Safety-related Machine Control Systems using standard EN ISO 13849-1

Andreas Söderberg, Johan Hedberg, Peter Folkesson, Jan Jacobson

RISE Report 2018: 01
Safety-related Machine Control Systems using standard EN ISO 13849-1

Andreas Söderberg, Johan Hedberg, Peter Folkesson, Jan Jacobson
Abstract

Safety-related Machine Control Systems using standard EN ISO 13849-1

Machine control systems shall be designed according to the European Machinery Directive and appropriate European standards. This report gives guidance when applying EN ISO 13849-1:2015 in projects, both for companies developing subsystems and for companies that are developing complete machines.

Key words: safety of machinery, machine control, safety function, PL, SIL, EN ISO 13849-1

RISE Research Institutes of Sweden AB
RISE Report 2018: 01
ISBN: 978-91-88695-33-8
Borås 2018
Content

Abstract .......................................................................................................................... 2
Content ......................................................................................................................... 3
Preface ........................................................................................................................... 5
Summary .......................................................................................................................... 6
1 Safety of machinery .................................................................................................. 8
   1.1 The Machinery Directive ....................................................................................... 8
   1.2 Machine Control Systems .................................................................................... 10
   1.3 Safety components and logic units ...................................................................... 12
2 Management of functional safety ........................................................................... 14
3 Risk assessment ........................................................................................................ 17
   3.1 Risk assessment and risk reduction at system level ............................................. 17
   3.2 Identify the safety functions to be performed by SRP/CS ................................... 20
   3.3 For each safety function specify the required characteristics ............................. 21
   3.4 Determine the required performance level PLr .................................................... 22
   3.5 Identify the safety related parts that carry out the safety function ...................... 23
   3.6 Evaluate the performance level ......................................................................... 26
4 Categories and designated architectures ............................................................... 27
   4.1 Safety principles .................................................................................................. 27
   4.2 Well-tried components ....................................................................................... 28
   4.3 Designated architectures ................................................................................... 30
      4.3.1 Category B .................................................................................................... 30
      4.3.2 Category 1 .................................................................................................... 31
      4.3.3 Category 2 .................................................................................................... 32
      4.3.4 Category 3 .................................................................................................... 33
      4.3.5 Category 4 .................................................................................................... 34
5 Probability of dangerous failure ............................................................................ 36
   5.1 Systematic failures ............................................................................................. 36
   5.2 Random hardware failures (MTTFD) ................................................................. 37
      5.2.1 Basic definitions ......................................................................................... 37
      5.2.2 Relation between MTTF and MTTFD ......................................................... 39
      5.2.3 Estimation of MTTFD for electric/electronic components ............................. 40
      5.2.4 Estimation of MTTFD for electromechanical, pneumatic or hydraulic components ................................................................. 41
      5.2.5 Estimation of MTTFD for individual SRP/CS ............................................. 42
6 Diagnostic coverage ................................................................................................ 48
   6.1 What is diagnostic coverage? .............................................................................. 48
   6.2 The reaction of the machine control system ....................................................... 49

© RISE Research Institutes of Sweden
Preface

RISE Research Institutes of Sweden is a notified body for the Machinery Directive. We perform EC Type Approvals of safety components. RISE is also a notified body for several other directives.

SMP Svensk Maskinprovning is also part of RISE. SMP is a notified body for many type of machinery.

This report is based on experience collected in many projects for safe machine control systems; both research projects and commissions for manufacturers of machinery. An earlier version of this report was issued in 2011 with the title “How to design safe machine control systems – a guideline to EN ISO 13849-1” (SP Report 2011:81).

The report should be read as guidance, and not be interpreted as requirements. The requirements can be found in the Machinery Directive and in harmonised standards.

Please obtain the full text of EN ISO 13849-1 to know all parts of the standard. Standards are protected by copyright and can be bought from ISO (www.iso.org) or your national standardisations (e.g. www.sis.se in Sweden).
Summary

This report gives a general guidance concerning how to apply EN ISO 13849-1, and describes a number of important aspects that need more detailed explanations:

- Safety of machinery
- Management of functional safety
- Risk assessment
- Categories and designated architectures
- Probability of dangerous failures
- Diagnostic Coverage
- Common Cause Failure
- Software
- Safety validation
- Achieved Performance Level

Software and complex electronics may be trusted. The risks are kept at a tolerable level by applying adequate techniques and measures to avoid faults during development and production, and to detect and handle faults at run-time before they manifest themselves as critical failures of the SRP/CS.

The concern for adequate safety has to be present all through the safety life cycle. Risks should be considered already in the first concept design, and then all through the development life cycle. Safety considerations must exist also during production, use and maintenance of the machine.
1 Safety of machinery

1.1 The Machinery Directive

There are risks associated with the use of machinery. The European machinery directive has the aim of harmonising the health and safety requirements applicable to machinery on the basis of a high level of protection of health and safety. Fulfilling these basic health and safety requirements also ensures the free circulation of machinery on the European market.

All machines that are used within the EU and EES area shall fulfil the EU machinery directive. There may also be other applicable directives such as the Low Voltage Directive (for electrical safety) and the EMC directive (for electromagnetic interference) which must be followed.

The Directive (Article 2, a) defines “machinery” as

*an assembly, fitted with or intended to be fitted with a drive system other than directly applied human or animal effort, consisting of linked parts or components, at least one of which moves, and which are joined together for a specific application,*

This general definition of machinery is developed further in the directive, which also explains types of machinery excluded from the scope of the directive. Weapons, seagoing vessels, equipment for use in fairgrounds and many types of motor vehicles, are some of the excluded types of equipment.
The machinery directive explains the principle of safety integration in three successive steps:

- **First priority** – eliminate hazards
  (e.g. use a low-power laser instead of a high-power laser, if possible)
- **Second priority** – protective measures applied for remaining risks
  (e.g. install physical protection around moving parts)
- **Third priority** – warn the users for remaining hazards
  (e.g. explain the noise level of a chain saw requires hearing protection)

The order of priority must be followed when selecting measures to deal with a given risk in order to satisfy the essential health and safety requirements. The machine manufacturer must exhaust all possibilities to eliminate hazards before applying protective measures. He must then exhaust the possible protective measures before relying on warnings and instructions to operators.

The requirements in the EU machinery directive are intentionally written in such way to make it possible for different technical solutions. The EU machinery directive does not want to prescribe a detailed technical solution that soon can become out of date. Interpretations of the directive are found in harmonized European standards.

It is the responsibility of the manufacturer to declare conformance with the machinery directive. The contents of the “EC declaration of conformity” are described in the directive. The manufacturer shall affix the European CE mark to a machine put on the market.
1.2 Machine Control Systems

Most machines have a control system to control the operation of the machine. Input signals are processed according to algorithms in logic. Output signals are set or reset. Most control systems are based on electronics and software. But other technologies such as hydraulics and mechanics may be applied for the control.

The concept “safety-related part of a control system” (SRP/CS) is used to describe the part of a control system that responds to safety-related input signals and generates safety-related output signals. A “part” can be a single component, a composition of components (a so called subsystem) or even a combination of different subsystems. Furthermore, a SRP/CS can be considered regardless of the type of technology and energy used (electrical, hydraulic, pneumatic, mechanical, etc.). Most of the control systems addressed today is based on electronics and software.

A safety function may be implemented by one or more SRP/CS to provide input, logic/processing and output (See Figure 2). It is possible that a SRP/CS implements both safety functions and non-safety related functions. It can also be that several safety functions may share one or more SRP/CS [e.g. a logic unit, power control element(s)].

![Figure 2 - Combination of safety-related parts of control systems providing a safety function](image)

The complexity of a control system may raise questions on how safety is achieved. What requirements can be put on a safety-related control system? How do we know that the requirements are fulfilled, and the risks are kept at a tolerable level?

The Machinery Directive (Appendix 1, Clause 1.2.1) states the basic requirement on control systems:

*Control systems must be designed and constructed in such a way as to prevent hazardous situations from arising. Above all, they must be designed and constructed in such a way that:*

- they can withstand the intended operating stresses and external influences,
- a fault in the hardware or the software of the control system does not lead to hazardous situations,
- errors in the control system logic do not lead to hazardous situations,
- reasonably foreseeable human error during operation does not lead to hazardous situations.
The Directive (Appendix 1, Clause 1.2.1) goes on to state that particular attention must be paid to following points:

- the machinery must not start unexpectedly,
- the parameters of the machinery must not change in an uncontrolled way, where such change may lead to hazardous situations,
- the machinery must not be prevented from stopping if the stop command has already been given,
- no moving part of the machinery or piece held by the machinery must fall or be ejected,
- automatic or manual stopping of the moving parts, whatever they may be, must be unimpeded,
- the protective devices must remain fully effective or give a stop command,
- the safety-related parts of the control system must apply in a coherent way to the whole of an assembly of machinery and/or partly completed machinery.

For cable-less control, an automatic stop must be activated when correct control signals are not received, including loss of communication.
1.3 Safety components and logic units

There are components intended especially to perform a safety function. The safety function serves to fulfil a protective measure to eliminate or, if not possible, to reduce a risk.

Examples of safety components are

- Guards for removable mechanical transmission devices
- Protective devices designed to detect the presence of persons
- Logic units to ensure safety functions
- Restraint systems to keep persons in their seats
- Emergency stop devices
- Two-hand control devices

Safety components may be purchased by the machine manufacturer and included in the machine control system.

![Examples of safety components with embedded electronics and software](image)

The Machinery Directive (Article 2, c) defines a “safety component” as a component:

- which serves to fulfil a safety function,
- which is independently placed on the market,
- the failure and/or malfunction of which endangers the safety of persons, and
- which is not necessary in order for the machinery to function, or for which normal components may be substituted in order for the machinery to function.

Many machinery components are critical for the health and safety of persons. However, purely operational components are not considered as safety components. Safety components are components intended by the component manufacturer to be fitted to machinery specifically to fulfil a protective role, in addition to any operational duty.
Components placed independently on the market that are intended by the component manufacturer for functions that are both safety and operational functions, or that are intended by the component manufacturer to be used either for safety or for operational functions, are to be considered as safety components.

Some devices (e.g. an industrial remote control) incorporate both non-safety related functions and one or more safety functions. As soon as a device serves to fulfil a safety function, it is considered as safety component in the sense of the machinery directive.

For certain types of machinery and logic units certain specific procedures for the CE-marking are prescribed. For components or logic units mentioned in Appendix 4 or 5 in the EU machinery directive, certain specific rules shall be followed to be able to fulfil the requirements. As an example, it can be necessary to use a notified body in this case.

