A software package of high integrity requires stringent development methods to be applied. The application of several development methods to the same piece of software will also help to increase the integrity.

A software package with limited influence on functional safety will also require a well-structured development procedure. But the development techniques applied may require less effort and leave less support. However, the software is regarded as important to the functional safety and shall be developed to a certain minimum standard.

5.3.1 Software safety requirements specification

Safety requirements for the automotive system shall be developed at system level. Safety requirements relevant for the software can be derived from these requirements. To specify the software safety requirements will be the first phase of the product development at software level.

Among the items listed in the software safety requirements specification are [IEC61508-3, clause 7.2.2.11]:

a) functional requirements
– functions to enable the system to achieve or maintain a safe state;
– functions related to the detection, annunciation and management of faults in the programmable electronics hardware;
– functions related to the detection, annunciation and management of sensor and actuators faults;
– functions related to the detection, annunciation and management of faults in the software itself (software self-monitoring);
– functions related to the periodic testing of safety functions on-line;
– functions related to the periodic testing of safety functions off-line;
– functions that allow the system to be safely modified;
– interfaces to non safety-related functions;
– capacity and response time performance;
– interfaces between the software and the electronic control unit.

b) the requirement for the software safety integrity:
– the safety integrity level(s) for each of the functions listed in a).

Computer aided specification tools, semi-formal specification methods, formal specification methods and specification in natural language are examples of applicable techniques and measures. [IEC61508]

There are also several techniques and measures available to verify the software safety requirements specification.

5.3.2 Software architecture and design

The software specification will be used to develop a specification of the major components of the software, how they interface, and how the required safety integrity will be achieved. The safety requirements shall be assigned to software components.

Fault detection and diagnostics, diverse programming, recovery mechanisms and computer-aided specification tools are examples of applicable techniques and measures. [IEC61508]

There are also several techniques and measures available to verify the software architecture and design.

5.3.3 Software implementation

The software implementation will be different if code-based software development or model-based software development is used. Manual implementation of code will require certain techniques and methods, while the use of models instead of written specifications require other techniques. There are modelling tools where automatic code generation is possible. Such software development will often be combined with manual development of certain hardware-related code modules.

The software implementation of a code-based development usually means portioning of the major components of the system into modules, procedures and functions. This will be followed by the coding.

The software implementation techniques can also depend on the size of the system. Development of small systems may combine the architectural design and the detailed software implementation.

Computer aided design tools, semi-formal methods, formal methods, design and coding standards and modelling are examples of applicable techniques and measures. [IEC61508]

There are also several techniques and measures available to verify the software implementation.

5.3.4 Software integration and test

The developed software modules will be integrated in the complete embedded software package. The software package will then be integrated in the electronic control unit.

The testing and integration shall be specified during the architectural and detailed design phases. A test plan and test specification shall be produced. The performed integration tests shall be documented.

Functional tests, performance testing, interface test and simulation are examples of applicable techniques and measures. [IEC61508]

5.3.5 Support in standards

The product development at software level is described in ISO26262-6 and in IEC 61508-3. Tables of techniques and measures are given in Annexes B and C of IEC 61508-3.

Support of software safety requirements can be found in clause 7.3 of IEC 61508-3 and in ISO 26262-6.

Support of software architecture and design and software implementation can be found in clause 7.4 of IEC 61508-3 and in ISO 26262-6.

Support of software integration and test can be found in clause 7.5, 7.7 and 7.9 of IEC 61508-3 and in ISO 26262-6.

6 Safety verification and validation

6.1 Safety validation and the overall safety lifecycle

Development of safety-related electronic systems cannot easily be described in one single model covering the development cycle, the overall safety lifecycle and the validation activities. One important reason for this is the iterative process used in all practical development projects. Another reason is the use of model based development where the initial validation is performed on models early in the development process, and the final validation is performed on the completed product. “Incremental validation” where pre-validated components are used to build dependable systems is another aspect which complicates the description of the validation process.

It is important to connect the validation activities to the development lifecycle. Verification and validation are parts of the development, and must support for the realisation of the product. Safety validation can nevertheless be independent of the realisation activities. A well-functioning independent validation will be accepted as an important safety acknowledgement by the design team.

The overall safety lifecycle specified by the IEC 61508 standard is chosen as a basis for the validation work. Specifically the following activities are connected to the validation:

- Hazard and risk analysis (mapping to phase 3 of the safety lifecycle)
- Specifying safety requirements (mapping to phase 4 of the safety lifecycle)
- Allocating the safety requirements (mapping to phase 5 of the safety lifecycle)
- Validation planning (mapping to phase 7 of the safety lifecycle)
- Verification and validation activities during development (mapping to phase 9 of the safety lifecycle)
- Overall safety validation (mapping to phase 13 of the safety lifecycle)
- Developing a safety case (mapping to several phases of the safety lifecycle)

The hazard and risk analysis will point out the risks and form the foundation for the design of the safety-related system.

The safety requirements are needed as input to the validation activities. All safety functions and their corresponding safety integrity levels must be listed. Unnecessary requirements should not be introduced. The traceability of sources must be documented to enable identification of all requirements.

The safety functions will be allocated to different subsystems of the automotive application. Some of them may be allocated to software or electronic hardware. Other safety functions may rely on conventional mechanical or hydraulic hardware.

The validation plan is developed in parallel with the realisation. It specifies how the overall safety validation shall be carried out. Test cases, validation methods, pass/fail criteria etc. are specified.

The overall safety validation is the activity to confirm that the right system has been built.

The safety case collects all safety evidence from the verification and validation. It also lists the safety requirements. A safety case reports more than the safety evidence generated by verification and validation. Examples of additional safety arguments are the use of an adequate development methodology and skilled staff.

Validation is defined as confirmation and provision of objective evidence that the particular requirements for a specific intended use are fulfilled. Safety validation gives evidence that the expected safety has been reached. The developer of vehicles, or components for use in vehicles, needs the proof of an adequate level of safety. It may also be requested by authorities to demonstrate safety issues as part of the type approval process.

6.2 Validation methods

A set of validation methods (a "tool box") is needed. There is no single validation method that can provide all answers to the safety questions raised. However, there are lots of different techniques, methods and tools to support verification and validation. A limited set of tools has to be recommended as "the contents of the tool box" used for validation.

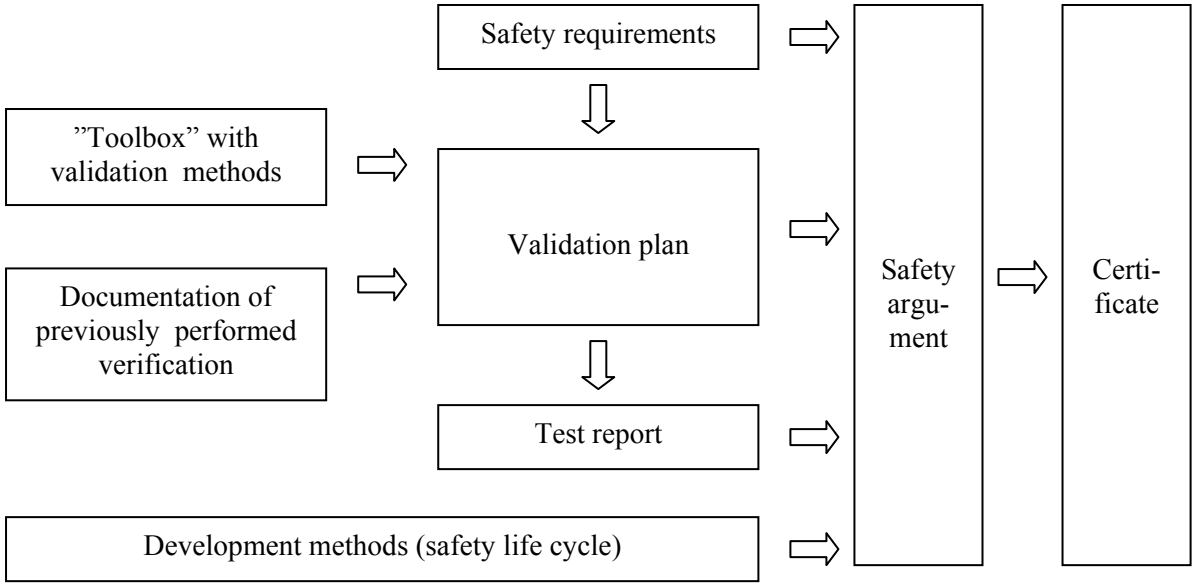


Figure 21. Elements of the validation process

The validation methods have to be combined together in a validation plan. (See figure 21.) The plan shall list requirements and validation methods. A validation procedure is useless without solid safety requirements. Both the safety functions and a measure of their performance (or integrity) must be specified. Activities performed earlier during the verification in earlier phases of the development work may also serve as evidence for adequate safety.

Different validation methods are applicable at different stages of the development life cycle. The overall safety validation is, according to the overall safety life cycle, conducted after the realisation and installation of the safety-related system. There are also other phases of the life-cycle that are important for the validation. Techniques and measures applied in other phases than "the overall safety validation phase" will contribute to the safety validation.

The safety validation has also to cope with the fact that the suppliers of the subsystems perform the validation of the subsystems. The automotive manufacturer will validate the safety of the total system. Thus the result of the validation of the subsystem will be an input to the validation of the total system.

A large number of techniques and measures exist for safety validation. It will be necessary to recommend a limited number for use in automotive applications. But it cannot be expected to find a set of techniques and measures applicable for all automotive systems. The different realisation techniques and the different types of systems will require several available methods.

6.3 Verification, validation and functional safety assessment according to IEC 61508

Techniques and measures are applied all through the development life cycle to verify the functional safety of a safety-related device or system. Verification is defined as confirmation by examination and provision of objective evidence that the requirements have been fulfilled. [IEC61508-4, clause 3.8.1] Verification is the activity of demonstrating for each phase of the relevant safety lifecycle, by analysis and/or tests, that, for the specific inputs, the deliverables meet in all respects the objectives and requirements for the specific phase.

Overall safety validation is performed as a specific phase after the realisation of the system, to confirm by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled. [IEC61508-4, clause 3.8.1] Validation is the activity of demonstrating that the safety-related system under consideration meets in all respects the safety requirements specification for that safety-related system.

Results of both the verification and the validation (V&V) may be used as evidence to demonstrate why a certain device or system may be considered safe enough for its intended use.

The verification and validation activities of the project are integrated partsof the development work. Several of the activities have to be performed at a certain phase of the development life cycle as a confirmation to continue the development work. It may not always be suitable to wait for confirmation until the overall safety validation. Late design modifications will be expensive and time-consuming.

Functional safety assessment is not the same as verification and validation. The functional safety assessment is an independent activity to investigate and arrive at a judgement on the functional safety achieved by the safety-related systems. It shall be carried out throughout the lifecycles, and may be carried out after each safety lifecycle phase or after a number of safety lifecycle phases.

The different V&V activities for the system [26262-4], the hardware [26262-5] and the software [26262-6] are interpreted in the overall safety lifecycle for the automotive industry and illustrated in figure 22

Compliance with standards for functional safety requires all aspects of the standard to be considered. Most standards address both the product properties and the properties of the development process.