The notified body may issue an EC Type Approval for logic units such as
- Logic units for two-hand control devices,
- Safety PLCs,
- Components for the logical processing of safety-related signals of Safety Bus Systems.

Examples of logic units to ensure safety functions include e.g.:
- Protective devices for indirect detection of the presence of persons, for example by the use of RFID technology;
- Protective devices for the detection and deactivation of possible hazards (not a warning system only), such as the detection of laser radiation;
- Safety control units, for example for the monitoring of speed, vibration, torque, temperature, pressure, force, guards, emergency stop devices, two-hand control devices, enabling device
- Rotary encoders, length measuring devices, speed measuring devices and braking control units with integrated logic intended to be used in safety functions;
- Safety PLCs;
- Remote controls contributing to at least one safety function, e.g. emergency stop;
- Power Drive Systems (for example PDS(SR) according to EN 61800-5-2) with one or more integrated safety functions (e.g. STO, SS1, SS2, SLS, SBC), e.g. frequency inverters, servo converters;
- Time delay devices for safety functions;
2 Management of functional safety

The standard EN ISO 13849-1 does not have a specific clause giving an overview of how to handle questions concerning management of functional safety. Nevertheless it is an important part when designing a SRP/CS or a safety function. Clause 10 in the standard lists which information shall be produced during the project.

- safety function(s) provided by the SRP/CS;
- the characteristics of each safety function;
- the exact points at which the safety-related part(s) start and end;
- environmental conditions;
- the performance level (PL);
- the category or categories selected;
- the parameters relevant to the reliability ($MTTF_d$, $DC$, $CCF$ and mission time);
- measures against systematic failure;
- the technology or technologies used;
- all safety-relevant faults considered;
- justification for fault exclusions (see EN ISO 13849-2);
- the design rationale (e.g. faults considered, faults excluded);
- software documentation;
- measures against reasonably foreseeable misuse.

Since EN ISO 13849-1 lacks requirements about management systems the authors of this report highly recommend studying (as inspiration) other functional safety standards which also contain such requirements. One example of such a standard is IEC 61508 (other examples are IEC 62061 and ISO 26262). Those parts of IEC 61508 which concerns management of functional safety are briefly summarized as follows:

IEC 61508-1, clause 6, claims two objectives for management of functional safety; to define responsibilities and to list activities for functional safety. IEC 61508-1 requires one or more persons to take overall responsibility for the system and for its lifecycle phases. The responsible persons shall coordinate the safety-related activities carried out in those phases. All persons, departments and organizations responsible for carrying out activities in the lifecycle phases shall be identified, and their responsibilities shall be fully and clearly communicated to them.
Figure 4  Standard IEC 61508-1 may be used for management of functional safety.

Standard IEC 61508-1 may also inspire the compilation of a functional safety plan, even if this is not required by standard EN ISO 13849-1. Guidance concerning which information to include in the functional safety plan can be found in IEC 61508-1, clause 6.

When working with the standard, at least the following parts are recommended to apply: Develop a so called functional safety plan (which preferably could be integrated into the validation plan, see chapter 9.1 in this report), describing:

- Activities during the project
- Identify persons and organizations responsible for different activities during the project
- Competence of the persons involved in the different activities (for further reading, see clause 6.2.13 and 6.2.14 in IEC 61508-1)
- How to document the different steps in the project
- Requirements when performing modification in the component/system (for further reading, see clause 7.16 in IEC 61508-1)
- How to perform the verification (can be efficient to split up in a separate document)
- How to plan and perform the validation (can be efficient to split up in two separate documents)
- How to handle issues identified during for instance risk analysis, verifications, validations, audits, reviews by independent organizations, incident reporting
- Which requirements that shall be placed on suppliers when they are responsible for parts of the design of a SRP/CS or a safety function
The most important part concerning the functional safety plan is to find out how to implement it in a way so it becomes easy to use and an integral part of the design process both for the current project and for future projects.

Guidance:

- It is important to early in the project to decide which documents that shall be developed by you as a manufacturer/integrator, and which documents that shall be developed by the organization responsible for the evaluation/certification, for more information see Clause 10 in EN ISO 13849-1
- Involve the organization responsible for evaluation/certification as early as possible in the project. The reason for this is to detect possible deviations from the requirements in EN ISO 13849-1 as early as possible
- A general aspect for these new standards concerning functional safety is that it is not enough to design a safe system. Additionally, you must also be able to show that your system is safe by showing that you have correctly documented all parts of your development, from the initial risk analysis until the SRP/CS and/or safety function is finalized
- The functional safety plan is an important document during all parts of the project life cycle and needs to continuously be updated as the project proceeds
- Documentation of good quality not only simplifies for you as a manufacturer/integrator, but also for the organization responsible for the evaluation/certification. In some situations, for instance when a company does not already have existing procedures it may be efficient to build up the document structure in accordance with the clauses and requirements as described in EN ISO 13849-1
- If possible, it is preferable to integrate the process requirements from the standard into the normal processes of the company to avoid having two different management systems
- A problem is to follow the functional safety plan developed during the whole project and also after the project is finalized and possibly evaluated/certified by another organization, and thus it is important to design the functional safety plan in such way that it is applicable and usable
- Take into consideration if it could be efficient to use a program that handles management of functional safety
- More detailed information about functional safety management can be found in clause 6 in IEC 61508-1.
3 Risk assessment

3.1 Risk assessment and risk reduction at system level

The risk assessment is performed by the manufacturer of the complete machine. The reason for this is that it is only the manufacturer of the complete machine that has knowledge about the risks that comes with the use of the machine, and in which environment the machine shall be used.

The aim of the risk assessment is to:

- Identify hazards
- Identify which hazardous events that could be connected to each hazard
- Determine whether a risk reduction is necessary or not
- Determine how the required risk reduction shall be reached
  - Remove the hazard
  - Intrinsic design
  - By safeguards
  - By information for use

Figure 5 below describes the work flow during the risk assessment.
Figure 5  Risk reduction process [EN ISO 12100, Figure 1 (modified)]

In those cases where it is decided that the risk reduction shall be realized by implementing E/E/PE (Electrical, Electronic, Programmable Electronic) safety functions EN ISO 13849-1 gives further support and outlines the following work procedure.
In the rest of this report E/E/PE safety functions are abbreviated and are only called safety functions.

Figure 6 Iterative process for design of safety-related parts of control systems (SRP/CS)
Below clauses gives more detailed information about what to consider during these different phases as described in Figure 6.
3.2 Identify the safety functions to be performed by SRP/CS

In clause 3.1.20 in EN ISO 13849-1 a safety function is defined as

*function of the machine whose failure can result in an immediate increase of the risk(s)*

Safety functions can be used as both inherently safe design measures and/or as safety guarding according to Figure 5.

Chapter 3.1.38 in EN ISO 13849-1 defines **high demand or continuous mode** for a E/E/PE safety function in the following way:

*mode of operation in which the frequency of demands on a SRP/CS is greater than one per year or the safety related control function retains the machine in a safe state as part of normal operation*

An example on a safety functions working in continuous mode of operation can for instance be the steering function if you have a mobile machine with electronic steering system. This example symbolizes when a safety function can be introduced already as part of inherently safe design measures.

Examples on safety functions used as safety guarding working in high demand mode of operation are:

- Light curtain functions
- Turn off power to hydraulics in a mobile machine if the driver is not placed in the driver seat

EN ISO 13849-1 does not cover safety functions where the demand rate is less than once a year. This type of demand rates is common for safety functions used within the process control sector where the functional safety standard EN 61511 is applicable.

It is important to be accurate when defining the safety function and in detail consider which specific identified hazardous event from the risk analysis, the safety function shall address to reduce the risk. A good idea is to give the safety function a name that clearly describes its functionality. As an example, a safety function named “Turn off power to the motor when the door is opened” would give indications about what action that triggers the safety function and what action that shall be made after it has been triggered.

Documentation: List of all safety functions performed by SRP/CS.
3.3 For each safety function specify the required characteristics

The aim of this part is to give a more detailed description of the characteristics of each safety function. This part is important both because it is the input to the design and technical realization, but also a basis for the validation of each safety function.

It is a common misunderstanding that it is enough to only give the safety function a name and then directly go on with the hardware- and software design. As a rule of thumb the safety requirements specification shall be on such detailed level that a person without earlier knowledge about the machine in principle shall be able to find enough information in it to be able to continue with the implementation of hardware and software.

Clause 5.1 in EN ISO 13849-1 informs about the minimum information that shall be considered when defining the safety requirements for each safety function. Clause 5.2 in EN ISO 13849-1 describes more in detail the safety requirements for certain safety functions.

Developing a safety requirements specification with good quality is an iterative process.

The safety requirements specification will look different if you design a complete safety function compared to when designing a SRP/CS to be included in a safety function.

For a manufacturer of a control system (Logic SRP/CS) the safety requirements can be on a very general level, for instance

- A certain input value shall generate a certain output value

On the other side for a manufacturer of a complete machine the safety requirements can be on a much more specific level, for instance

- X ms after the door to the hazardous zone is opened the rotating part shall be completely stopped

Guidance:

- For companies developing only a SRP/CS (e.g. an off-the shelf subsystem, only implementing part of a safety function), the safety requirements specification will look different compared to a company developing a complete safety function, for instance:
  - The PLr will be based on a judgment of the market expectations or a PLr requirement from a product standard
  - It will only include requirements on the specific SRP/CS and not for the complete safety function.

- The safety requirements specification shall describe the functional requirements for each safety function, and thus it is important to not include any implementation specific requirements.
• The quality of the safety requirements specification will be increased if a number of persons with different competences are included in the work, for instance persons working with development, service and quality issues. Another efficient method is to let someone who has not been involved in the development of the document review the safety requirements specification.

• Go through clause 5.1 and 5.2 in EN ISO 13849-1 to get guidance concerning which information that shall be included in the safety requirements specification.

• When the safety requirements specification documentation is ready, it is possible to start writing the safety validation plan, which describes how each specific requirement in the safety requirements specification will be validated.

### 3.4 Determine the required performance level PLr

In EN ISO 13849-1, five different risk reduction levels (Performance Levels) are defined, from PLa to PLe, where PLe gives the highest risk reduction and PLa gives the lowest risk reduction.