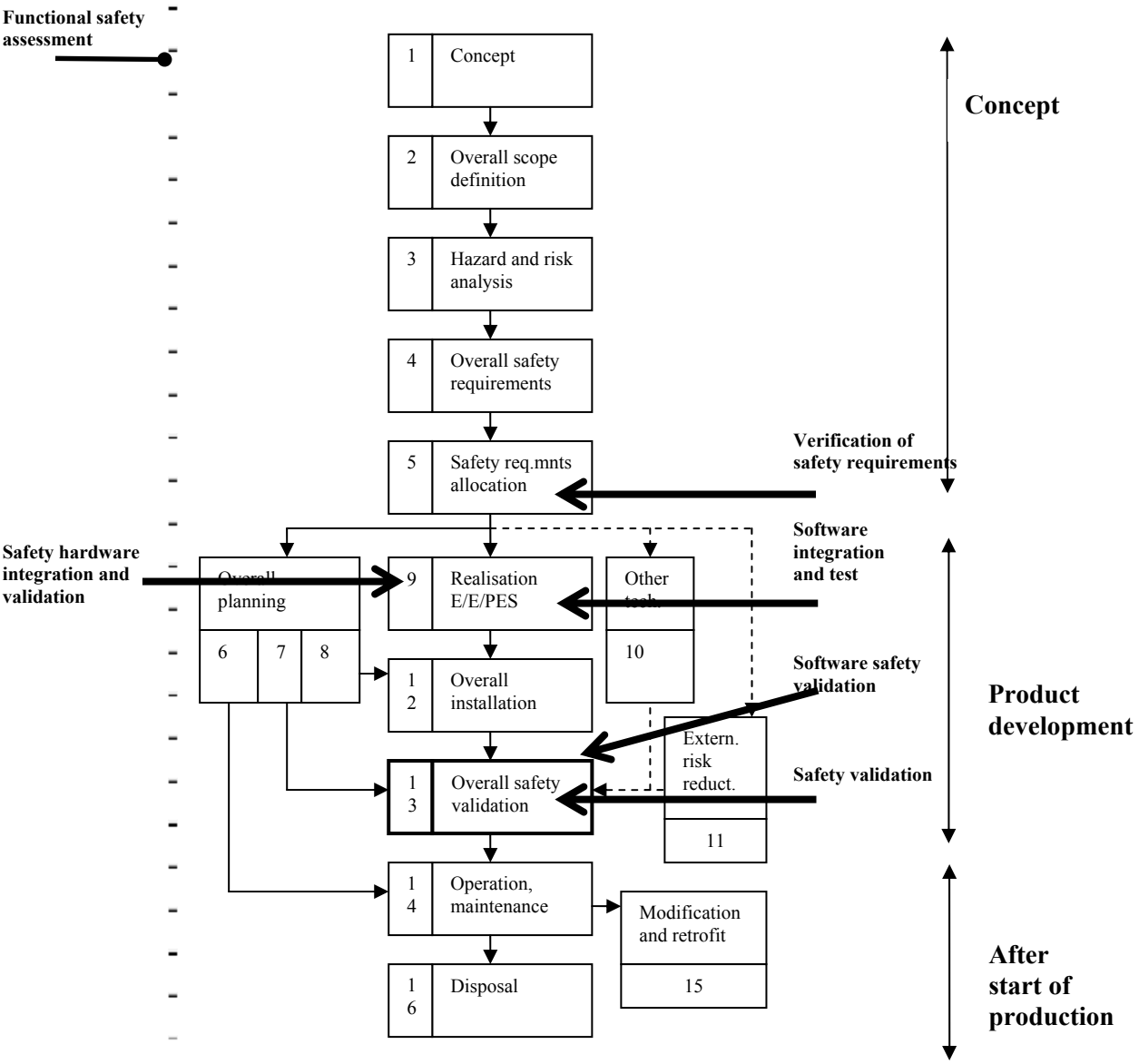


Figure 22. Verification and validation activities in the IEC 61508 overall safety lifecycle

6.4 V&V according to the AutoVal project

The validation plan may be divided into several parts describing different aspects:
- functional safety requirements
- hardware safety integrity
- software safety integrity
- overall safety integrity (including safety-related functionality)

The functional safety assessment can be divided into two parts:
- development procedure according to the safety life cycle
- safety management

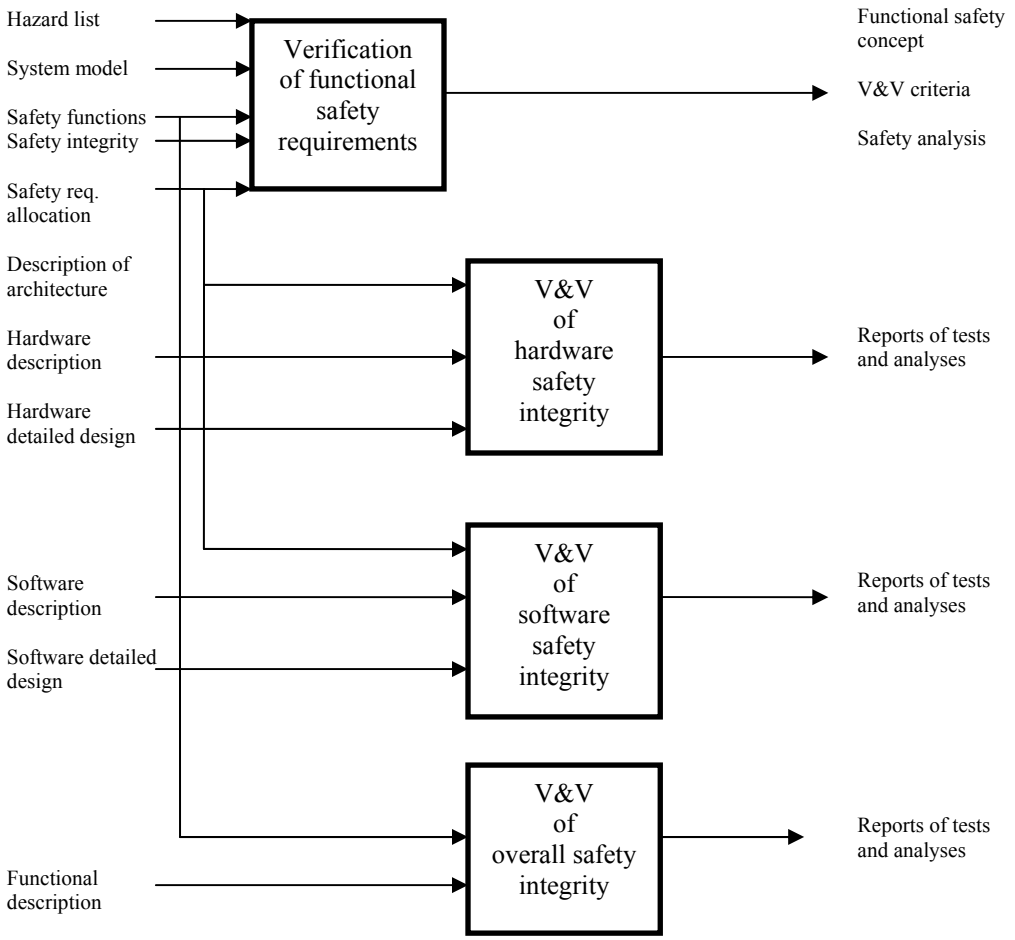


Figure 23. Verification and Validation

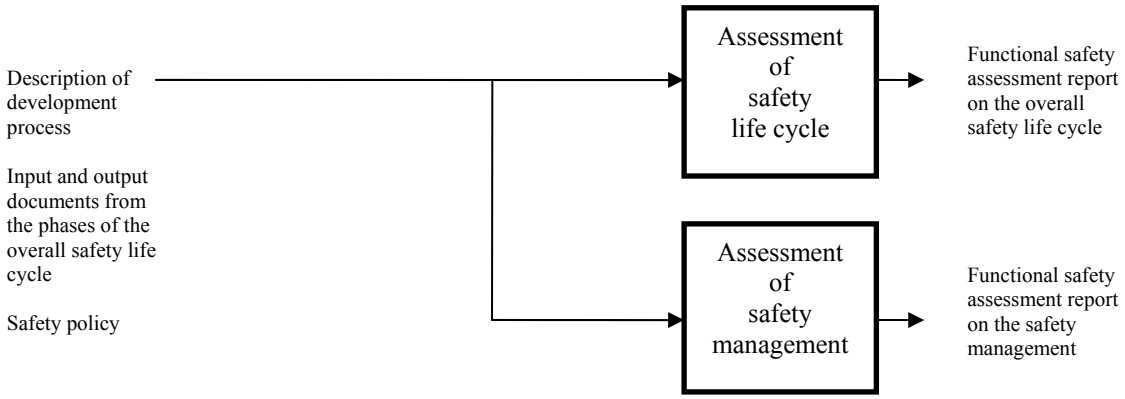


Figure 24. Functional safety assessment

6.5 Validation plan

The validation plan specifies how the overall safety validation will be carried out. The validation methods and the safety requirements are listed or referenced from other documents. The validation methods and the safety requirements are the main elements of the validation plan.

The pass/fail criteria shall be listed in the validation plan. Each safety requirement shall be tested in the validation process and the passing criteria shall be declared. It is also important to declare the person(s) who make the decision when something unexpected happens, and to approve the result of the safety validation.

Validation planning is fairly straight-forward if all validation is performed when a "final" prototype has been developed. That is normally not the case. The models developed of the system may be used to validate certain functions and aspects of the design, even before the work with the detailed design has started. Some software functions may be validated before they can be executed at the intended hardware platform. The validation activities may be grouped depending on if they are applied for the model, the software, the hardware or the complete system.

Use of pre-validated components must also be specified. A subsystem may already have been validated for use with an intended interface. The objective would then be to make use of the already performed validation of the component. Examples of pre-validated components may be generic control systems, software packages, sensors or actuators. The use of pre-validated components requires that they are used according to the specification and in the way intended.

The validation plan should be developed in parallel to the activities to implement the hardware and the software of the system.

The persons, department or organisation responsible for performing the validation should be specified in the validation plan. The degree of independence from the design team should be explained. High-integrity systems may require validation by an independent organisation. Systems for less critical applications may be sufficiently validated by a person not taking active part in the realisation of the system. Validation performed by the design engineer should always be avoided for safety-critical systems.

The validation plan should include:
- validation methods
- safety requirements
- pass/fail criteria
- specification of pre-validated components to be included in the system under validation
- the persons/department/organisation responsible for the safety validation
- the fault model applied
- references to specification documents

6.6 Support in standards

The safety verification and validation are described in the standards
IEC 61508-1 clause 7, IEC 61508-2 clause 7, IEC 61508-3 clause 7,
ISO 26262-4 and ISO 26262-8.

7 Application examples, damper system

An electronic damper system is used to demonstrate some of the principles for the development of safety related systems. The semi-active damper system is a simplified example which may not provide the functional safety required for a road vehicle. The examples of this chapter are not sufficient to claim a safe and reliable automotive system. Neither can the conclusions for this damper system be regarded valid for all other types of damper systems.

7.1 Item definition

This application example is a dependability analysis of a typical future semi-active damper system (SADS) for passenger cars. The analysis has been performed in accordance with ISO/WD 26262.

7.1.1 Objectives

The functional objectives consist of

- Improved dynamic behaviour of the vehicle. The system should adapt to different driving situations.
- Increased flexibility in terms of vehicle characteristics, allowing for different driver preferences.
- Improved safety level for all driving situations.

Production objectives are

- Portable and scalable software- and hardware implementation.
- Low-cost solution.

7.1.2 System Requirements

The requirements are derived directly from the objectives.

7.1.2.1 Improved dynamic behaviour

- Awareness of the driving situation, e.g. access to relevant real-time environmental- and vehicle data (velocity, acceleration, brake power, steering angle).
- Ability to control vehicle dynamics, e.g. adjustable dampers.
- Different control policies depending on the situation.

7.1.2.2 Increased flexibility

- Offer different modes, e.g. comfort or sport mode. (Requires vehicle-driver interaction.)

7.1.2.3 Improved safety level

- A control approach that, for any given situation, improves, or at least maintains the safety level (in the chassis domain).

7.1.2.4 Portable and scalable software- and hardware implementation

- Well-defined software architecture (e.g. system layers and services).
- Robust and flexible hardware, supporting different types of dampers and sensors.
- Tuneable control system parameters.

7.1.3 System Overview

The main system components are four damper actuators, the damping factor of which is controlled by electrical current and three body accelerometers that measure vertical acceleration. There is also a vehicle CAN bus that provides the system with the real-time data needed for driving situation awareness.

In order to switch between the modes offered by the system, an existing sport/comfort switch is used. The value of this switch is distributed on CAN. The main part of the system is the ECU that runs the control algorithms.

7.1.4 Control Procedure

Given the previously described hardware, the control procedure can be summarized as follows.

1. Data acquisition. Read sensors and get signals from CAN.
2. Run main control algorithm and generate a new damping factor request for each damper.
3. Run the local current controllers that try to fulfil the damping factor requests by controlling the current flow through each damper. The output values are damper PWM duty cycles.
4. Update damper actuators with the new PWM duty cycles.

7.1.5 Software

7.1.5.1 Architecture

For the embedded ECU software, a general architecture that is heavily inspired by AUTOSAR is used. An overview of this architecture is shown in figure 25.

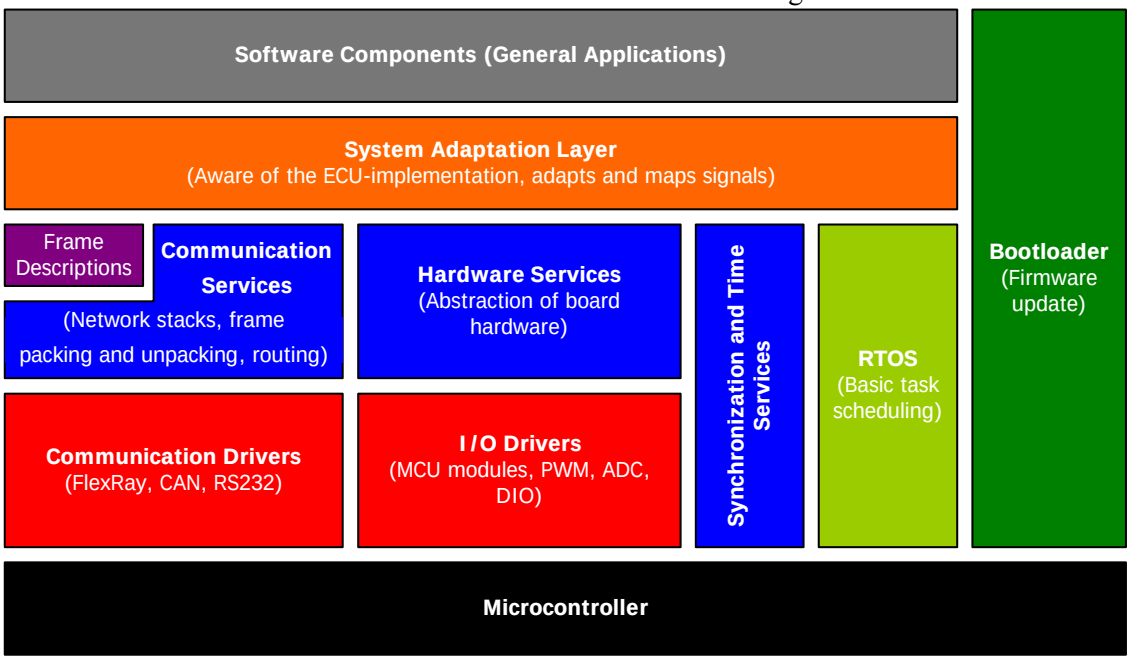


Figure 25 - Overview of the software architecture.

The idea is to deploy high level software components (such as control systems or other applications) on a highly abstracted interface. The underlying services provide standardized interfaces to communication, I/O etc.

7.1.5.2 Method and Implementation

All control systems are designed in Simulink. There are two such models – one for the main, global controller and one for the local current loops that control each damper. The resulting software components are C-language modules generated using Realtime Workshop with Embedded Coder. They are deployed directly to the software architecture.

As previously described, the control systems require several CAN signals as inputs. To enable the underlying communication services to extract the relevant signals for this ECU, a vehicle database is provided. This database defines all the signals needed by the control system. A tool is used to generate OSEK COM-compatible communication services out of the signal database. The output from this step is a set of C-language modules that expose a high-level signal interface to the software components.

In figure 26, the work methodology is illustrated.

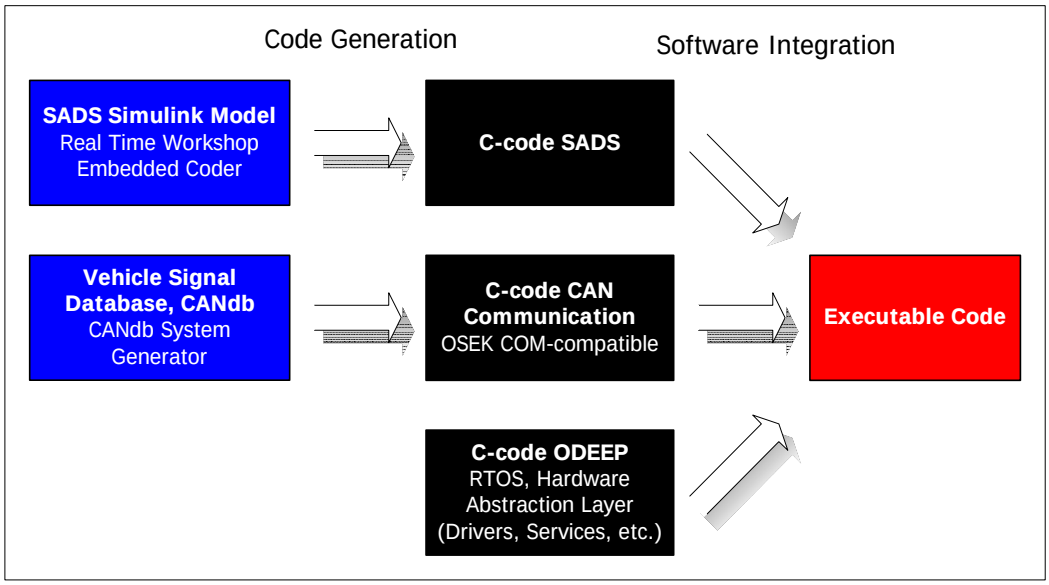


Figure 26 - Work methodology

7.1.6 Description of a semi-active damper system

The semi-active damper system is part of the chassis domain. It continuously adjusts the dampers in order to optimize the comfort and/or performance of the car, while also improving the safety level. The system consists of the following hardware components.

- An electronic control unit (ECU).
- Four electrically controllable shock absorbers / dampers.
- Three body accelerometers.

These components are defined as the item.

7.1.7 Functional Requirements

The functionality of the item is summarized below with a number of functional requirements.

- [**fr_1**] Roll, pitch and vertical motion of the vehicle shall be reduced.
- [**fr_2**] The braking distance shall be shortened.
- [**fr_3**] Two different suspension characteristics shall be offered (comfort & sport).
- [**fr_4**] The driver must be able to switch between the two modes at any time.
- [**fr_5**] In comfort mode, the driving comfort shall be improved.
- [**fr_6**] In sport mode, the vehicle performance shall be improved.

NOTE: Functional requirements should be expressed as measurable values. Numerical values have been removed from the example due to confidentiality.

7.1.8 Item Interface

7.1.8.1 Introduction

This section describes how the item interferes with other systems in the vehicle and how the item boundary is defined.

7.1.8.2 Item boundary

The SADS system has a number of electrical dependencies. Those that are not considered part of the item itself are listed below.

- Power supply – Power is needed for system operation.
- CAN bus – Critical input data is read from the CAN bus.

Furthermore, included in the item itself are the following dependencies.

- CDC dampers – The four damper actuators, one for each wheel.
- Body accelerometers – The three accelerometer sensors that provide important input data to the system.

7.1.8.3 Interfaces with other functions, systems, components or items

Many input signals are read from the CAN bus. This means that SADS, i.e. the item, is indirectly connected to, and therefore also depends on a number of other vehicle components. These components are listed below.

- Anti-lock Braking System Module (CAN).
- CAN Interface Module (CAN).
- Engine Control Module (CAN).
- Transmission Control Module (CAN).
- Yaw Rate Module (CAN).

7.1.8.4 Effects on other functions, systems components or items

- Not part of this analysis

7.1.8.5 Requirements on other items, and other items requirements

- Provide valid real-time data

7.1.8.6 The hazards that can affect the safety and reliability of the item

What could affect the safety and reliability of the SADS system is incorrect damper control. This includes both incorrect and total lack of compensation.

7.1.8.7 The driving situations and the operating conditions in which the item can initiate hazards

In driving situations where the vehicle chassis is exposed to significant movement, the SADS system could potentially initiate hazards. These situations are identified and listed in the next chapter.

7.2 Hazard and risk analysis

This application example is a hazard and risk analysis of a typical future semi-active damper system (SADS) for passenger cars. The analysis has been performed in accordance with ISO/WD 26262.

7.2.1 Failure modes

The failure modes are derived directly from section 7.1.8.6.

- [**fm_1**] Incorrect compensation.

The dampers are not controlled properly. Either all dampers or any individual damper can be very stiff or very hard, or anything in between.

7.2.2 Driving situations

As previously described, the relevant driving situations are those where the chassis movement is intense. For each such situation, an exposure factor is given and motivated.

- **[ds_1]** Heavy cornering with a speed above 50km/h.
  During heavy cornering, the vehicle tends to roll. If the dampers are too soft, this can result in under-steer or loss of stability. On the one hand, the average driver is not exposed to this situation very often, but on the other hand, a driver with an active driving style is exposed to this much more often. Therefore, a relatively high exposure rate is chosen.
  Exposure: **E3** (once a month or more often).
- **[ds_2]** Hard braking.
  During hard braking, the vehicle tends to pitch. In order to minimize the braking distance (and hence reduce risk of collision), the dampers need to be controlled actively. Normally, the average driver does not brake this hard very often.
  Exposure: **E2** (a few times a year).
- **[ds_3]** Very rough road.
  When driving in very rough road conditions, the vehicle might become unstable at certain speeds, unless the chassis is controlled actively. Rough roads are not that common, so the average driver is not exposed to this situation very often.
  Exposure: **E2** (a few times a year).

7.2.3 Hazardous situations

Some potentially hazardous situations can occur in the previously listed driving situations. Each hazardous situation is given a controllability factor.

- [**hs_1**] Loss of stability.
  In very tough conditions, the chassis stability might be lost due to the rolling and pitching tendencies of the car. However, the SADS system does not introduce any abrupt instability situations, but degrade rather gracefully. In addition, the passive wheel suspension is always present, no matter what happens to the dampers. Therefore, the situation is normally considered controllable.
  Controllability: **C2** (normally controllable)

- [**hs_2**] Lengthened braking distance.
  If the vehicle pitch is not compensated for properly, the braking distance might be lengthened. This is usually a rather controllable situation since the car behaviour is close to normal.
  Controllability: **C2** (normally controllable)

7.2.4 Hazardous events

The hazardous situations can result in a number of hazardous events, each which is given a certain severity level.

- [**he_1**] Collision.
  Collision is always a very serious situation that can lead to death, hence the high severity level.
  Severity: **S3** (life-threatening, survival uncertain)

7.2.5 Hazardous scenarios

From the information above, a number of scenarios can be identified. For each scenario, a safety goal is provided with a corresponding ASIL level.

7.2.5.1 Scenario 1

Incorrect compensation during heavy cornering leads to loss of stability, which leads to a collision.
[**ds_1**] x [**hs_1**] x [**he_1**] = **E3** x **C2** x **S3**
Safety goal: A malfunctioning SADS shall not cause loss of stability. **ASIL B**
Safe state: The driver shall be notified.

7.2.5.2 Scenario 2

Incorrect compensation during hard braking leads to loss of stability, which leads to a collision.
[**ds_2**] x [**hs_1**] x [**he_1**] = **E2** x **C2** x **S3**
Safety goal: A malfunctioning SADS shall not cause loss of stability **ASIL A**
Safe state: The driver shall be notified.

7.2.5.3 Scenario 3

Incorrect compensation during hard braking leads to lengthened braking distance, which leads to a collision.
[**ds_2**] x [**hs_2**] x [**he_1**] = **E2** x **C2** x **S3**
Safety goal: A malfunctioning SADS shall not lengthen the braking distance. **ASIL A**
Safe state: The driver shall be notified.

7.2.5.4 Scenario 4

Incorrect compensation during very rough road conditions leads to loss of stability, which leads to a collision.
[**ds_3**] x [**hs_1**] x [**he_1**] = **E2** x **C2** x **S3**
Safety goal: A malfunctioning SADS shall not cause loss of stability **ASIL A**
Safe state: The driver shall be notified.

7.2.6 What Causes Analysis

What causes analysis is only performed at scenario 1 since this scenario has the highest classification and is the only scenario that might imply fault tolerant processing of the control algorithms.

7.2.7 Scenario 1: ASIL B

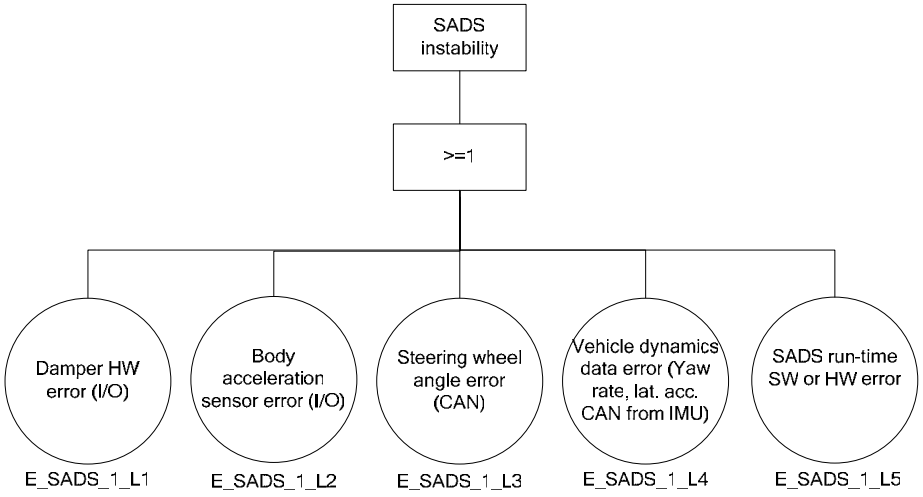


Figure 27: Fault tree, scenario 1: ASIL B

The fault tree shows some parts in the SADS-system that need special attention during design and where design requirements for increased safety have to be set up.