**Table 1 PL correspondence to PFHₜ₀ [EN ISO 13849-1, Table 3]**

<table>
<thead>
<tr>
<th>PL</th>
<th>Average probability of dangerous per hour (PFHₜ₀) [1/h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>≥ 10⁻⁵ to &lt; 10⁻⁴</td>
</tr>
<tr>
<td>b</td>
<td>≥ 3 x 10⁻⁶ to &lt; 10⁻⁵</td>
</tr>
<tr>
<td>c</td>
<td>≥ 10⁻⁶ to 3 x 10⁻⁶</td>
</tr>
<tr>
<td>d</td>
<td>≥ 10⁻⁷ to &lt; 10⁻⁶</td>
</tr>
<tr>
<td>e</td>
<td>≥ 10⁻⁸ to &lt; 10⁻⁷</td>
</tr>
</tbody>
</table>

Each PL defined in above Table 1 corresponds to a certain average probability of dangerous failure per hour (PFHₜ₀) for the safety function.

For a manufacturer of a certain SRP/CS, a suitable PL can be found by checking the expectation from the market or if specific requirements on PL can be found in a certain product standard.
According to EN ISO 13849-1 the following figure can be used when deciding an appropriate risk reduction level:

![Risk Graph Diagram](image)

**Risk parameters:**
- S    Severity of injury
  - S1 Slight (normally reversibly injury)
  - S2 Serious (normally irreversible injury or death)
- F    Frequency and/or exposure to hazard
  - F1 Seldom-to-often and/or exposure time is short
  - F2 Frequent-to-continuous and/or exposure time is long
- P    Possibility of avoiding hazard or limiting harm
  - P1 Possibility under specific condition
  - P2 Scarcely possible

**Figure 7  Risk graph [ISO 13849-1, Figure A.1]**

Figure 7 provides a simple method to determine the PLₐ for a safety function. In EN ISO 13849-1 it exist a possibility to reduce the PLₐ by one step by motivating that the probability of occurrence of the hazardous event is low. According to clause A.2.3.2 in Appendix A in EN ISO 13849-1 the estimation whether the probability of the hazardous event is low or not shall be based on factors including:

- reliability data;
- history of accidents on comparable machines.

**NOTE** A low number of accidents does not necessarily mean that the occurrence of hazardous situations is low, but that the safety measures on the machines are sufficient.

Where comparable machines
- include the same risk(s) that the relevant safety function is intended to reduce,
- require the same process and operator action,
- apply the same technology causing the hazard.

It is important to be careful when trying to use this possibility to reduce the PLₐ by one step and have a very well-motivated reason to this.

### 3.5 Identify the safety related parts that carry out the safety function

A safety function is always built up by a combination of sensor, logic and a final element. Note 1 in clause 3.3.1 in EN ISO 13849-1 describes the complete safety function in the following way:
The combined safety-related parts of a control system start at the point where the safety-related input signals are initiated (including, for example, the actuating cam and the roller of the position switch) and end at the output of the power control elements (including, for example, the main contacts of a contactor).

The safety function shall only include parts that are safety related. The following definition of subsystem in clause 3.2.5 in IEC 62061 is useful also when working with EN ISO 13849-1 to identify which parts that shall be included in the safety function:

*entity of the top-level architectural design of the SRECS where a dangerous failure of any subsystem will result in a dangerous failure of a safety-related control function*

Looking into Figure 8 can also give support when defining the safety function and understand which parts of the machine that are considered as operative parts and which parts of the machine that can be considered as control system parts.

![Figure 8 Schematic representation of a machine (ISO 12100, Figure A.1)](image)

It is only those parts/components that belong to control system part in the above Figure 8 that needs to be included in the safety function.
This phase concerns the design and technical realization of the safety functions. A safety function is normally built up by a number of SRP/CSs, where each SRP/CS separately includes input, logic and output as described below:

Figure 9  Safety function provided by a combination of different SRP/CS (e.g. several off-the-shelf subsystems)

But in some cases both input, logic and output can be integrated in the same SRP/CS as described below:

Figure 10  Safety function provided by one SRP/CS (e.g. a control system designed for a specific purpose)

Or in some cases, by a combination of Figure 9 and Figure 10, e.g.:

Figure 11  Safety function provided by a combination of SRP/CS (e.g. an off-the-shelf subsystem combined with a control system designed for a specific purpose)

Guidance:

- Figure 8 gives guidance concerning which parts/components that needs to be included in the safety function
- It is important to identify which SRP/CS that are included in each safety function
- A rule of thumb is if a fault in the SRP/CS will lead to a failure of the safety function then the SRP/CS shall be included as part of the safety function
- At this high level description of the safety function it will be built up by a number of SRP/CS combined in serial.
- In some situations, the safety functions can be more complicated and for instance include two different input SRP/CS.
3.6 Evaluate the performance level

When all safety functions and their corresponding SRP/CS are identified, the next step is to go on with the design of the safety function. EN ISO 13849-1 describes that the following issues are important to consider:

The PL of the SRP/CS shall be determined by the estimation of the following aspects:

- the MTTF\textsubscript{D} value for single components (see section 5 in this report, and Annexes C and D in EN ISO 13849-1)
- the DC (see section 6 in this report, and Annex E in EN ISO 13849-1)
- the CCF (see section 7 in this report, and Annex F in EN ISO 13849-1)
- the structure, i.e. the behavior of the safety function under fault condition(s) (see section 4 in this report and Clause 6 in EN ISO 13849-1)
- safety-related software (see section 8 in this report and Clause 4.6 and Annex J in EN ISO 13849-1)
- the ability to perform a safety function under expected environmental conditions (not covered in this report).
- systematic failure (see section 5.1 in this report and Annex G in EN ISO 13849-1)

When finally the design of the safety-related control system implementing the safety function(s) is finished according to the above, the achieved PL for each safety function shall be evaluated. Different strategies for how to do this are discussed in section 10 in this report.
4 Categories and designated architectures

The categories specified in EN ISO 13849-1 represent a classification of the ability of the SRP/CS structure (hardware architecture) to handle (resist/tolerate) random hardware faults which may occur within the SRP/CS internal design. The SRP/CS must be assigned at least one category.

EN ISO 13849-1 defines five different categories: B, 1, 2, 3 and 4 where category B represents the least fault resistant structure and category 4 the most fault resistant structure of the SRP/CS.

In accordance to EN ISO 13849-1, fault resistance of the categories is accomplished by either increasing the reliability of the SRP/CS (i.e. the electric/electronic components of the hardware design) or by applying redundancy in order to achieve fault tolerance in combination with some requirements on behavioral properties. In addition to this, the category also requires that certain so called safety principles are used in the design of the SRP/CS and which provides best practice design advices that shall be utilized (with respect to EN ISO 13849-1).

The category is one of the basic parameters in EN ISO 13849-1 for determining the achieved PL of the safety function(s) implemented by the SRP/CS.

It is essential to take the category into consideration at an early stage in the development process of the SRP/CS (especially if the SRP/CS is implemented by an embedded system) since it will affect both the subsequent hardware design and software design. If it at a late stage in the development process becomes evident that the targeted category for the SRP/CS is not fulfilled, this may require a complete redesign of the control system structure (hardware).

Historically (obsolete standard EN 954-1:1996) the category of the SRP/CS constituted the main/primary (and only) argument for functional safety of control systems for machinery. However, due to the level of complexity of the control systems used for safety critical applications today, the category alone is no longer sufficient as an argument. For control systems of higher complexity (which includes e.g. software) systematic failures (see Section 5.1) are more plausible leading to the loss of the safety function and thus also needs to be specifically addressed which requires other and additional techniques.

* Remember that product standards or the risk assessment can give other required categories due to PLc.

4.1 Safety principles

There are two types of safety principles according to EN ISO 13849-1, Basic safety principles and Well-tried safety principles. Both types of safety principles are specified in EN ISO 13849-2, for each type of technology used (mechanic-, hydraulic-, pneumatic and electric).

The safety principles are lists (tables) of different design principles/considerations (i.e. guidance) that shall have been applied during the design and development of the
SRP/CS. It is highly unlikely to successfully claim/argue that these safety principles all have been applied after the design of the SRP/CS is finished if not taken into account already during development.

According to the requirements in EN ISO 13849-1 the basic safety principles shall have been applied for SRP/CS conforming to category B. For all other categories (1-4), both basic safety principles and well-tried safety principles shall have been applied.

Table 2  Example of a basic safety principle [EN ISO 13849-2, Table D.1]

<table>
<thead>
<tr>
<th>Basic safety principle</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use of suitable materials and adequate manufacturing</td>
<td>Selection of material, manufacturing methods and treatment in relation to e.g. stress, durability, elasticity, friction, wear, corrosion, temperature, conductivity, dielectric rigidity.</td>
</tr>
</tbody>
</table>

Table 3  Example of a well-tried safety principle [EN ISO 13849-2, Table D.2]

<table>
<thead>
<tr>
<th>Well-tried safety principle</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>No undefined states</td>
<td>Avoid undefined states in the control system. Design and construct the control system so that, during normal operation and all expected operating conditions, its state, e.g. its output(s), can be predicted.</td>
</tr>
</tbody>
</table>

As can be seen in Table 2 and Table 3 above, these requirements are not always verifiable, thus a recommendation is to copy-paste the safety principles into separate tables and then to establish a new column (e.g. to the right) in which the developer describes by which means the corresponding safety principle has been applied.

4.2  Well-tried components

A well-tried component shall be carefully selected and also be demonstrated that it is suitable for the intended application.

The aspects that influence if a component can be regarded as well-tried are:

- follow well-tried safety principles
- have low complexity and
- are demonstrated suitable by applying applicable standards.

For category 1 solutions the well-tried component is a key component for safety.

Just because a component is qualified as “well-tried” it does not mean that any “fault exclusions” can be pre-assumed. It only means that it is suitable and reliable enough for
its purpose. For more information about fault exclusion, please refer to section 9.3 in this report.

Description of a well-tried component in EN ISO 13849-1:

A “well-tried component” for a safety-related application is a component which has been either

a) widely used in the past with successful results in similar applications, or

b) made and verified using principles which demonstrate its suitability and reliability for safety-related applications.

It is important to understand that the qualification of a component to be well-tried depends on its application. If safe operation relies on a single component, it is of great importance that this component is designed and implemented for the final application by following basic and well-tried safety principles. A well-tried component used in some application can be inappropriate for other applications.

For example cabling to external enclosure should be protected against mechanical damage (including e.g. vibration or bending) in order to be regarded as a “well-tried” component.

Complex electronic components (e.g. PLC, microprocessor, application-specific integrated circuit) cannot be considered as equivalent to “well tried”. Complex electronic components are characterized by (according to IEC 62061, clause 3.2.8):

– Their failure modes are not well-defined; or

– Their behavior under fault conditions cannot be completely defined

EN ISO 13849-1 provides specific lists of some well-tried components in Table A.3 (mechanical) and D.3 (electrical). These tables also contain requirements that have to be fulfilled in order to classify these components as well-tried.