7.3 Functional Safety Concept

This application example is a specification of a functional safety concept of a typical future semi-active damper system (SADS) for passenger cars. The specification has been performed in accordance with ISO/WD 26262.

7.3.1 SADS input signal requirements

7.3.1.1 Body acceleration sensors

Requirement 1: LF sensor, ASIL B
This can probably be implemented by SW and monitoring within the SADS ECU.

Requirement 2: RF sensor, ASIL B
This can probably be implemented by SW and monitoring within the SADS ECU.

Requirement 3: RR sensor, ASIL B
This can probably be implemented by SW and monitoring within the SADS ECU.

7.3.1.2 Steering wheel sensors

Requirement 4: Hand Wheel Angle, ASIL B
It is assumed that the steering wheel sensors will send data that fulfills ASIL B via CAN.

Requirement 5: Wheel Angular Velocity, ASIL B
It is assumed that the steering wheel sensors will send data that fulfills ASIL B via CAN.

7.3.1.3 Vehicle dynamics sensors from IMU

Requirement 6: Yaw rate, ASIL B
It is assumed that the IMU will send data that fulfills ASIL B via CAN.

Requirement 7: Lateral acceleration, ASIL B
It is assumed that the IMU will send data that fulfills ASIL B via CAN.

7.3.2 SADS function implementation requirements

Requirement 8: Conceptual requirements Hardware/Software monitoring, ASIL B
Given the SADS criticality this requires redundancy in hardware/software. The validity of the critical input data and the calculation of the critical output data has to be monitored by independent software executed in another processor than the main function.

Requirement 9: Damping control signals calculation, ASIL B
The monitor shall verify the "damping control signals" before being used to control the dampers electrically. This is done by redundant reading of critical input signals and separate (simplified) signal calculations.

Requirement 10: Current feedback monitoring, ASIL B
The monitor shall verify the current feed back from the dampers is as expected.

Requirement 11: Transient filters
Before the system sets an error, transient filters shall be applied. This is to avoid transient disturbances that can cause limited availability and unnecessary service.

Requirement 12: Safety shall be prioritized
When tuning transient filters, safety shall have priority over availability.

Requirement 13: Healing
If safe healing can be done to improve availability it shall be done during the drive cycle. Else, healing shall be done at start-up of the next driving cycle.

Requirement 14: False healing
Faults with a certain frequency for long time shall not be detected as transients and then be falsely healed.

Requirement 15: Fail Silence
When the function processor that transmits information of CAN does not agree with the monitor (after transient filtering) the node shall be fail silent i.e. either not send any signals or send signals that the consumers can identify as erroneous.

7.3.3 SADS output signal requirements

Requirement 16: Driver information, ASIL A
Errors in the SADS functionally, i.e. loss of SADS, have to be shown to the driver.

7.4 System development

7.4.1 Specification of fault detection in a safety-related system

Fault detection must be implemented in all safety-related systems. It shall be implemented in each subsystem for memory, I/O, supply power and other internal parts of the subsystem. Fault detection shall also be implemented for functions of a subsystem of a safety-related system.

Techniques and measures to be used in checking of functionality are listed in table 4. Techniques and measures to be used in self-checking of internal parts within a subsystem are listed in table 5.

Requirements are specified in standards and may differ between different safety integrity levels (SIL or ASIL).

The checklists in table 4 and 5 may serve both as a support for specification of the design, and as a checklist for validation of techniques and measures implemented to control failures. Some of the listed aspects may not be relevant for all systems and may be listed as "not applicable" ("N.A.").

Table 4. Techniques and measures to detect failures in functions of subsystems

System:					
Target SIL/ASIL:					
Component	**Ref. IEC 61508-7**	**Req. ISO 26262-4**	**App / N.A.**	**Techniques used**	**DC**
Failure detection by on-line monitoring	A.1.1	Clause 6			
Comparator	A.1.3	“			
Majority voter	A.1.4	“			
Test by redundant hardware	A.2.1	“			
Dynamic principles	A.2.2	“			
Monitored redundancy	A.2.3	“			
Hardware with automatic check	A.2.6	“			
Analogue signal monitoring	A.2.7	“			

Table 5. Techniques and measures to detect failures within parts of a subsystem

System:					
Target SIL/ASIL:					
Component	**Req. IEC 61508-2**	**Req. ISO 26262-5**	**App / N.A.**	**Techniques used**	**DC**
Bus		Annex A			
CPU	Table A.4	“			
Interrupt		“			
Invariable memory	Table A.5	“			
Variable memory	Table A.6	“			
I/O units and interface	Table A.7	“			
Data paths	Table A.8	“			
Power supply	Table A.9	“			
Program sequence	Table A.10	“			
Ventilation and heating	Table A.11	“			
Clock	Table A.12	“			
Communication	Table A.9	“			
Mass storage	Table A.13	“			
Sensors	Table A.14	“			
Final elements	Table A.15	“			

8 Application examples, brake system

An electronic braking system is used to demonstrate some of the principles for development of safety related systems. The braking system is a simplified example which may not provide the functional safety required for a road vehicle. The examples of this chapter are not sufficient to claim a safe and reliable automotive system. Neither can the conclusions for this braking system be regarded valid for all other types of braking systems.

8.1 Preliminary Safety Analysis

This example is based on the draft approach suggested by MISRA for Preliminary Safety Analysis. The objective of the PSA process is to identify the hazards that may be associated with the functionality of a system, to quantify the rigour that must be employed in engineering that system, and derive its High-Level Safety Requirements [MISRA, chapter 3.5].

PSA is a qualitative process, beginning with a model of the system, together with its functional description, which is subsequently analysed to deduce any hazards associated with the functionality and the outputs. Each hazard is assigned a Risk Classification, from which a SIL is formulated for the system as a whole. In addition to the Risk Classification, a set of High-Level System Safety Requirements is derived from the analysis results, and this forms one of the outputs of the PSA, and inputs to the DSA (Detailed Safety Analysis).

Input:	Safety Envelope
	Safety Policy
	Overall System Requirements and Environment
Output:	Refined Safety Envelope
	High-Level Safety Requirements
	Safety Integrity Level
	Hazard List
	Safety Argument
	Project Safety Plan
	System Model

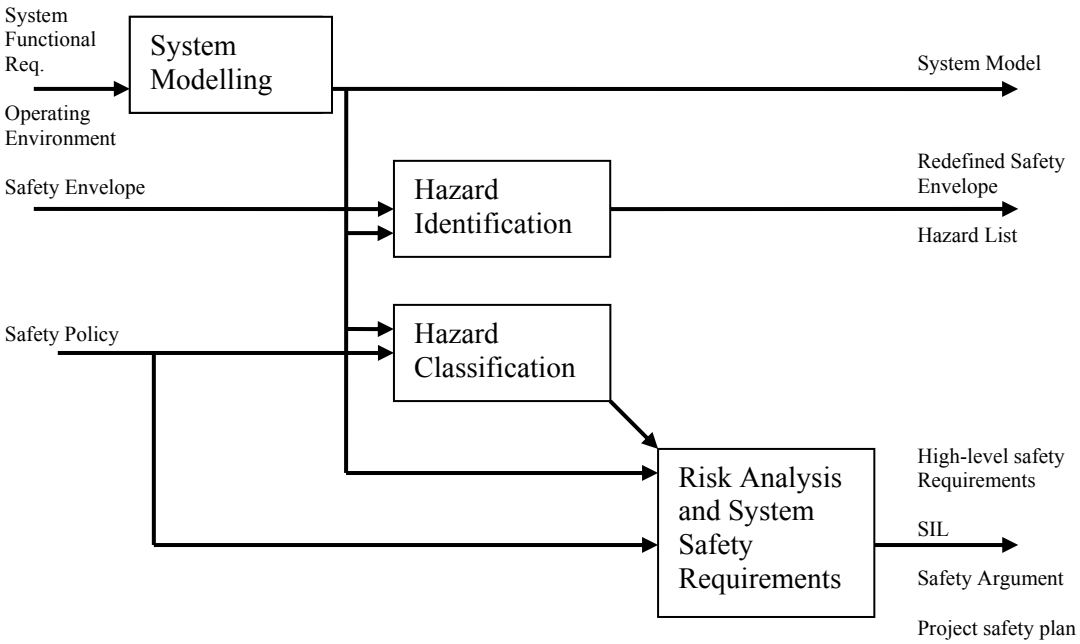


Figure 28. The Preliminary Safety Analysis Process [MISRA]

The following chapters are intended to define a set of folders in a PSA project binder. Each of the folders describes an activity, and defines input and output documents. Major chapter numbers are related to activities, and minor numbers to documents within the activities. The documents are described with template forms. Checklists for the activities provide guidelines on how to produce the outputs.

Change tracking is implemented by having a revision history of each document and a unique issue number (integer sequence). For each document it is specified which issue of the input documents that have been used. All issues of each document are to be stored in the binder.

The table below summarizes the major (mandatory) document outputs. The documents have a standard format, and are stored electronically in files with name: [DOCUMENT_NAME]_issue[ISSUE_NUMBER].doc. Printout of each issue of each document is to be stored in the project binder.

Activity	Input	Output
System modelling	PSA_SYSTEM_REQUIREMENTS PSA_OPERATING_ENVIRONMENT	PSA_SYSTEM_MODEL
Hazard Identification	PSA_SYSTEM_MODEL PSA_SAFETY_ENVELOPE	PSA_HAZARD_LIST PSA_SAFETY_ENVELOPE
Hazard Classification	PSA_SYSTEM_MODEL PSA_SAFETY_ENVELOPE PSA_SAFETY_POLICY PSA_HAZARD_LIST	PSA_HAZARD_CLASSIFICATION
Risk Analysis	PSA_SYSTEM_MODEL PSA_SAFETY_ENVELOPE PSA_SAFETY_POLICY PSA_HAZARD_LIST PSA_HAZARD_CLASSIFICATION	PSA_RISK_ANALYSIS
Safety requirements allocation	PSA_SYSTEM_MODEL PSA_SAFETY_ENVELOPE PSA_SAFETY_POLICY PSA_HAZARD_LIST PSA_HAZARD_CLASSIFICATION PSA_RISK_ANALYSIS	PSA_SAFETY_REQUIREMENTS

8.1.1 Input to PSA

These folders contain input documents that are used in the PSA. They are produced in activities preceeding the PSA, but may be revised (refined) in the PSA activities.

Activity	Input	Output
		PSA_SYSTEM_REQUIREMENTS PSA_OPERATING_ENVIRONMENT PSA_SAFETY_ENVELOPE PSA_SAFETY_POLICY

8.1.2 Safety Policy

The safety policy comprises a set of statements of commitment and intent from the board of directors of a company or organisation. The statements should define, at high level, the fundamental aims and responsibilities for managing product and process safety issues [MISRA chapter 4.3].

Example Document for PSA_SAFETY_POLICY

Omitted in this example.

8.1.3 Safety Envelope

The Safety Envelope defines the ‘expected’ or ’intended’ use of the vehicle with the system fitted and operational, so that the risks associated with the system can be evaluated under defined conditions, should it fail to perform an intended function correctly [MISRA, chapter 4.5].

Example Document: PSA_SAFETY_ENVELOPE

The following safety envelope identifies a consistent set of conditions that will be assumed when analyzing the system.

- The vehicle is considered to be equipped with all systems, which have electrical interfaces to the brake system (maximum configuration) and are of interest for the analysis.
  - Engine control unit
  - Automatic gearbox control unit
  - Instrument control unit
  - Light control unit
  - Suspension control unit
  - Vehicle control unit

- The vehicle is being used on highways

- The vehicle is being operated in a manner that conforms with the drivers understanding of how it should be used.

- The vehicle is not defective in any way other than the issues identified in the safety analysis.

- The vehicle is in a legal and roadworthy condition.

- The vehicle is being operated in a manner consistent with its type.

- The weather conditions are assumed to be average Nordic conditions.

- The vehicle is being driven in a manner that does not significantly amplify the consequences of a hazard, or provokes a hazard on its own.

8.1.4 Overall system requirements

Example Document: PSA_SYSTEM_REQUIREMENTS

Functional Requirement	**Description**
Service brake	The system provides the service brake function commanded by pedal by the driver.
Interface	The system shall interact with the vehicle HMI (human machine interface) in order to control the braking of the vehicle and to inform other systems and / or the driver regarding status of the brake system.

Functional Requirement	**Description**
Brake control	The system shall perform the control of the braking function of the vehicle. The brake application versus pedal stroke shall be adjustable by tuning of software parameters in order to meet the customer demand of different “brake feelings” on different vehicle platforms. The system shall be configurable to brake the vehicle with deceleration control or with clamp force control using the same input signal.
Slip control	The wheel brake shall be able to control the wheel slip with an algorithm based upon the wheel speed information and a calculated vehicle reference speed.
Load proportioning	The system shall distribute brake force according to the axle load of the vehicle. The level of load proportioning can be based on external signals or fully by a calculated value.
Parking brake	The system shall include a parking brake function.
Battery management	The system shall know the status of the batteries either by use of an external subsystem or by integrated functionality. If the system detects a low power condition the system shall be able to set itself to a safe default state.
External brake command	The system must have an input for an external brake demand.
Diagnostics	The system shall interact with the vehicle diagnostic system over standard serial bus or / and be able to control a standard driver interface with warning / error lamps.

Non-functional Requirement	**Description**
Brake response	The delay time from the initiation of a brake application until the vehicle starts to brake shall be less than x ms. The brake torque shall be controlled with the accuracy of $\pm$ x Nm. The minimum brake torque (threshold value) shall be less than x Nm. The brake torque application rate shall be controlled within 0–x kNm/sec.
Failure handling	Any single failure in the system may not affect the system more than it reduces the brake performance of the vehicle by max. 25%. All brakes shall if possible be released if required. If simultaneous failures occur in the system they shall not affect the system except reducing brake performance of the vehicle by max. 50%. Any multiple faults should be handled in the safest way possible by the system. An alternative activation of the brakes is in this case allowed, i.e. if a brake pedal activation results in no braking, use of the parking activation is allowed.
Lifecycle	The expected life of the brake system without any maintenance shall be more than x years or more than x00 000 km.
Environment	The brake system shall operate in ambient temperature from -45°C (snow and ice) to +125 °C. Altitude up to 2500m. Vibrations to a level that could be expected in a normal installation shall have no effect on the brake system.
Regulations	The brake system shall comply with regulations ECE-R13. (71/320/EEG)

8.1.5 Operating environment

Example Document: PSA_OPERATING_ENVIRONMENT

Omitted in this example.

8.1.6 System modeling

The objective of system modeling is to produce an abstraction, or model, of the system. This will be used as a base for the analysis [MISRA, chapter 5.1].

The objective of the concept phase of the overall safety lifecycle is to develop a level of understanding of the EUC and its environment (physical, legislative etc.) sufficient to enable the other safety lifecycle activities to be satisfactorily carried out [IEC 61508, clause 7.2.1].

The objective of the overall scope definition phase is to determine the boundary of the EUC and the EUC control system; and to specify the scope of the hazard and risk analysis (for example process hazards, environmental hazards). [IEC 61508, clause 7.3.1].

By its very nature a model must be an abstraction, since the only "model" that can be identical to the system is the system itself. It is therefore necessary to choose the abstractionso that it highlights the features necessary for the particular task. Although some definitions of a hazard limit themselves to physical situations with a potential for human injury, in practice one should include all situations that can threaten people, property and the natural environment; thus any model that is used as a target for a PSA and/or a PHA should highlight the boundary between the system under consideration and those enitities that might suffer from the hazard and the interaction between them.

Activity	**Input**	**Output**
System modelling	PSA_SYSTEM_REQUIREMENTS PSA_OPERATING_ENVIRONMENT	PSA_SYSTEM_MODEL

Example Document: PSA_SYSTEM_MODEL

System components

Component	**Description**
HMI	Driver interface (excluding pedal): warning/indicator lamps; switches (including ignition key)
Pedal	The pedal assembly is a unit with a mechanical foot-pedal with sensors for pedal stroke (and force). The pedal unit has only electrical interface to the brake system.
ABS sensors	Sensors for wheel rotational speed, mounted on each wheel carrier.
Power source	Two independent batteries with independent charge units. Batteries may also supply other systems. Therefore, the batteries are outside the zone of responsibility for HBP.
System communication interface	CAN sockets for connection to external vehicle systems.
Communication system	Physical bus(es) to transfer information internally within the brake system.
Brake units	Electromechanical sliding-calliper brake actuator (EBA) with DC-motor, spring-actuated parking-brake, sensors (force, position), embedded microcontrollers and closed-loop brake control.
Brake ECU	Central electronic control unit.

System boundaries

The system components and system boundaries are illustrated in Figure 29. Note that this figure only shows logical connections without any details of the system realization. The batteries are excluded from the zone of responsibility since they are likely to be part of several other subsystems (beside the brake system). It will therefore fall within the responsibility of the vehicle OEM. The *target of evaluation* (TOE) includes only parts of the brake units since this safety analysis focuses on system properties. The electrical/logical parts are included while the mechanical parts are excluded.

System interaction model

The interaction of the system (TOE) with its environment is illustrated in Figure 30. The terminator in the brake units is defined to be the drive shaft that transfers torque to the gear. The corresponding boundary elements are, the DC-motor and the electro-magnet that holds the parking-brake spring. Figure 31 is a passport diagram, which shows how the boundary elements are transferring data (in a wide sense) to/from the interior of the system.

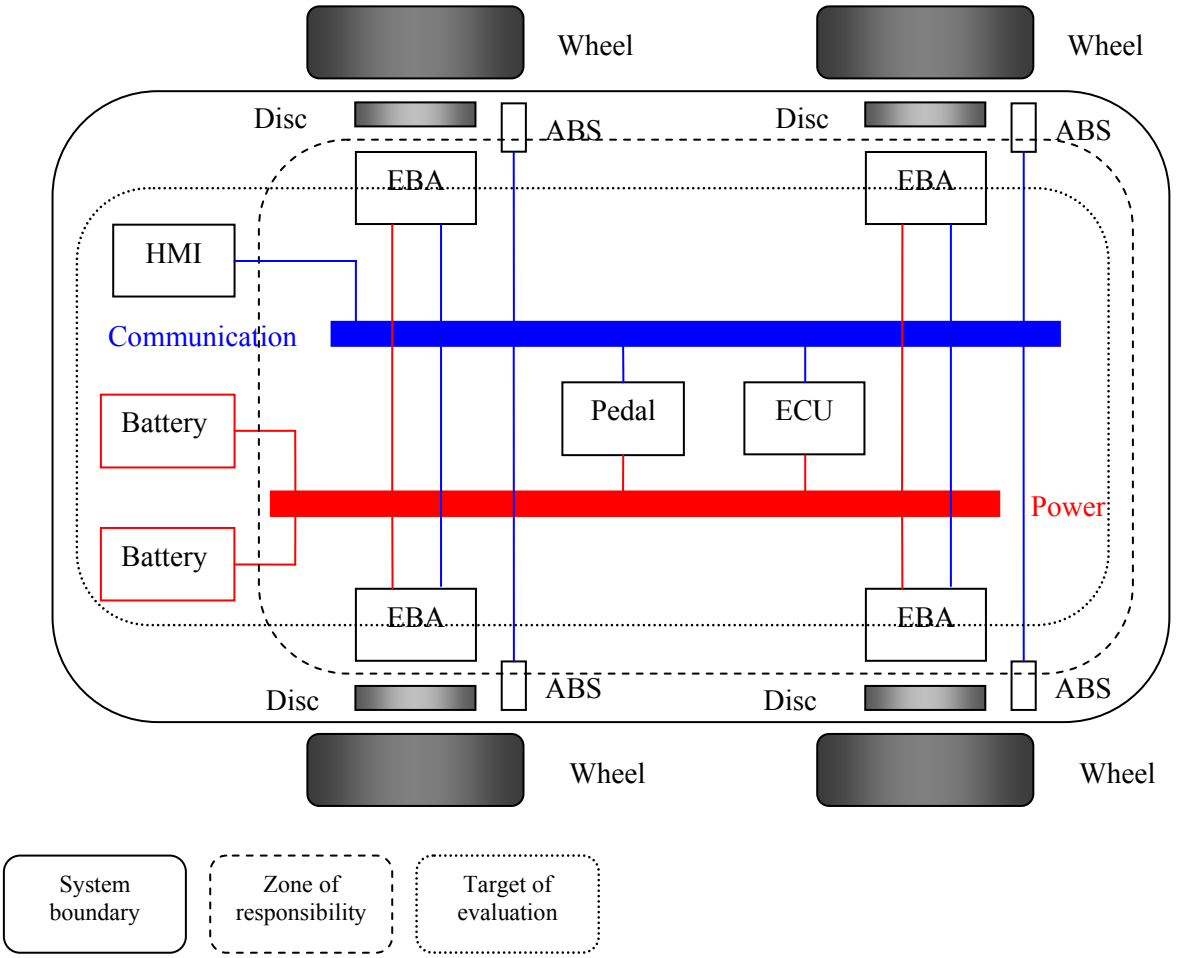


Figure 29. System boundaries. Note that power and communication systems are illustrated schematically. The details on implementation and configuration (redundancy, bus types, etc) is left open at this stage.

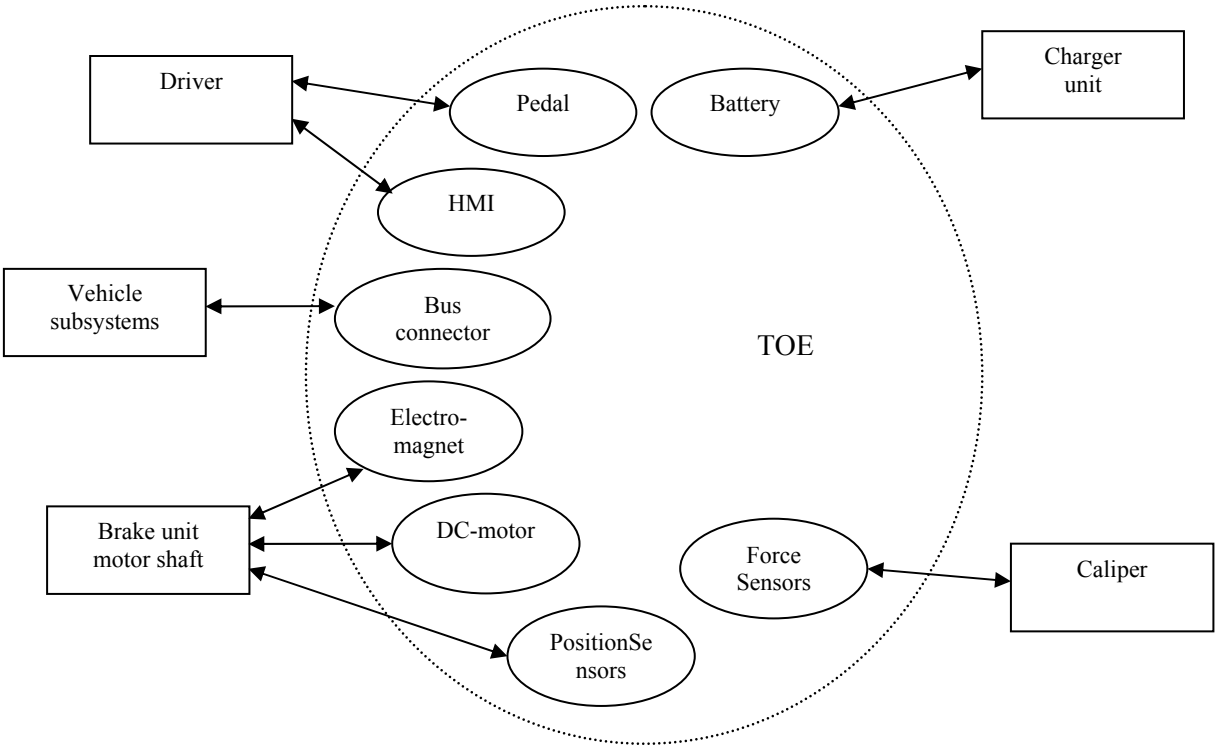


Figure 30. System interaction. *Boundary elements* within the target of evaluation interact with *terminators* outside the TOE boundary.