Furthermore, any components listed in EN ISO 13849-2, Annex A, B, C or D are candidates for being qualified as well tried if the requirements mentioned in this section are fulfilled.

Table 4  Example of a well-tried component [EN ISO 13849-2, Table D.3]

<table>
<thead>
<tr>
<th>Well-tried component</th>
<th>Additional conditions for “well-tried”</th>
<th>Standard or specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cable</td>
<td>Cabling external to enclosure should be protected against mechanical damage (including, e.g. vibration or bending).</td>
<td>IEC 60204-1:2005, Clause 12</td>
</tr>
</tbody>
</table>
4.3 Designated architectures

For each category in EN ISO 13849-1 an example structure for the SRP/CS is presented and illustrated as a generalized diagram. These diagrams are called designated architectures and are basically simplified reliability block diagrams (RBDs) that are extended by additional information about monitoring.

An important and sometimes challenging task when working with EN ISO 13849-1 is to map the actual/physical structure of the SRP/CS to one (or several) designated architectures.

Designated architectures are presented by a graphical structure with boxes and arrows for each category by the standard. To be able to apply the simplified method the architecture of the SRP/CS shall be in accordance with one of these designated architectures.

If the SRP/CS cannot be mapped to a designated architecture other methods for calculation of hardware reliability are possible, such as those appointed by IEC 61508 (not covered by this report). However, also in this case the requirements of the corresponding category must still be proven fulfilled.

4.3.1 Category B

The figure below presents the designated architecture of the SRP/CS for category B. “Category B” means that “Basic safety principles” have been applied in the design and development of the SRP/CS. Furthermore, category B may be achieved by a single channeled SRP/CS (i.e. without redundancy).

![Designated architecture for category B](EN ISO 13849-1, Figure 8)

All abbreviations used in Figure 12 are defined in Annex C in this report.

SRP/CS that fulfils category B is mainly characterized by the selection of its components, the occurrence of a fault can lead to the loss of the safety function.

The B category gives “basic requirements”, these requirements are also required for all other categories (1, 2, 3 and 4).

The requirements for category B means that the components of the SRP/CS are suitable for the intended use with respect to:

- design, construction, selection, assembly and combination so the SRP/CS components are in accordance with relevant standards
- environmental conditions, for example temperature, vibrations, dust, moisture, humidity, water
- operating stresses, influences of materials processed and other relevant influences.
• basic safety principles, see section 4.1 in this report

It is not possible to generally claim that a component is suitable for category B since this will depend on the intended use and the environmental conditions of the SRP/CS. Thus, it is the responsibility of the developer of the SRP/CS to ensure that the specification of each used component complies with the intended application in accordance to applicable basic safety principles.

Example of category B solutions:
• Interlocking device for a house-hold washing machine which prevents the machine to start when the front hatch is open

4.3.2 Category 1

The figure below presents a designated architecture for category 1. Category 1 may be achieved by a single channeled SRP/CS (i.e. without redundancy).

![Figure 13 Designated architecture for category 1](image)

All abbreviations used in Figure 13 are defined in Annex C in this report.

The category 1 structure is mainly characterized by selection of components, the same principle as category B, and the occurrence of a fault can lead to the loss of the safety function. The probability of occurrence of a fault is lower than a category B structure in comparison.

Basic requirements of category B shall apply but in addition well-tried safety principles (see section 4.1 in this report) and well-tried components (see section 4.2 in this report) shall be used.

Example of category 1 solutions:
• door interlock switch for a wood working machine
• emergency stop device
4.3.3 Category 2

The figure below presents a designated architecture for category 2. In difference from category B and 1, category 2 is not a single channel system. Basic requirements of category B shall apply and were applicable, well-tried safety principles shall be used.

In comparison to category B and 1 has category 2 an additional test equipment (TE) which checks the safety function (i.e. the functional channel implemented by the Input, Logic and Output) by a suitable interval (see the dashed lines in Figure 14). If a fault is detected by the check, the TE initiates a safe state (see section 6.1) using its dedicated output (OTE). Although, if the safety function is classified to PLr = a, b or c, and it is not practicable possible to initiate a safe state, then it is sufficient if the OTE generates a warning. The occurrence of a fault can lead to the loss of the safety function between the checks.

![Designated architecture for category 2](EN ISO 13849-1, Figure 10)

All abbreviations used in Figure 14 are defined in Annex C in this report.

The checking interval is depending on the application and can be time scheduled or based on the operating cycle or the machine cycle. The checking interval needs to be evaluated/determined during the risk assessment for the application but shall be kept as short as possible. I, L and O shall be checked/monitored.

All “boxes” of the designated category 2 structure need a corresponding hardware unit.

In addition, EN ISO 13849-1 states that the TE may be integral with, or separate from the safety related part(s) providing the safety function as illustrated with grey-shaded lines in Figure 14. This is usually interpreted so that it is allowed to integrate TE into L (e.g. into the same microcontroller), leading to a single channeled structure for this part of the SRP/CS.

The Output of Test Equipment (OTE) needs however has to be separated/independent from the Output (O). Thus, a category 2 structure is a mixture of a category B and a category 3 system since input is only one sensing unit and output is two separate units.
In some applications category 2 is difficult to realize since some of the components (I, L or O) may not be checked. In this case a category 3 system may be more suitable since a category 3 structure is based on two independent hardware channels with comparison/monitoring of the two channels. In some cases it can be appropriate to subdivide the category 2 structure into two different SRP/CS, one input-SRP/CS fulfilling category 3, and another logic-and output-SRP/CS fulfilling category 2 (see Figure 11). Hence the checking interval of the input can be made continuous.

Example of O and OTE components:

- Relay
- Contactor
- Transistor

Example of category 2 solutions:

- Force limitation system for an industrial door

4.3.4 Category 3

Category 3 is a redundant system with monitored inputs and outputs (in other words a two channeled system with diagnostics). Single faults shall not lead to loss of safety function.

Basic requirements of category B shall apply and applicable well-tried safety principles shall be used.

The designated architecture for category 3 is presented in EN ISO 13849-1.

![Diagram](image)

Figure 15 Designated architecture for category 3 [EN ISO 13849-1, Figure 11]

All abbreviations used in Figure 15 are defined in Annex C in this report.

Some faults are not detected by a category 3 system; these faults shall have a motivation why they are not detected. All “boxes” of the designated category 3 architecture need a corresponding hardware unit.
Inputs (I1 and I2) are checked so that discrepancies are detected. When a discrepancy (single fault) is detected, an action is taken to enforce a pre-defined safe state (i.e. the safety function is performed). See section 6.1 about safe states.

Logic (L1 and L2) are checked so that discrepancies are detected. When a discrepancy is detected, action is taken to enforce a safe state.

Outputs (O1 and O2) are checked so that discrepancies are detected. When a discrepancy is detected action is taken to enforce a safe state.

Example of category 3 solution(s):

- Input circuit for an interlock door for Machinery. The I1 and I2 are two separate electric channels of one electro-mechanic door key switch with positive mode of operation. The switch has two electrical channels but only one mechanical channel (the key). Mechanical faults are in this case excluded since this component is regarded as well-tried due the mechanical design and the contact elements I1 and I2 have positive mode of operation.

4.3.5 Category 4

Category 4 is a redundant system with monitored inputs and outputs (in other words a two channel system that has monitoring of inputs and outputs). Single faults shall not lead to loss of safety function and accumulation of undetected faults shall not lead to the loss of the safety function. Category 4 offers a higher degree of resistance to faults in comparison with category 3.

Basic requirements of category B shall apply and applicable well-tried safety principles shall be used.

A designated architecture for category 4 is presented in EN ISO 13849-1.

![Designated architecture for category 4](EN ISO 13849-1, Figure 12)
All abbreviations used in Figure 16 are defined in Annex C in this report.

The accumulation of two faults (also see section 9.3 in this report) is considered to be sufficient in EN ISO 13849-1:

The difference between category 3 and category 4 is a higher DCavg in category 4 and a required MTTFD of each channel of “high” only. In practice, the consideration of a fault combination of two faults may be sufficient.

Inputs (I1 and I2) are checked so that discrepancies are detected. When a discrepancy (single fault) is detected action is taken to enforce a safe state (i.e. the safety function is performed). See section 6.1 about safe states.

Logic (L1 and L2) are checked so that discrepancies are detected. When a discrepancy is detected action is taken to enforce a safe state.

Outputs (O1 and O2) are checked so that discrepancies are detected. When a discrepancy is detected action is taken to enforce a safe state.

Note that in EN ISO 13849-2, table D.8 about electromechanical position or manually operated switches states that for PL e (i.e. category 4, or in some cases also category 3), fault exclusions for mechanical and electrical aspects is not allowed. Thus redundancy is necessary. Emergency stop devices are however excluded from this requirement if they fulfill appropriate standards.

Example of a category 4 solution:

- Input circuit for an interlock door for machinery. The I1 and I2 are two separate electro mechanic door key switches. Then each key switch has one electrical channel and one mechanical channel (key) each. Mechanical faults are in this case not excluded since the combination of two separate electro mechanical switches achieves category 4.
5 Probability of dangerous failure

The probability of dangerous failure of the safety function depends on several factors, including hardware and software structure, reliability of components, design process, operating stress, environmental conditions and operation procedures. The reliability of components can be described by using mean time to dangerous failure (MTTF$_D$), the extent of fault detection mechanisms, diagnostic coverage (DC) and common cause failure (CCF).

The aim of the clause is to give a short introduction in the concept of MTTF$_D$, how to retrieve MTTF$_D$-values for components and how to estimate the total MTTF$_D$ for a SRP/CS.

The MTTF$_D$ is given in three levels and shall be taken into account for each channel of the SRP/CS individually.

Table 5 Levels of MTTF$_D$ [EN ISO 13849-1, Table, 4]

<table>
<thead>
<tr>
<th>Denotation of each channel</th>
<th>Range of each channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>3 years ≤ MTTF$_D$ &lt; 10 years</td>
</tr>
<tr>
<td>Medium</td>
<td>10 years ≤ MTTF$_D$ &lt; 30 years</td>
</tr>
<tr>
<td>High</td>
<td>30 years ≤ MTTF$_D$ &lt; 100 years</td>
</tr>
</tbody>
</table>

A channel can have a MTTF$_D$ maximum value of 100 years. If the estimation results in a channel with a MTTF$_D$ > 100 years, the resulting MTTF$_D$ is set to 100 years, except for Category 4 SRP/CS for which the maximum MTTF$_D$ for each channel is 2500 years.

The following sub-clauses are guidance.