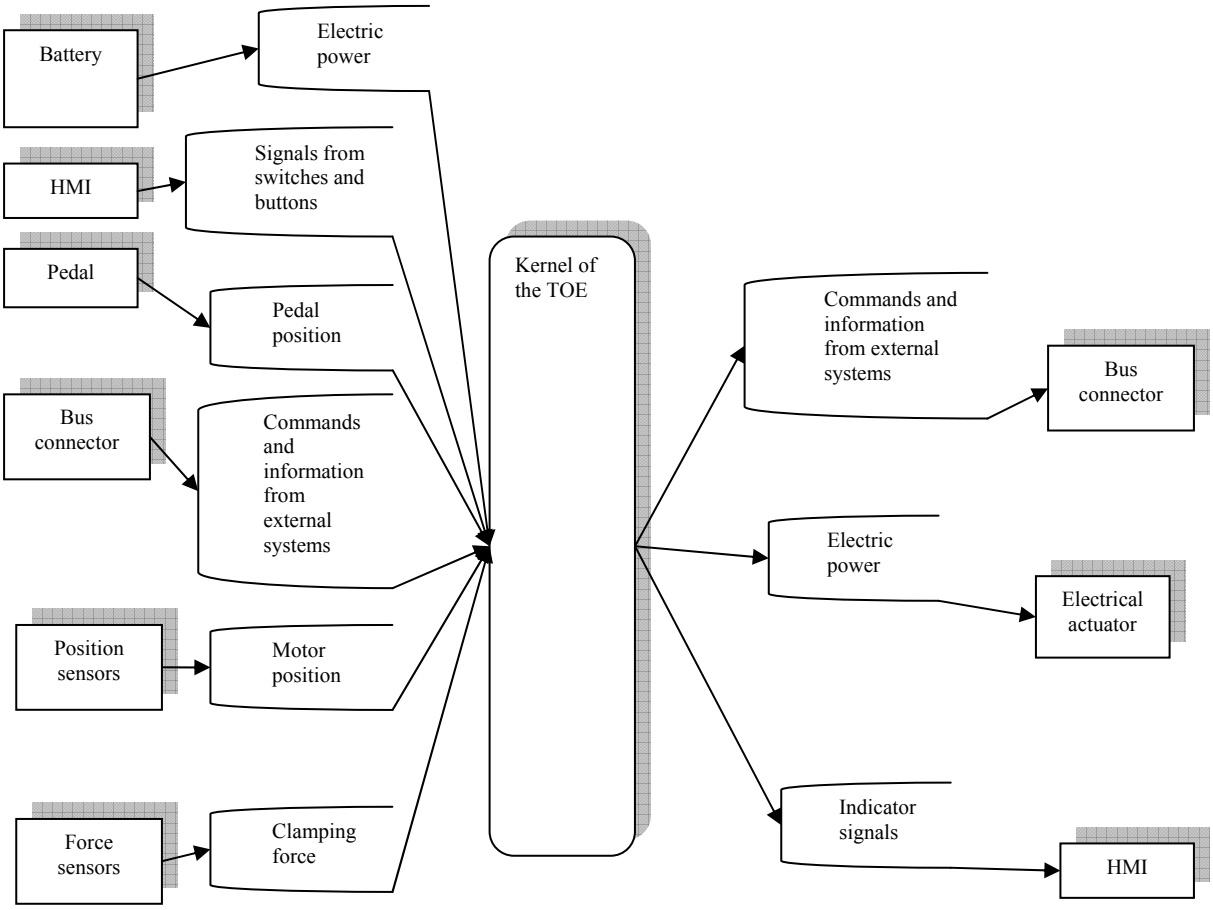


Figure 31. Passport diagram of system interaction with environment.

8.1.7 Hazard Identification

The objective of Hazard Identification is to identify all the hazards associated with the system being analysed to a sufficiently high degree of confidence. [MISRA chapter 6.1]

Activity	**Input**	**Output**
Hazard Identification	PSA_SYSTEM_MODEL PSA_SAFETY_ENVELOPE	PSA_HAZARD_LIST PSA_SAFETY_ENVELOPE

Example Document: PSA_HAZARD_LIST

Hazard ID	**Hazard Name**	**Description/Comments**
H1	Undemanded brake application	The brakes (one or multiple) are (suddenly, spontaneously) applied without command from driver or external system. Worst case: uncontrollable wheel lock, both unsymmetrical and symmetrical.
H2	Dragging brake	Brakes (one or multiple) are constantly applied (with small to moderate force)
H3.0	HMI malfunction	Malfunction of dashboard indicators and switches. (Hazards for switches to be added.) Split into sub-hazards.
H3.1	False fault indication	Indicates a fault although the system is working.
H3.2	Absent fault indication.	No fault is indicated although the system will not respond as expected. Worst case: no response to pedal push.
H4.0	Brake light malfunction	Split into sub-hazards
H4.1	False brake-light	Brake light is lit although no retardation. Could be confusing to fellow road users.
H4.2	Absent brake-light	No brake lights when brake pedal is pushed. Will not warn fellow road users.
H5	No pedal response	No response (brake actuation) when pedal is pushed.
H6	Unexpected pedal response	Unexpected response (brake application), e.g. lateral imbalance or incorrect level of retardation.
H7.0	Interface malfunction	Erroneous information communicated to/from external systems. Split into sub-hazards.
H7.1	Interface malfunction in	Interface error for incoming information. Worst case: external brake commands.
H7.2	Interface malfunction out	Interface error for outgoing information. No information to surrounding systems that the brake pedal is pushed and the brake is active.
H8	ABS malfunction	ABS does not engage properly. Wheel-lock at (large) brake applications possible.
H9.0	Parking-brake malfunction	Split into sub-hazards.
H9.1	Parking-brake uncommanded release	Uncommanded release. Worst case: the vehicle is parked in a steep decline.
H9.2	Parking-brake stuck	Stuck. Difficulties when releasing parking brake.
H9.3	Parking-brake engaged when moving	Engaged when moving. One or more wheels will be locked. Difficult and time consuming to release.

8.1.8 Hazard Classification

The outcome of the classification is a measure of the risks associated with a hazardous event. [MISRA chapter 7.1]

Activity	**Input**	**Output**
Hazard Classification	PSA_SYSTEM_MODEL PSA_SAFETY_ENVELOPE PSA_SAFETY_POLICY PSA_HAZARD_LIST	PSA_HAZARD_CLASSIFICATION

PSA_HAZARD_CLASSIFICATION

Hazard ID	**Hazard Name**	**Interdependency**	**Degree of control**	**Provisional back up**	**Reaction time**	**Controllability / Possibility to avoid**
H1	Undemanded brake application	D	A	A	A	C4
H2	Dragging brake	E	C	C	E	C2
H3.0	HMI malfunction	-	-	-	-	-
H3.1	False fault indication	D	D	E	D	C1
H3.2	Absent fault indication.	D	C	C	C	C2
H4.0	Brake light malfunction	-	-	-	-	-
H4.1	False brake-light	E	E	E	E	C0
H4.2	Absent brake-light	C	D	D	E	C2
H5	No pedal response	E	C	C	B	C3
H6	Unexpected pedal response	E	D	C	B	C3
H7.0	Interface malfunction	-	-	-	-	-
H7.1	Interface malfunction in	B	A	C	C	C3?
H7.2	Interface malfunction out	B	C	C	C	C2?
H8	ABS malfunction	B	C	C	B	C3
H9.0	Parking-brake malfunction	-	-	-	-	-
H9.1	Parking-brake uncommanded release	E	A	A	A	P2
H9.2	Parking-brake stuck	E	E	C	E	P1/C0
H9.3	Parking-brake engaged when moving	E	B	A	A	C4

8.1.9 Risk Analysis and System Safety Requirements

The objective of the hazard and risk analysis is
- to determine the hazards and hazardous events of the EUC and the EUC control system (in all modes of operation), for all reasonably foreseeable circumstances including fault conditions and misuse;
- to determine the event sequences leading to the hazardous events;
- to determine the EUC risks associated with the hazardous events.
[IEC 61508-1, clause 7.4.1]

The objective of the safety requirements allocation phase is
- to allocate the safety functions, contained in the specification for the overall safety requirements (both the safety functions requirements and the safety integrity requirements), to the designated E/E/PE safety related systems, other technology safety-related systems and external risk reduction facilities;
- to allocate a safety integrity level to each safety function.
[IEC 61508-1, clause 7.6.1]

Activity	**Input**	**Output**
Risk Analysis	PSA_SYSTEM_MODEL PSA_SAFETY_ENVELOPE PSA_SAFETY_POLICY PSA_HAZARD_LIST PSA_HAZARD_CLASSIFICATION	PSA_RISK_ANALYSIS
Safety requirements allocation	PSA_SYSTEM_MODEL PSA_SAFETY_ENVELOPE PSA_SAFETY_POLICY PSA_HAZARD_LIST PSA_HAZARD_CLASSIFICATION PSA_RISK_ANALYSIS	PSA_SAFETY_REQUIREMENTS

Example Document: PSA_RISK_ANALYSIS

Hazard ID	**Hazard Name**	**Severity**	**Frequency**	**Controllability / Possibility to avoid**	**Risk**
H1	Undemanded brake application	S2	F2?	C4	R4
H2	Dragging brake	S1	F1	C2	R1
H3.0	HMI malfunction	-	-	-	-
H3.1	False fault indication	S1	F2	C1	R1
H3.2	Absent fault indication.	S2	F2	C2	R2
H4.0	Brake light malfunction	-	-	-	-
H4.1	False brake-light	S1	F2	C0	R1
H4.2	Absent brake-light	S1	F2	C2	R1
H5	No pedal response	S2	F2	C3	R3
H6	Unexpected pedal response	S1	F1	C3	R1
H7.0	Interface malfunction	-	-	-	-
H7.1	Interface malfunction in	S1	F2	C3	R1
H7.2	Interface malfunction out	S1	F2	C2	R1

H8	ABS malfunction	S1	F1	C3	R1
H9.0	Parking-brake malfunction	-	-	-	-
H9.1	Parking-brake uncommanded release	S2	F1	P2	R3
H9.2	Parking-brake stuck	S1	F1	P1/C0	R1
H9.3	Parking-brake engaged when moving	S2	F2	C4	R4

Document name: PSA_SAFETY_INTEGRITY_REQUIREMENTS

Hazard ID	Hazard Name	Risk	Random integrity requirement	Systematic integrity requirement	Risk class
H1	Undemanded brake application	R4	[1e-8,1e-7]	SIL3	
H2	Dragging brake	R1		SIL1	
H3.0	HMI malfunction	-	-	-	-
H3.1	False fault indication	R1		SIL1	
H3.2	Absent fault indication.	R2		SIL1	
H4.0	Brake light malfunction	-	-	-	-
H4.1	False brake-light	R1		SIL1	
H4.2	Absent brake-light	R1		SIL1	
H5	No pedal response	R3		SIL2	
H6	Unexpected pedal response	R1		SIL1	
H7.0	Interface malfunction	-	-	-	-
H7.1	Interface malfunction in	R1		SIL1	
H7.2	Interface malfunction out	R1		SIL1	
H8	ABS malfunction	R1		SIL1	
H9.0	Parking-brake malfunction	-	-	-	-
H9.1	Parking-brake uncommanded release	R3		SIL2	
H9.2	Parking-brake stuck	R1		SIL1	
H9.3	Parking-brake engaged when moving	R4		SIL3	

8.1.10 Safety Argument

Omitted in this example.

8.1.11 Project Safety Plan

Omitted in this example.