5.1 Systematic failures

A systematic failure is a failure caused by a human mistake in any phase of the development process, e.g. requirement specification errors, software bugs and/or other design mistakes. It is not possible to quantify the probability of “remaining” systematic failures in SRP/CS (as it is for random hardware failures, see section 5.2 below).

Instead the approach is to firstly reduce the possibility for introducing design mistakes early (avoidance) and as far as practicable, and secondly to handle (control) remaining systematic failures (when they manifest) in the same manner as if they would be random hardware failures.

EN ISO 13849-1, Annex G (informative) proposes measures for:

- the control of systematic failures
- avoidance of systematic failures and
- avoidance of systematic failures during SRP/CS integration.
An example from this Annex G is a set of measures for controlling the effects of voltage breakdown, voltage variations, overvoltage and under voltage. This means that to forget to add/implement technical means for monitoring/supervising the power supply (by some aspect) is to be considered as having introduced a systematic failure. Another such example regards data communication processes (if included in the SRP/CS) for which EN ISO 13849-1 directly refers to IEC 61508-2:2000, clause 7.4.8. Although the reference to the standard is written with a date (formally meaning that it is this specific version of the standard that applies) we recommend to use the latest version (IEC 61508-1:2010, clause 7.4.11) instead.

When applying annex G it sometimes seems as some of its measures overlaps other measures (diagnostic techniques) that already have been applied in order to fulfill e.g. the required DC_{avg} of a category (see e.g. EN ISO 13849-1, Table 10). However, for certain SRP/CS, additional measures might be required to be able to control systematic failures even if the PL_c or category provides no such requirements. An example is if the user implements highly complicated/advanced software (SRESW) as part of a safety function classified as PL_c = b, then is still might be a good idea to implement a watchdog timer even though there are no requirements on DC by the category.

It should be noted that EN ISO 13849-1, Annex G is informative, which means that the user of the standard may choose to apply other sources/methods as an argument for handling systematic failures rather than this annex. However, if this annex is used (the most common situation), its specified measures are all to be considered as requirements. In this case the recommendation is to copy-paste every listed measure from the standard into an own table with an extra column in which the application of each measure is clearly justified (same reasoning as in section 7 in this report).

### 5.2 Random hardware failures (MTTF_D)

#### 5.2.1 Basic definitions

All hardware components have a probability of failure per unit time; this probability is called the component failure rate and is denoted with the symbol \( \lambda \) (lambda). Failure rate is often estimated in failures in time (FIT) which means that if a component has a failure rate of 1 FIT then the probability of failure for that component is \( 1 \times 10^{-9} \) per hour.

The failure rate for a certain type of component can be subdivided into three phases according to the following figure:
Phase 1 is the early life of the component. During this period the failure rate is expected to be high because of e.g. a not sufficiently adjusted manufacturing process.

During phase 2 the failure rate is assumed to be constant for electric/electronic, hydraulic and pneumatic components. This period is called the useful life of the component which often is symbolized with \( \theta \) (theta).

Phase 3 is the wear out phase which starts when the useful life of the component ends. In this phase the component is worn out because of physical reasons and the failure rate cannot longer be assumed to be constant.

Because the failure rate is assumed to be constant during the useful life period it can be shown that the mean time to failure (MTTF) can be calculated according to:

\[
MTTF = \frac{1}{\lambda} \text{[hours]}
\]

It is very important to make a difference between the MTTF and the \( \theta \) because these two measures have no relationship. For example, wet electrolytic capacitors often have a limited \( \theta \) because of drying in time. However before the end of \( \theta \) these capacitors usually have a very low failure rate and thus a very large MTTF.

Sometimes the term MTTF is confused with the term MTBF (mean time between failures). According to reliability theory literature MTBF is defined as follows:

\[
MTBF = MTTF + MTTR
\]

Where MTTR means: mean time to repair and is a measure of the expected time to successfully repair a component/system. Usually MTTR \(<\) MTTF (e.g. 8 hours compared with 100 years). The term MTBF is normally important in maintainability/availability analysis and will not be further considered in this report.
5.2.2 Relation between MTTF and MTTF_D

Example

Consider a relay with one contact supplying a motor. The failure rate for the relay is known ($\lambda_{RE}$). The relay has two failure modes, stuck open or stuck closed and the relay manufacturer has specified that if a relay failure occurs, it is equally probable that any of these failure modes occur. This is called distribution of the failure rate among the failure modes of a component. Reliability prediction handbooks may provide guidance for distribution for certain types of components (but not for all types) if not the distribution is carried out by good engineering practice.

<table>
<thead>
<tr>
<th>Failure mode</th>
<th>Failure effect</th>
<th>Failure rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stuck-open</td>
<td>The motor cannot start, or stops unexpectedly</td>
<td>Safe failure effect = $50% \cdot \lambda_{RE}$</td>
</tr>
<tr>
<td></td>
<td>(safe failure)</td>
<td></td>
</tr>
<tr>
<td>Stuck close</td>
<td>Unexpected start, or the motor does not stop</td>
<td>Dangerous failure effect = $50% \cdot \lambda_{RE}$</td>
</tr>
<tr>
<td></td>
<td>(dangerous failure)</td>
<td></td>
</tr>
</tbody>
</table>

In a realistic case, there would be a lot more components in the FMEA. When the FMEA is completed the total failure rate leading to safe failure effects is added together. This total failure rate is denoted with the symbol $\lambda_s$ (safe failure rate) and the total failure rate leading to dangerous failure effects is denoted with the symbol $\lambda_d$ (dangerous failure rate) where:

$$MTTF_D = \frac{1}{\lambda_d} \text{[years]}$$

When estimating MTTF_D for a component the following procedure for finding data shall be followed according to Clause 4.5.2 in EN ISO 13849-1:

a) Use manufacturer's data
b) Use methods in Annexes C and D in the standard
c) Choose ten years
5.2.3 Estimation of $MTTF_D$ for electric/electronic components

There are different techniques to estimate the failure rate for components, either the failure rate is determined by counting failures in the field use for a large population of components, and then use statistical methods (which is the most accurate method) or the failure rate is predicted using a reliability prediction handbook.

Always check if the manufacturer specifies the $MTTF_D$ value in the component datasheet. In some cases the datasheet only contains a $\lambda_D$ value (this is common for electronic modules such as I/O modules and sensors). In this case use the formula $MTTF_D = 1/\lambda_D$.

However, for standard passive components (transistors, diodes, resistors etc.) for which no reliability data is available, use the following:

EN ISO 13849-1, Annex C provides reliability figures for most discrete electronic components and may be used unless the component manufacturer provides reliability data. For complex components (integrated circuits) consult a reliability expert who can help predicting failure rates if not provided in the data sheet.

Example from Table C.2 (in the standard EN ISO 13849-1), Bipolar transistor which is assigned with the following values:

$MTTF = 38052$ years

$MTTF_D = 76104$ years

For each electronic component in Annex C in the standard EN ISO 13849-1 it is assumed that 50\% of all the component failure modes leads to a dangerous failure providing the typical $MTTF_D$:

$$MTTF_D = 2 \times MTTF \text{ (since } \lambda_D = 0.5 \times \lambda = \frac{1}{2}\lambda \Rightarrow \frac{1}{MTTF} = \frac{1}{2 \times MTTF}).$$

Power electronics often contribute most of all electronic components to the total $MTTF_D$.

If no reliability data can be found for an electronic component or module use 10 years (e.g. standard industrial PLCs).
5.2.4 Estimation of MTTF$_D$ for electromechanical, pneumatic or hydraulic components

The procedure of estimating the MTTF$_D$ for electromechanical components (relays, contactors, push buttons, levers limit switches, guard interlocks etc), pneumatic components and hydraulic components is clearly described in Annex C in the standard EN ISO 13849-1.

$B_{10D}$ is the number of operations a set of electromechanical or pneumatic components can perform until 10% of the set of components failed dangerously. This value is derived in a $B_{10}$-test and is to be acquired from the component manufacturer.

The $B_{10D}$ value is used to estimate the MTTF$_D$ for the above mentioned components.

In order to be able to use Table C.1 in EN ISO 13849-1 which prescribes $B_{10D}$ values for electromechanical, pneumatic components and a MTTF$_D$ value for hydraulic components the requirements in Annex C.2 and C.3 shall be documented by the component manufacturer, e.g. in the datasheet. Otherwise the manufacturer shall deliver the $B_{10D}$ value or the MTTF$_D$ value.

With a $B_{10D}$ value available, the following formula may be used for deriving the MTTF$_D$ value:

$$MTTF_D = \frac{B_{10D}}{0.1 \times n_{op}}$$

Where $n_{op}$ is the mean number of annual operations for the component. E.g. for a relay is one relay activation and the subsequent relay de-activation two operations. Equation C.2 in EN ISO 13849-1 suggests how $n_{op}$ can be derived. However, to be able to show the rationale behind the estimation of $n_{op}$ is more important than strictly applying Equation C.2.

$n_{op}$ is also used when estimating the MTTFD for hydraulic components according to Table C.1 in EN ISO 13849-1.

For some components it is difficult for the component manufacturer to provide a $B_{10D}$ value because it is application dependent which failures that actually is dangerous. In this case the manufacturer only provides a $B_{10}$ value. The following pessimistic assumption is in this case feasible (see Annex C.4.2, footnote 2 in EN ISO 13849-1):

- $B_{10D} = 2 \times B_{10}$ (assuming that 50% of the components failure modes leads to dangerous failure effects)

In cases where the component manufacturer cannot provide a $B_{10}$ value for the component, a pessimistic assumption that $B_{10}$ equals the components specified electrical life as stated in the datasheet is permissible.

Because the MTTF$_D$ for electromechanical or pneumatic components depends on the application of the component it is common that these type of components has a large impact on the total SRP/CS MTTF$_D$ value.
Furthermore, when both the $n_{op}$ and the $B_{10D}$ values are obtained for a component, the next step is to calculate the $T_{10D}$ (the mean time until 10% of the components fail dangerously) according to:

$$T_{10D} = \frac{B_{10D}}{n_{op}} = 0.1 \cdot MTTF_D$$

If either the resulting $MTTF_D$ or $T_{10D}$ value for the component is less than 20 years, the $MTTF_D$ calculation for the whole SRP/CS can be accounted for only if the component is replaced before the value is exceeded by preventive maintenance (in these cases this must be documented in the information for use supplemented with the control system/machine).

Consider the following example using a contactor relay with nominal load which gives a $B_{10D}$ of 400000 according to Table C.1 in the standard EN ISO 13849-1.