8.1.12 Safety Case

The following suggested template may be used for a safety case:

1 Background to the documentation

2 Prerequisites
- Safety Policy
- Project Safety Plan

3 System Model

4 Safety Envelope

5 Identified Hazards
- Hazards with classification

6 System Analysis
6.1 FTA
6.2 FMEA

7 Requirements
7.1 Safety Requirements
- Requirements for risk mitigation functions
7.2 Safety Integrity Requirements
- Requirements for SIL and failure probability

8 Residual Risk and Risk Assessment
- Discussion of outcome of risk/hazard mitigation

9 Safety Case
- Safety Argument (link between analysis and requirements)

10 Statement of Compliance
- Summary of previous information:
- Safety Policy
- Project Safety Plan
- Definition of broadly acceptable risk
- Hazards
- Safety requirements
- SIL assignment

8.2 Hardware development

8.2.1 The application of reliability block diagrams (RBD) for hardware reliability analysis

There are two available approaches when designing an RBD, either to map the physical hardware that realizes the function to an RBD or to map the functions of the system to a RBD. It is up to the application and person that carries out the analysis to decide which approach to use. However, it is important to keep the RBD understandable and only include relevant parts of the system. In the following sub-clause an example is shown where RBD is applied to a functional level of the described system.

In this example there are no limitations considered in the applied fault model, so the example does not e.g. only addresses single point faults or multiple faults. Such limitations may affect the design of a reliability block diagram.

8.2.1.1 The electronic brake control system

This example illustrates an electronic brake control system according to figure 32.

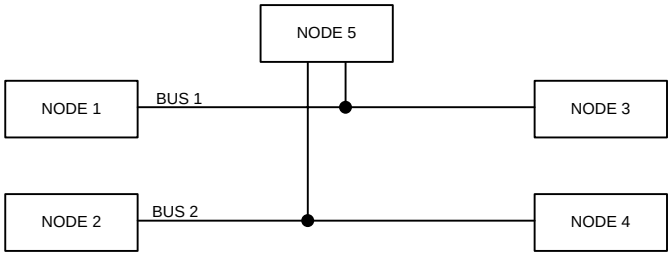


Figure 32: Simplified block diagram of an automotive brake control system

The block diagram in figure 32 represents a distributed electronic control system where nodes 1-4 are used for actuating the brakes and node 5 is an ECU (i.e. a node with higher performance). The ECU may communicate with the nodes through two physically separated busses, one for each node pair. No hydraulic or mechanical components are included.

A detailed FMEA has previously been performed for all nodes in the system. The system contains several functions (safety functions) intended to prevent faults from propagating which may result in a dangerous situation. One possible approach is to design an RBD for each safety related function in order to compose the total reliability of the system. This example considers only one safety function and is not completed.

Function to be considered: Brake actuation
Case: The driver pushes the brake pedal in order to slightly reduce the speed of the vehicle.
Failure mode: A single fault has occurred (or occurs) in the power electronics that controls the brake force actuated by node 1, which causes another brake application than demanded from the ECU.
Required behaviour at fault: Adapt the brake force on node 2 in order to reduce the fault influence on the steering

The following RBD design serves only as an example that may be performed in different ways and would be more detailed in a real analysis:

(1) If the system in figure 33 has no fault detecting/handling functions employed the RBD for this particular failure mode becomes:

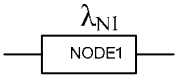


Figure 33. RBD of erroneous brake actuation in a single electronic node

This implies that the system is solely dependent upon the reliability of that node. When the node fails, with λ_{N1} , the unwanted failure consequence will occur.

(2) In this example is it however required that a function is added which can detect and handle this particular failure. This function may be modelled as:

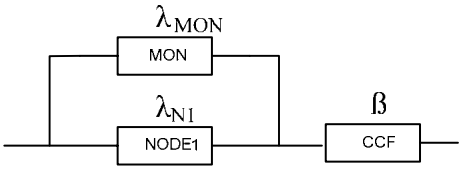


Figure 34. RBD of the target failure and the monitoring function

In order to detect and handle a failure a redundant function is needed which is included in the RBD as the block MON. A redundant function always adds a probability for common cause failures (both functions fail simultaneously due to the same cause) which is modelled as the CCF block. This RBD should be interpreted as: The specified functionality of this system will be kept if either the brake actuating function (NODE 1) or the monitoring function (MON) operates correctly and no common cause failure occurs.

(3) There are no physically separated functions in this example system so the monitoring function (MON) is integrated in the distributed control system. The required behaviour at fault is that the other brake actuator (NODE 2) compensates for this failure. In order to detect the failure the ECU must be able to correctly interpret the erroneous response from the faulty brake controller via bus 1. In order to handle the failure the ECU (NODE 5) must correctly recalculate and transmit the adjusted brake demand through bus 2 and the brake controller (NODE 2) must correctly apply the new brake demand. The monitoring function may therefore be modelled as:

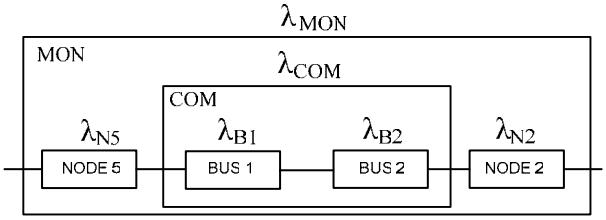


Figure 35. RBD for the redundant monitoring function (in this example)

For simplification the two bus contributions may be composed into one block and thus the final RBD is derived for this example according to:

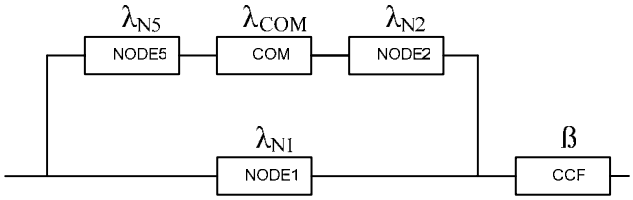


Figure 36: Final RBD for this example

When solving the RBD for the combined model the series and parallel formulas for two element systems are used. The variables used in the calculations address implicitly dangerous faults which were identified in an assumed previously performed analysis such as an FMEA.

For the fault detecting/handling part of the RBD:
All blocks in series: $R_{N5}R_{COM}R_{N2}$ which are connected in parallel with node 1:
$R_{N1} + (R_{N5}R_{COM}R_{N2}) - R_{N1}(R_{N5}R_{COM}R_{N2})$ This composes the control system that is finally connected in series with the common cause block:
$R_{SYS} = R_{CCF}(R_{N1} + (R_{N5}R_{COM}R_{N2}) - R_{N1}(R_{N5}R_{COM}R_{N2}))$

If all included failure rates are assumed to be constant the reliability function may be expressed as (if a non constant failure rate is used the following part of this example is not applicable):
$R_{SYS} = e^{-\beta\lambda_{N1}t}(e^{-\lambda_{N1}t} + e^{-(\lambda_{N5}+\lambda_{COM}+\lambda_{N2})t} - e^{-(\lambda_{N1}+\lambda_{N5}+\lambda_{COM}+\lambda_{N2})t})$

$R_{SYS} = e^{-(\lambda_{N1}+\beta\lambda_{N1})t} + e^{-(\beta\lambda_{N1}+\lambda_{N5}+\lambda_{COM}+\lambda_{N2})t} - e^{-(\beta\lambda_{N1}+\lambda_{N1}+\lambda_{N5}+\lambda_{COM}+\lambda_{N2})t}$

By using the reliability function and the formula for MTTF the average probability of dangerous failure per hour may be approximated according to (PFH = 1/MTTF):

$$PFH_{SYS} = \left(\frac{1}{\lambda_{N1}+\beta\lambda_{N1}} + \frac{1}{\beta\lambda_{N1}+\lambda_{N5}+\lambda_{COM}+\lambda_{N2}} - \frac{1}{\beta\lambda_{N1}+\lambda_{N1}+\lambda_{N5}+\lambda_{COM}+\lambda_{N2}}\right)^{-1}$$

For simplicity the diagnostic coverage and MTTR are not considered in this example, any dangerous and detected fault are assumed to be instantly handled by the control system at any time of occurrence. The vehicle is in that case always put into a safe state with no time delay.

8.2.2 The application of Markov models for hardware reliability analysis

Figure 37 below describes a possible hardware realization of an assumed brake pedal sensing system which is a redundant two-channel system

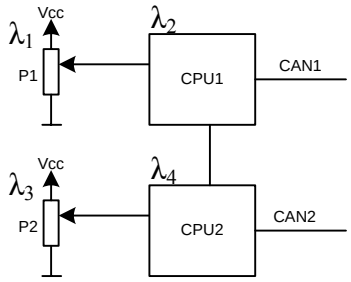


Figure 37. Brake pedal sensor system

The first step in the detailed analysis (in this example) is to perform an FMEA. The analysis is performed by considering one single channel (i.e. without monitoring functions) in figure 37. This is described in part 1 of the FMEA sheet below. Each fault effect is divided into a safe fraction and a dangerous fraction. All failure rates used are assumed to be constant.

A software analysis has been performed and the following diagnostic function used for monitoring the processor and the potentiometer has been identified (notice that the communication system is considered as integrated in the CPUs in this example):

Monotoring function	Description
A- Comparison between redundant CPUs	The function continuously compares the potentiometer feed-back position values, performs control-flow tests of the CPUs and compares the communication channels. Any deviation between the redundant channels is handled by a special algorithm which forces the most incorrect channel into a passive safe-state.

Part 1 – Without considering monitoring functions							Part 2 – Taking the monitoring functions into account	
Comp.	Mode	Rate [FITs]	Distr. [%]	Effect	S [%]	D [%]	Monitoring function.	Coverage [%]
P1	SC	[700]	0.5	Either stucked at max or min position, or reduced range of the potentiometer	10	90	A	90
	OC		25	Floating feedback voltage	50	50	A	90
	D		65	Indicating the wrong position – continuously	30	70	A	90
	F		9.5	Indicating the wrong position - instantaneously	10	90	A	90
CPU1	F	[1300]	100	Indicating the wrong position	50	50	A	90

A Markov model may be produced out of the system functional specification:

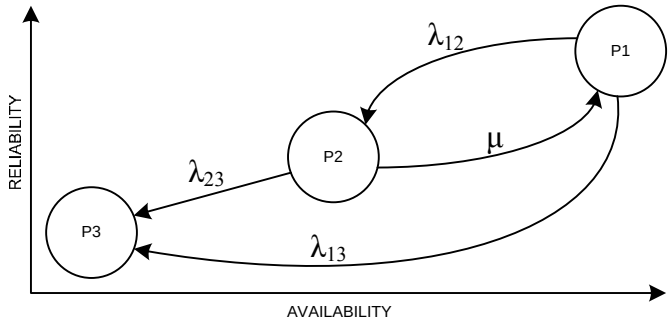


Figure 38. Potential Markov model of a two sensor system

Markov diagram state description	
P1	Normal operation. The brake pedal functionality corresponds with the specification.
P2	Degraded operation (safe-state). A fault has been detected by the system which only operates on one of the two channels. The driver is notified about the fault and is urged to drive to the closest garage in order to repair the system.
P3	Dangerous operation. A fault has occurred which severely affects the brake pedal response and which may have consequences in a dangerous operation.

By using the FMEA result the transition rates may be determined according to:

Markov diagram transition description	
λ_{12}	The failure rate of all single safe or dangerous faults that may occur in both channels and which are detected by the diagnostic function is: 2*0.9*(700 + 1300) = **3600 FITs**
λ_{13}	The failure rate of all dangerous single faults and common cause faults that are not detected by the diagnostic function is (assume β = 0.01): 2*(1-0.9)*((0.05*0.9+0.25*0.5+0.65*0.7+0.095*0.9)*700 + 0.5*1300) + 0.01*0.9*((0.05*0.9+0.25*0.5+0.65*0.7+0.095*0.9)*700 + 0.5*1300) = **376 FITs**
λ_{23}	The failure rate of all dangerous single faults that may occur in the active channel when the system remains in safe-state is: ((0.05*0.9+0.25*0.5+0.65*0.7+0.095*0.9)*700 + 0.5*1300) = **1147 FITs**
μ	The probability per hour that the system becomes repaired when the safe state is entered is (assuming the worst case when the driver ignores the fault until the annual service and approximating the repair time = 0): 1/((365*24)/2+0) = **228E-6**

After this it is possible to state the mathematical expression of the Markov model where the transition matrix becomes:

$$T = \begin{bmatrix} 1-(\lambda_{12}+\lambda_{13}) & \lambda_{12} & \lambda_{13} \\ \mu & 1-(\mu+\lambda_{23}) & \lambda_{23} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} (1-3976\cdot 10^{-9}) & 3600\cdot 10^{-9} & 376\cdot 10^{-9} \\ 228000\cdot 10^{-9} & (1-229147\cdot 10^{-9}) & 1147\cdot 10^{-9} \\ 0 & 0 & 1 \end{bmatrix}$$

The main task in this example is to evaluate the average probability to reach the state P3 per hour (PFH_d). A computer aided tool has been used for processing the above T matrix according to the discrete time method in order to obtain the $MTTF_d$. The result is:

$MTTF_d = 2.5778 \cdot 10^6$ hours

The average probability of dangerous failure per hour then becomes: $PFH_d = 3.87 \cdot 10^{-7}$

This example is severely simplified and is only intended for demonstrating the principles for the reliability analysis procedure with Markov modelling.