<table>
<thead>
<tr>
<th>$n_{op}$ (mean relay contactor operations per year)</th>
<th>Relay contactor $MTTF_D$ [years]</th>
<th>Relay contactor $T_{10D}$ [years]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/year</td>
<td>4 million</td>
<td>400 thousand</td>
</tr>
<tr>
<td>1/month</td>
<td>333 thousand</td>
<td>33 thousand</td>
</tr>
<tr>
<td>1/week</td>
<td>77 thousand</td>
<td>8 thousand</td>
</tr>
<tr>
<td>1/day</td>
<td>11 thousand</td>
<td>1 thousand</td>
</tr>
<tr>
<td>1/hour</td>
<td>457</td>
<td>46</td>
</tr>
<tr>
<td>1/minute</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>

### 5.2.5 Estimation of MTTF$_D$ for individual SRP/CS

When each safety related component is identified together with the SRP/CS structure every component is gathered in a spreadsheet (e.g. Excel or similar) and are grouped to their respective Input-block, Logic-block or Output block.

The designated architectures are in fact simplified reliability models based on a concept called channels. A channel is defined so that in all components within the channel there are failure modes which can cause the loss of the safety function. Each series of Input-Logic-Output is a channel and thus relates to the structures according to the following table:
Table 8  Categories and reliability model

<table>
<thead>
<tr>
<th>Structure</th>
<th>Reliability model configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category B</td>
<td>Single channel</td>
</tr>
<tr>
<td>Category 1</td>
<td>Single channel</td>
</tr>
<tr>
<td>Category 2</td>
<td>Single channel</td>
</tr>
<tr>
<td>Category 3</td>
<td>Dual channel</td>
</tr>
<tr>
<td>Category 4</td>
<td>Dual channel</td>
</tr>
</tbody>
</table>

According to the simplified method in EN ISO 13849-1, Annex D the MTTF\textsubscript{D} of each channel is determined by the following formula:

\[
\frac{1}{MTTF_{D,\text{channel}}} = \sum_{i=1}^{K} \frac{n_i}{MTTF_{D,i}}
\]

Where:

- \(i\) is the component type
- \(K\) is the number of different component types within the channel
- \(n_i\) is the number of components of type \(i\) within the channel
- \(MTTF_{D,i}\) is the MTTF\textsubscript{D} value for the particular component type \(i\)

No hardware solely used for implement diagnostics shall be part of the SRP/CS estimated MTTF\textsubscript{D}

5.2.5.1  Exceptions to the requirement of MTTF\textsubscript{D} calculation

If the output part of the SRP/CS contains only hydraulic, pneumatic or mechanical components (or a mixture of these) and there are no reliability data available for these components, then EN ISO 13849-1 makes an exception to the requirement of MTTF\textsubscript{D} calculation (instead of using MTTF\textsubscript{D} = 10 years for each such component).

In these cases the PL for this part of the SRP/CS is instead solely determined by its structure (category), the applied diagnostic techniques and the measures taken against CCF according to the following table:
### Table 9  Determining the PL for the output part of the SRP/CS without MTTF\(_D\) calculation [Also see EN ISO 13849-1, Table 7]

<table>
<thead>
<tr>
<th>PL</th>
<th>Cat. B</th>
<th>Cat. 1</th>
<th>Cat. 2</th>
<th>Cat. 3</th>
<th>Cat. 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>HR</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>b</td>
<td>HR</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>c</td>
<td>N/a</td>
<td>HR</td>
<td>HR</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>d</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
<td>HR</td>
<td>R</td>
</tr>
<tr>
<td>e</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
<td>HR</td>
</tr>
</tbody>
</table>

**Notes**

- R – Recommended (if chosen, all notes in the same column still apply)
- HR – Highly recommended
- N/a – Not applicable/allowed

Note 1: Well-tried components and well-tried safety principles must be used.

Note 2: MTTF\(_D\) for the TE must be at least 10 years.

Note 3a: DC shall be low to medium.

Note 3b: DC shall be high.

Note 4: Proven in use OR well-tried components and well-tried safety principles must be used (T10D may be determined based on proven in use data).
5.2.5.2 Example of estimating the MTTF$_D$ for a SRP/CS

Consider the following hypothetical SRP/CS:

![Diagram of SRP/CS](image)

Figure 18  Fictive SRP/CS

This system is a fictive application specific SRP/CS. The safety function is a hatchet (guard) which shall remain physically locked with two hypothetical plungers preventing access to hazardous movement while a motor shaft rotates. The locking devices can only open the plungers with electromagnets controlled by the contactors. Power loss to the electromagnets causes mechanical locking of the hatchet by the plungers.

The SRC/PS shall fulfill PL$_r$ = d.

Previously, the hardware has been analyzed (FMEA) and the structure category 3 was identified. The hardware was illustrated as channels excluding any diagnostics, i.e. no feedback signals, bus-communication between Microcontroller 1 and PLC, watchdogs etc. are included. The two channels were also found to be not identical.

Refer to section 8.8 in this report regarding the use of a standard industrial PLC in a safety-related application.

The estimation of the MTTF$_D$ was performed using the following two tables.

Note that some of the values in these tables are fictive and not intended for professional use.
Table 10  MTTFd for Channel 1

<table>
<thead>
<tr>
<th>Device</th>
<th>Block</th>
<th>Component</th>
<th>Source of MTTFd</th>
<th>Number of components [ni]</th>
<th>MTTFd [years]</th>
<th>ni/MTTFd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Sensor</td>
<td>B1, Inductive sensor 1, JXZoK65J7</td>
<td>Datasheet provides $\lambda_d$</td>
<td>1</td>
<td>700</td>
<td>0.001429</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logic</td>
<td>Interfacing electronics in1</td>
<td>Suppressor</td>
<td>Table C</td>
<td>1</td>
<td>3196</td>
<td>0.000313</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Resistor, Metal film</td>
<td>Table C</td>
<td>3</td>
<td>114 155</td>
<td>0.000021</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Capacitor, Ceramic</td>
<td>Table C</td>
<td>2</td>
<td>4566</td>
<td>0.000438</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inductor, Low freq.</td>
<td>Table C</td>
<td>1</td>
<td>4566</td>
<td>0.000219</td>
</tr>
<tr>
<td></td>
<td>Microcontroller 1 (Large)</td>
<td>IC1</td>
<td>Datasheet</td>
<td>1</td>
<td>1114</td>
<td>0.000298</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Suppressor</td>
<td>Table C</td>
<td>1</td>
<td>3196</td>
<td>0.000313</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Resistor, Metal film</td>
<td>Table C</td>
<td>4</td>
<td>114 155</td>
<td>0.000035</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Capacitor, Ceramic</td>
<td>Table C</td>
<td>2</td>
<td>4566</td>
<td>0.000438</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MOS, power</td>
<td>Table C</td>
<td>1</td>
<td>228</td>
<td>0.004386</td>
</tr>
<tr>
<td>Output</td>
<td>Actuator</td>
<td>Contactor, NO</td>
<td>B_{od} = 400 000, n_{op} = 24*365</td>
<td>1</td>
<td>457</td>
<td>0.002188</td>
</tr>
</tbody>
</table>

1/MTTFd Channel 1 = Add all calculated ni/MTTFd in column 7 = 0.010677

Resulting MTTFd for Channel 1 (inverse of the result of the previous row) 94
Table 11 MTTFD for Channel 2

<table>
<thead>
<tr>
<th>Device</th>
<th>Block</th>
<th>Component</th>
<th>Source of MTTFD</th>
<th>Number of components [n]</th>
<th>MTTFD [years]</th>
<th>n/MTTFD,i</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Sensor</td>
<td>B1, Inductive sensor 1, Component data</td>
<td>Datasheet provides λ_D</td>
<td>1</td>
<td>700</td>
<td>0.001429</td>
</tr>
<tr>
<td>Logic</td>
<td>N/a</td>
<td>PLC</td>
<td>Not available</td>
<td>1</td>
<td>10</td>
<td>0.1</td>
</tr>
<tr>
<td>Output</td>
<td>Actuator</td>
<td>Contactor, NO</td>
<td>B_{ind} = 400 000, n_{op} = 24*365</td>
<td>1</td>
<td>457</td>
<td>0.002188</td>
</tr>
</tbody>
</table>

1/MTTF_D Channel 2 = Add all calculated n_i/MTTF_D,i in column 7 = 0.103617

Resulting MTTFD for Channel 2 (inverse the result of the previous row) = 9

Clause 4.5.2 in EN ISO 13849-1 requires that the MTTFD for each channel is considered individually, if there are different MTTFD for two channels the lower shall be selected. In this example:

MTTF_D Channel 1 = 94 years
MTTF_D Channel 2 = 9 years

Please read also refer to section 10 in this report before reading the following:

Means that MTTFD SRP/CS total = 9 years, and thus “low” according to Table 17 in this report. According to Figure 29 in this report it is only possible to reach PL = c for a category 3 structure if MTTFD = low which contradicts the previous PL = d.

However, the standard provides a technique in order to resolve this problem by allowing the use of the following formula:

\[ MTTF_{D,eqv} = \frac{2}{3} [MTTF_{D,chl} + MTTF_{D,ch2} - \frac{1}{MTTF_{D,chl}} + \frac{1}{MTTF_{D,ch2}}] \]

Which in this example would provide:

\[ MTTF_{D,eqv} = \frac{2}{3} [94 + 9 - \frac{1}{94} + \frac{1}{9}] = 63 \text{ years} \]

This SRP/CS MTTFD value is within the interval for “high” MTTFD and thus is sufficient for PL = d according to Figure 29 in this report.
6 Diagnostic coverage

6.1 What is diagnostic coverage?

A control system should have diagnosis routines to find faults before they manifest themselves as failures of the functionality. The diagnostic routine should detect as many faults as possible, i.e. the objective is a high diagnostic coverage. The aim of this clause is to discuss the concept and meaning of diagnostic coverage (DC) and to explain some techniques which may be applied in order to estimate the DC for a function or a module included in a SRP/CS.

Recalling section 5.2 in this report, the failure rate for a component can be subdivided into different fractions depending on the components different failure mode effects on the control system (\( \lambda_d \) and \( \lambda_s \)). If automatic self-checking and error detecting mechanisms are included in the control system and which detects certain dangerous failures a third fraction can be derived from the FMEA called \( \lambda_{dd} \) which means dangerous detected failure rate.

The definition of diagnostic coverage is [IEC 61508]:

\[
DC = \frac{\sum \lambda_{dd}}{\sum \lambda_d} [\%]
\]

Where:

\( \sum \lambda_d \) = The part of the SRP/CS total failure rate which leads to a dangerous failure, e.g. the loss of the safety function (dangerous failure rate); and

\( \sum \lambda_{dd} \) = The part of the SRP/CS total dangerous failure rate which is detected (AND handled) by automatic diagnostic tests (online monitoring) that are implemented in the SRP/CS or by other systems external to the SRP/CS.

Safety-related faults may also be revealed by manual tests or inspections. However, the coverage of such procedures does not contribute to the diagnostic coverage.

When a fault is detected, the monitoring mechanisms shall handle the fault by initiating an appropriate action which is application dependent. For many applications within the machinery sector such an appropriate action is to initiate a so called safe-state (i.e. the safety-function is performed). The term safe-state implies that the control system removes the hazard instantly (e.g. by immediately stopping/preventing hazardous movement of a part of a machine by remove the power to a motor). For other machines or applications other actions may be more appropriate, such as issuing an alarm.
Unfortunately standard EN ISO 13849-1 does not define the term safe-state (at all) but refers to this term anyway at several locations. In standard IEC 61508:2010, Part 4, Clause 3.1.13 the term “safe-state” is defined as:

**safe state**

state of the EUC when safety is achieved

**NOTE** In going from a potentially hazardous condition to the final safe state, the EUC may have to go through a number of intermediate safe states. For some situations a safe state exists only so long as the EUC is continuously controlled. Such continuous control may be for a short or an indefinite period of time.

Note: EUC = Equipment Under Control (i.e. the machine)

With regard to this definition the following general conclusions of the term “safe-state” can be made:

- There is no pre-defined/standardized control system output state which shall be achieved in order to enter safe-state (e.g. all outputs de-energized)
- The safe-state can be entered by a controlled sequence
- The machine does not have to be stopped when the safe-state is entered

### 6.2 The reaction of the machine control system

The machine control system reaction on a failure may be visualized in accordance to the following figure:

![Diagram of machine behavior when a fault occurs](image)

**Figure 19** Machine behavior when a fault occurs
Figure 19 is subdivided into the following time intervals

a. During this time interval the machine operates according to its specification (i.e. safe operation
b. A fault (e.g. random hardware fault or systematic failure) has occurred in the control system, the machine may now operate dangerously
c. The control system detects and registers the fault by internal diagnostic mechanisms and initiate a preventive action
d. During this time period the machine performs its preventive actions (e.g. applying brakes, remove power from drives, etc.)
e. During this time the machine is in its safe state
f. The time elapsed between a failure and the hazardous event without considering diagnostic mechanisms

If possible (b + c + d) shall be less than f.

If a diagnostic test is not performed frequently enough, there is a probability that a failure causes a hazardous event between tests.

Table 12 in this report specifies four levels of diagnostic coverage according to the following:

<table>
<thead>
<tr>
<th>Denotation</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>DC &lt; 60%</td>
</tr>
<tr>
<td>Low</td>
<td>60% ≤ DC &lt; 90%</td>
</tr>
<tr>
<td>Medium</td>
<td>90% ≤ DC &lt; 99%</td>
</tr>
<tr>
<td>High</td>
<td>99% ≤ DC</td>
</tr>
</tbody>
</table>
The average diagnostic coverage for the whole SRP/CS be estimated using the following equation when applying the simplified method (see Annex E of standard EN ISO 13849-1):

\[ DC_{avg} = \frac{\sum_{n=1}^{k} DC_n}{\sum_{n=1}^{k} \frac{1}{MTTF_{d,n}}} \]

6.3 Guidance

Incorrect detection of faults in the SRP/CS may cause the operation to shut down unnecessarily. It is important to consider the robustness of the machine operation when designing diagnostic techniques in order to avoid unnecessary loss of production and/or to reduce the risk that the machine operator tries to tamper with safety functions.

Diagnostic test technique should be designed to allow natural variations. One example is when applying cross monitoring of the feed-back signals from two redundant electromechanical components. Even if the components are identical they will have different response times (so called discrepancy times). Therefore the control system must tolerate a certain time deviation between the feedback signals. The length of the maximum allowed time deviation is to be determined in the hazard and risk analysis. A similar reasoning is applied when measuring analog values etc.

Examples of possible (non-exhaustive) interpretations of the implementation of some diagnostic techniques are presented in Appendix A in this report.
7 Common Cause Failure

Redundancy is a common structure for a SRP/CS when control with a high performance level is needed. The redundant structure is efficient to control random failures occurring in only one of the redundant channels. But common cause failures (CCF) can affect both channels in a negative way. Examples of CCF are short circuits, extreme temperatures and electromagnetic interference.

![Diagram of a common cause failure affecting two channels.](image)

Figure 20 A single common cause failure affects two channels

In order to prevent that a common cause failure can affect safety the aim is to reduce the probability for common cause failures. During the design phase of a SRP/CS it is possible to implement measures in order to reduce the probability for common cause failures.

The standard EN ISO 13849-1 requires the performance level of the control system to be determined with estimation of CCF as one important aspect. An assessment of CCF is necessary for every safety validation, but can be performed in different ways.

The standard provides a (qualitative) procedure for estimating the CCF measures implemented for SRP/CS with a category 2, 3 or 4 structures. The procedure is presented by a scoring table F.1 in EN ISO 13849-1. The proposed procedure is described in an informative part of the standard. Other procedures can be used to judge measures against CCF, but the proposed score table F.1 is commonly applied. The scoring table covers the following areas:

- Separation of signal cables and also creepage distances on printed-circuit boards

The intention is to avoid short circuits between the redundant channels. It is
usually very important when static input and output signals are used. The need for separation may be reconsidered when dynamic signals are used.

- **Diversity in technologies, design or physical principles**
  The intention is to reduce the probability of a fault affecting both channels. An example is different sensitivity to electromagnetic interference in different components, e.g. an electromechanical or an electronic sensor. Another example is when diversity in software is applied to reduce the risk of a programming mistake affecting both channels.

- **Design/application/experience**
  The intention is to reduce the probability of an external factor affecting both channels at the same time. An example is when a high voltage transient bursts from inductive loads destroys electronic components in both channels.

- **Failure mode and effect analysis covering CCF failures**
  The intention is to identify critical components of the design and reduce the probability of a fault appearing in both channels.

- **Competence/training in order to understand the causes and consequences of CCF**
  Both design engineers and maintenance staff should be trained to understand the significance of reducing CCF.

- **Suitable design with respect to environmental impact**
  Environmental aspects may affect both channels at the same time. An example is that EMC performance of the design has been tested and approved. This will reduce the probability of a disturbance affecting both channels.

For each area above points are presented. In order to fulfill the requirements a score of minimum 65 points or better is needed. For each listed measure, only the full score or nothing can be claimed. If a measure is only partly fulfilled, the score according to this measure is zero. The maximum score is 100 points. When performing the assessment, a motivation for every judgment shall be noted.

Table 13  Scoring process and quantification of measures against CCF [EN ISO 1849-1, Table F.1]

<table>
<thead>
<tr>
<th>No</th>
<th>Measure against CCF</th>
<th>Max score</th>
<th>Achieved score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Separation / segregation</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>Diversity</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>Design: Protection against overvoltage, current, etc.</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>3.2</td>
<td>Design: Components are well tried</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Assessment / analysis</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>Competence / training</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>6.1</td>
<td>Environmental: EMC</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>6.2</td>
<td>Environmental: Other influences</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>100</td>
<td>90</td>
</tr>
</tbody>
</table>
8 Software

Today many safety functions are depending on both the correct functioning of hardware as well as the correct functioning of software. Earlier, mainly non-safety-critical functions were implemented in software whereas safety-critical functions were traditionally hard-wired. The recent trend has been to realize more safety-critical functions by software.

It is often advantageous to be able to realize both safety-critical and non-safety-critical software in the same fail safe system, but it is important to understand that safety is not automatically reached in this case and that the software developers are still responsible for developing safe software. Situations that have to be avoided are for instance that logical faults in the software do not result in unexpected behaviour of the machine.

The standard gives support to the software designer on what to consider during the different steps of the software safety lifecycle to, as far as it is possible, minimize the introduction of faults.

8.1 General requirements

Clause 4.6 “Software safety requirements” in the standard gives some general information about software management, for instance that a V-model may be used when developing the software, see Figure 21.

Two important activities in the software safety lifecycle are verification and validation. To distinguish between the two activities, it is often stated that verification is performed to answer the question “Are we building the system right?” while validation should answer the question “Are we building the right system?”
The verification activities are illustrated by dashed arrows in Figure 21. The aim of the verification in Figure 21 is to check that each phase in the software safety life cycle has been performed correctly. For example, a person that has not been involved in the development of the safety-related software specification may review the specification document to check that it is correct and fulfils the applicable requirements in the standard. Another example is software testing which is performed to verify that the code is implemented and works according to the design.

The V-model is suitable when the prerequisites of the software are stable. However, there are many variants which may be more suitable, e.g., when the prerequisites are less stable or due to specific demands within the development organization or depending on the tools used. For example, one may to work iteratively between the development phases of the V-model so that one or more steps are repeated until subsequent steps can be satisfactorily performed. In evolutionary development, a product is developed that always can be used by customer but is continuously improved. In incremental development, the whole development process is applied for just a part of the system first and then the other parts are added. Model based or component based development may be used to reduce or eliminate the work on different phases of the V-model, e.g., during the implementation phases. Agile development processes may involve the customer during the whole development process by always providing something executable.

Two different types of languages are defined in the standard

### limited variability language (LVL)

- type of language that provides the capability of combining predefined, application-specific library functions to implement the safety requirements specifications

Clause 3.1.34 in the standard

Typical examples of LVL are ladder logic and function block diagrams which are often used for programming PLCs.

### full variability language (FVL)

- type of language that provides the capability of implementing a wide variety of functions and applications

Clause 3.1.35 in the standard

Typical examples of FVL are C, C++ and assembler. In the field of machinery, FVL are typically used for programming embedded computer systems.
**application software**

software specific to the application, implemented by the machine manufacturer to meet the SRP/CS requirements

Clause 3.1.36 in the standard

For the application software the abbreviation SRASW (safety-related application software) is used.

**embedded software / firmware / system software**

software that is part of the system supplied by the control manufacturer and which is not accessible for modification by the user of the machinery.

Clause 3.1.37 in the standard

For the embedded software the abbreviation SRESW (safety-related embedded software) is used.

Figure 22 describes the relation between SRESW and SRASW. Note that parameterization of both application SW and embedded SW can be performed by an external parameterization tool, see Section 8.7.
According to Clause 4.6.3, the requirements for SRESW shall apply for SRASW written in FVL. This means that the software requirements in Clause 4.6.2 in the standard shall be followed when the software is written in FVL.