8.2.3 Validation of memory checking

Techniques and measures should be implemented to control failures during operation. The checklist in this example may be used when validating techniques for memory checking.

Aim: To validate the memory checking, and to calculate its coverage.

Reference: IEC 61508-2, table A.5. ISO/WD 26262-5, annex A.

Description: Programme code and constants will be stored in invariable memory (ROM, EPROM etc.). A memory map will give an overview of the memory ranges. The hardware circuit diagram or the microprocessor manual may have to be studied. Software routines for memory checking and handling of memory faults shall be analysed. The analysis can be summarised in the following check list:

	Invariable Memory Ranges				
		Ap.*)	Yes	No	Comment
A	Is automatic and periodic checking of the invariable memory range made at a periodic interval?				
B	Is automatic checking of the invariable memory range made at power-up?				
C	Have techniques and measures according to IEC 61508-2, table A.5 been used?				
D	Have techniques and measures according to ISO 26262, annex A been used?				
E	Have the software routines used for memory checking been identified?				
F	Have the software routines for handling of faults been identified?				
G	Is the complete address range of the invariable memory covered by the test?				
H	Has the software for memory checking been analysed without finding faults?				
I	Has the software for handling of faults in memory been checked without finding faults?				
J	Has the Diagnostic Coverage been determined?				
K	Has the Dangerous Failure Coverage been determined?				
L	Has the Monitoring Coverage been determined?				
M	Are the techniques and measures used adequate for the intended SIL (or ASIL)?				

*) Ap. = Applicable question (Some questions may not be applicable for all control systems.)

9 Conclusions

An internationally accepted framework for functional safety in road vehicles is needed.

The use of embedded systems in automotive applications will continue to grow. There is presently no international guidelines accepted by the automotive industry. Company internal documents for risk analysis and functional safety exist. The standard IEC 61508 exists since a couple of years, but has not been accepted by the automotive industry. The development of standard ISO 26262 for functional safety of road vehicles is championed by the automotive industry, but will not be established before 2008. Safety guidelines exist from independent organisations such as MISRA.

It is difficult to work with parallel frameworks available. Future establishment of the most important functional safety standard for the automotive industry is expected to solve this. The different use of terminology causes confusion. Safety engineers, software developers, embedded systems engineers and tool vendors from different organisations have to reach a common understanding on definitions and use of words. Also this is expected to be supported establishing of a common standard.

It is possible to specify functional safety requirements.

Specification of functional safety goals requires focus on functionality. The development engineer often has knowledge of too many technical details from similar embedded systems. It is often difficult for the engineer to omit the technical implementation and focus on the safety-related functionality. Means to assist this implementation-independent analysis and design are necessary, such as model-based design.

It is possible to specify a technical safety concept.

The allocation of the identified functionality to different parts of the embedded system is part of the technical safety concept. Specification of the safety integrity level, and techniques and measures to use are also necessary. A sound architecture is important to a safe and cost-efficient system, and is thus an important part of the technical safety concept.

Safety validation is possible.

Validation requires the use of several validation methods to show functionality, hardware safety integrity and software safety integrity. The validation plan should include different methods for different aspects, and the validation activities should be planned to form an integrated part of the development work. Calculation of electronic hardware reliability is new to many automotive engineers. Tools to support reliability calculations are available. It may be hard to see how faults propagate and how they affect the overall functionality of a complex system.

This report is hoped to be read and discussed among developers of automotive embedded systems. International standards are not intended to be read as textbooks. The text and the examples of this report can be read to explain some of the concepts and issues. It should also stimulate the debate on concepts used for developing automotive systems.

Further examples of safety analysis and safety validation are available inthe AutoVal reports concerning validation methods [AutoValSP07:14] and model driven V&V [AutoVal SP07:15].

Annex A References

A.1 Draft standard ISO 26262

[ISO26262]
Preliminary work item ISO/TC22/SC3/WG16 Functional Safety
ISO/WD 26262
Road vehicles – Functional safety
Part 1: Glossary
Part 2: Management of functional safety
Part 3: Concept phase
Part 4: Product development at system level
Part 5: Product development - Hardware
Part 6: Product development - Software
Part 7: Production and operation
Part 8: Supporting processes
(Information will be available at www.iso.ch or from national standardisation organisations.)

A.2 International standard IEC 61508

[IEC61508]
International standard IEC 61508
Functional safety of electrical / electronic / programmable electronic safety-related systems
Part 1: General requirements
Part 2: Requirements for electrical/electronical/programmable electronic safety-related systems
Part 3: Software requirements
Part 4: Definitions and abbreviations
Part 5: Examples of methods for the determination of safety integrity levels
Part 6: Guidelines on the application of parts 2 and 3
Part 7: Overview of techniques and measures
(The standard can be purchased at www.iec.ch or from national standardisation organisations.)

[IEC]
IEC Functional Safety Zone.
General information on functional safety available at www.iec.ch/functionalsafety .

A.3 International safety regulations

[ECE-R13]
E/ECE/324, E/ECE/TRANS/505
United Nations Regulation No. 13
UNIFORM PROVISIONS CONCERNING THE APPROVAL OF VEHICLES OF CATEGORIES M, N AND O WITH REGARD TO BRAKING
Annex 18, Special requirements to be applied to the safety aspects of complex electronic vehicle control systems

[ECE-R79]
E/ECE/324, E7ECE/TRANS/505
United Nations Regulations No.79
UNIFORM PROVSIONS CONCERNING THE APPROVAL OF VEHICLES WITH REGARD TO STEERING EQUIPMENT

[GRRF/2003/27]
TRANS/WP.29/GRRF/2003/27 Proposal for a new draft regulation uniform technical prescriptions concerning the approval of complex electronic control systems affecting the direct vehicle control by the driver (Download at http://www.unece.org/trans/main/welcwp29.htm)

A.4 Additional documents

[MISRA]
Guidelines for the Safety Analysis of Vehicle Based Programmable Systems
The Motor Industry Software Reliability Association, MISRA
Draft issued 2005

[MISRAcontrol]
Hazard Classification for Moving Vehicle Hazards, Controllability
MISRA Technical Report
Version 1, May 2004
The Motor Industry Software Reliability Association, MISRA
(Download at www.misra.org.)

[MISRAweb]
Web site of The Motor Industry Software Reliability Association
www.misra.org.uk

[Goble]
Control Systems Safety Evaluation & Reliability 2nd Edition
Wiliam M. Goble
ISBN 1-55617-636-8

[MIL-HDBK-217F]
Military Handbook no 217F
Reliability Prediction of Electronic Equipment

[MIL-HDBK-338B]
Military Handbook no 338B
Electronic Reliability Handbook

A.5 AutoVal reports

The AutoVal project has released the following reports available at www.sp.se :

[AutoVal SP07:13]
Jan Jacobson, Andreas Söderberg,
Lars-Åke Johansson QRtech, Henrik Lönn Volvo Technology
Safety requirements and validation methods for safety-related automotive electronics
SP Report 2007:13

[AutoVal SP07:14]
Lars Strandén, Andreas Söderberg, Jan Jacobson, Josef Nilsson
Methods for Verification and Validation of Safety
SP Report 2007:14

[AutoVal SP07:15]
Lars Strandén, Josef Nilsson, Henrik Lönn Volvo Technology
Model Driven Software Verification and Validation
SP Report 2007:15

A final report is available from the IVSS project:
[AutoValIVSS]
AutoVal – Validation methods and safety requirements for safety-related automotive systems

Annex B Glossary

automotive safety integrity level (ASIL): one of four classes to specify the risk and its requirements for risk reduction with *D* representing the highest and *A* the lowest risk reduction class.

equipment under control (EUC): equipment, machinery, apparatus or plant used for manufacturing, process, transportation, medical or other activities
[IEC 61508-4, clause 3.2.3]

functional safety: part of the overall safety relating to the EUC and the EUC control system which depends on the correct functioning of the E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities
[IEC 61508-4, clause 3.1.9]

safety: freedom from unacceptable risk
[IEC 61508-4, clause 3.1.8]

safety function: function to be implemented by an E/E/PE safety-related system, other technology safety-related system or external risk reduction facilities, which is intended to achieve or maintain a safe state for the EUC, in respect of a specific hazardous event
[IEC 61508-4, clause 3.5.6]

safety integrity: probability of a safety-related system satisfactorily performing the required safety functions under all the stated conditions within a stated period of time
[IEC 61508-4, clause 3.5.2]

safety integrity level (SIL): discrete level (one out of a possible four) for specifying the safety integrity requirements of the safety functions to be allocated to the E/E/PE safety-related systems, where safety integrity level 4 has the highest level of safety integrity and safety integrity level 1 has the lowest [IEC 61508-4, clause 3.5.6]

safety life cycle: necessary activities involved in the implementation of safety-related systems, occurring during a period of time that starts at the concept phase of a project and finishes when all of the E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities are no longer available for use
[IEC 61508-4, clause 3.7.1]

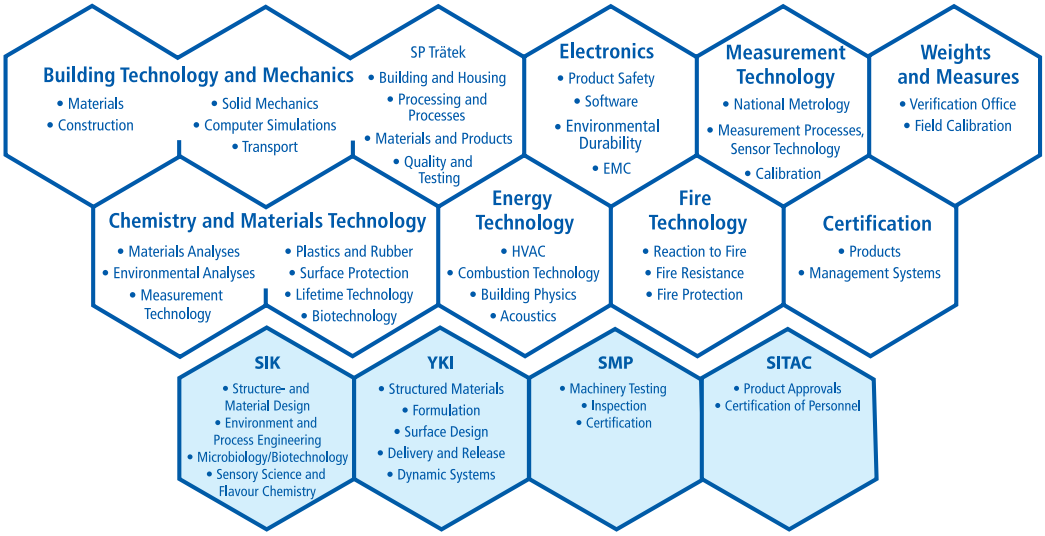
safety-related system: designated system that both
– implements the required safety functions necessary to achieve or maintain a safe state for the EUC; and
– is intended to achieve, on its own or with other E/E/PE safety-related systems, other technology safety-related systems or external
[IEC 61508-4, clause 3.4.1]

verification : confirmation by examination and provision of objective evidence that the requirements have been fulfilled [IEC 61508-4, clause 3.8.1]

validation : confirmation and provision of objective evidence that the particular requirements for a specific intended use are fulfilled [IEC 61508-4, clause 3.8.2]

SP Technical Research Institute of Sweden develops and transfers technology for improving competitiveness and quality in industry, and for safety, conservation of resources and good environment in society as a whole. With Swedens widest and most sophisticated range of equipment and expertise for technical investigation, measurement, testing and certfication, we perform research and development in close liaison with universities, institutes of technology and international partners.

SP is a EU-notified body and accredited test laboratory. Our headquarters are in Borås, in the west part of Sweden.



SP is organised into eight technology units and four subsidiaries

SP Technical Research Institute of Sweden

Box 857, SE-501 15 Borås, Sweden

Telephone: +4610 516 50 00

Telefax: +46 33 13 55 02

E-mail: info@sp.se

www.sp.se

SP Electronics

SP REPORT 2007:13

ISBN 978-91-85533-83-1

ISSN 0284-5172

A Member of

United Competence