Components with PLr e written in FVL shall comply with SIL 3 according to IEC 61508. However, when using diversity in specification, design and coding, for the two channels used in SRP/CS with category 3 or 4, PLr e may be achieved by employing the measures required for PLr c-d in Clause 4.6.2.

The standard points out that it is important to document the results from the software safety life cycle phases. For SRESW, or SRASW written in FVL, for components with PLr a-b, this is particularly important for the specification and design phases, while all relevant phases of the lifecycle must be documented for components with PLr c-d. To be able to go back after a project is finalized and check how each step in the software safety life cycle is performed, the documentation has to be complete, available, readable and understandable.

The following additional measures for SRESW, or SRASW written in FVL, for components with PLr c-d are described in the standard:

- *project management and quality management system comparable to, e.g. IEC 61508 or ISO 9001*
- *configuration management to identify all configuration items and documents related to a SRESW release*

For SRASW written in LVL for components with PLr from c to e, the following additional measures with increasing efficiency (lower for PLr of c, medium for PLr of d, higher for PLr of e) are required or recommended:

- *Configuration management. It is highly recommended that procedures and data backup be established to identify and archive documents, software modules, verification/validation results and tool configuration related to a specific SRASW version.*
The subsequent sections of this chapter discuss the requirements of Clause 4.6 in EN ISO 13849-1 aligned with the phases of the V-model shown in Figure 21.

### 8.2 Safety-related software specification

The first phase in the V-model shown in Figure 21 is to develop a safety-related software specification. In this phase it is important to review the safety requirements specification for each safety function to check which of these requirements influence or need to be further detailed in the safety-related software specification. A number of aspects that needs to be considered in the safety-related software specification are given below:

- safety functions with required PL and associated operating modes
- performance criteria, e.g. reaction times
- real time properties
- hardware architecture with external signal interfaces detection and control of external failure
- detection and management of faults in sensors, logic units, actuators, and in the software itself (self-diagnosis)
- operating modes

For SRESW, or SRASW written in FVL, for components with PL c or d, a structured specification with safety requirements and design is a required measure according to clause 4.6.2 in the standard.

A structured specification may reduce complexity by successively creating a hierarchical structure of partial requirements until the simplest possible, visible relations exist between the requirements. According to IEC 61508-7 B2.2.1, this method emphasises the interfaces of the partial requirements and is particularly effective for avoiding interface failures.

The following list of requirements on requirements is useful when creating a hierarchical structure of partial requirements:

- **Unique** – only one requirement exists addressing a specific aspect
- **Atomic** – the requirement addresses one aspect. This also improves the possibility of modifications (less dependences with other requirements)
- **Complete** – within the scope of the individual requirement e.g. if above 150 is handled then also equal and less than 150 shall be handled
- **Unambiguous** – no room for different interpretations
- **Identifiable** – can be uniquely referenced
- **Correct** – shall address what is intended
- **Concise** – a focused formulation
- **Verifiable** – e.g. by using tests, analysis, inspection, proofs etc.
- **Traceable** – relationship to both upper and lower level requirements
• **Understandable** – i.e. anybody can understand the requirement. This might be somewhat in conflict with Concise.
• **Rationale** – a motivation for the requirement. This could be important since it will improve the understanding of the individual requirement as well as groups of requirements. However, include Rationale information only when really needed

When the safety-related software specification is developed it shall be verified. This can be done by a person not involved in the development of the safety-related software specification reviewing this document and checking it for correctness.

Apart from the specification document, another important output from this phase is the safety-related software validation plan specifying the tests to be performed in the validation phase.

### 8.3 System- and module design

When the safety-related software specification is ready, it is possible to go on with the system- and module design. The aim of this phase is to give a high level description of how the software will function.

The standard states that a modular and structured design shall be used. To fulfil this, the system is divided into a number of different modules. One reason for this requirement is that the risk of introducing faults during coding may be reduced and it will also be possible to test each module separately.

SRESW, or SRASW written in FVL, for components with PLc-d require separation in non-safety-related software to prevent the non-safety-related parts of the system from influencing the safety-related parts in undesired ways. For example, failures of any non-safety-related functions should not cause dangerous failures of the safety-related functions. There are also other advantages of separating the non-safety-related software. The complexity may be reduced while most system safety lifecycle activities may be carried out easier, e.g., the effort for testing the safety-related systems may be reduced.

Another requirement for SRESW, or SRASW written in FVL, for components with PLc-d is to use suitable programming languages and computer-based tools with **confidence from use**. ISO 26262-8, clause 11.4.7.2 states that a software tool shall only be argued as having increased confidence from use, if evidence is provided for the following:

• the software tool has been used previously for the same purpose with comparable use cases and with a comparable determined operating environment and with similar functional constraints,
• the justification for increased confidence from use is based on sufficient and adequate data,
• the specification of the software tool is unchanged, and
• the occurrence of malfunctions and corresponding erroneous outputs of the software tool acquired during previous developments are accumulated in a systematic way.
For SRASW written in LVL for components with PL from c to e, the following additional measures with increasing efficiency (lower for PL of c, medium for PL of d, higher for PL of e) are required or recommended:

- semi-formal methods to describe data and control flow, e.g. state diagram or program flow chart
- modular and structured programming predominantly realized by function blocks deriving from safety related validated function block libraries
- function blocks of limited size of coding
- code execution inside function block which should have one entry and one exit point
- architecture model of three stages, Inputs ⇒ Processing ⇒ Outputs
- assignment of a safety output at only one program location
- use of techniques for detection of external failure and for defensive programming within input, processing and output blocks which lead to safe state

During the design phase, it is also important to define how each module shall be tested. This information may be described in System- and module test plans.

8.4 Coding

When the system- and module design is ready, the next step is to start writing the detailed code for each module in the system.

For SRASW, or SRASW written in FVL, for components with PL c-d the following measures related to coding are given in the standard:

- modular and structured programming
- limited module sizes with fully defined interfaces
- use of design and coding standards
- coding verification by walk-through/review with control flow analysis

The following subsections of Section 8.4 describe these requirements in more detail followed by the requirements related to coding for SRASW written in LVL.

8.4.1 Modular and structured programming

The requirements on modular and structured programming as well as limited module sizes with fully defined interfaces are exemplified and described in more detail in IEC 61508-7 C.2.9. They can be summarized as follows:

- A software module shall have a single well-defined task or function to fulfil
- Connections between software modules shall be limited and strictly defined, coherence in one software module shall be strong
- Collections of sub-programs shall be built providing several levels of software modules
- Software module size shall be restricted to a specified value, typically two to four screen sizes
- Software modules shall have a single entry and a single exit
- Software modules shall communicate with other software modules via their interfaces – where global or common variables are used they shall be well
structured, access shall be controlled, and their use shall be justified in each instance

- All software module interfaces shall be fully documented
- Any software module interfaces shall contain only those parameters necessary for its function

8.4.2 Use of design and coding standards

An example of a coding standard commonly used in the development of safety-critical embedded systems is the MISRA-C coding standard by the Motor Industry Software Reliability Association. The latest MISRA-C:2012 version provides guidelines for C language programming in the form of 143 rules and 16 directives. The difference between rules and directives is that rules may typically be checked by static analysis tools while directives are more open to interpretations and typically require additional information in order to be checked.

The MISRA-C guidelines are categorised into mandatory, required or advisory guidelines. Mandatory guidelines always have to be used while required guidelines do not have to be used if a formal deviation is raised which is reviewed and approved by someone else. Advisory guidelines should be followed whenever possible but formal deviations do not have to be raised if it is not possible to follow the guideline. There are 10 mandatory and 100 required rules given in the standard while there are no mandatory and 9 required directives. In Table 14, the mandatory MISRA-C:2012 rules are listed while Table 15 gives a listing of the required MISRA-C:2012 directives.

MISRA-C compliance is often checked by automatic analysis tools. Examples of such tools include QAC, LDRA, Klocwork, Polyspace, Astrée and Axivion Bauhaus Suite.

Apart from MISRA-C there exist many other coding standards for various programming languages. It is not uncommon for larger organizations to employ their own coding standard or variant.

Table 14 Mandatory MISRA-C:2012 rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1</td>
<td>The value of an object with automatic storage duration shall not be read before it has been set</td>
</tr>
<tr>
<td>13.6</td>
<td>The operand of the sizeof operator shall not contain any expression which has potential side effects</td>
</tr>
<tr>
<td>17.3</td>
<td>A function shall not be declared implicitly</td>
</tr>
<tr>
<td>17.4</td>
<td>All exit paths from a function with non-void return type shall have an explicit return statement with an expression (see also Rule 15.5 (advisory): A function should have a single point of exit)</td>
</tr>
<tr>
<td>17.6</td>
<td>The declaration of an array parameter shall not contain the static keyword between the [ ]</td>
</tr>
<tr>
<td>19.1</td>
<td>An object shall not be assigned or copied to an overlapping object</td>
</tr>
<tr>
<td>22.2</td>
<td>A block of memory shall only be freed if it was allocated by means of a Standard Library function (see also Directive 4.12 (required): Dynamic memory allocation shall not be used)</td>
</tr>
<tr>
<td>22.4</td>
<td>There shall be no attempt to write to a stream which has been opened as read-only</td>
</tr>
</tbody>
</table>
### Rule Description

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.5</td>
<td>A pointer to a FILE object shall not be dereferenced</td>
</tr>
<tr>
<td>22.6</td>
<td>The value of a pointer to a FILE shall not be used after the associated stream has been closed</td>
</tr>
</tbody>
</table>

### Table 15 Required MISRA-C:2012 directives

<table>
<thead>
<tr>
<th>Directive</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Any implementation-defined behaviour on which the output of the program depends shall be documented and understood</td>
</tr>
<tr>
<td>2.1</td>
<td>All source files shall compile without any compilation errors</td>
</tr>
<tr>
<td>3.1</td>
<td>All code shall be traceable to documented requirements</td>
</tr>
<tr>
<td>4.1</td>
<td>Run-time failures shall be minimized</td>
</tr>
<tr>
<td>4.3</td>
<td>Assembly language shall be encapsulated and isolated</td>
</tr>
<tr>
<td>4.7</td>
<td>An object shall not be assigned or copied to an overlapping object</td>
</tr>
<tr>
<td>4.10</td>
<td>Precautions shall be taken in order to prevent the contents of a header file being included more than once</td>
</tr>
<tr>
<td>4.11</td>
<td>The validity of values passed to library functions shall be checked</td>
</tr>
<tr>
<td>4.12</td>
<td>Dynamic memory allocation shall not be used</td>
</tr>
</tbody>
</table>

### 8.4.3 Control flow analysis

The aim of control flow analysis is to detect poor and potentially incorrect program structures by static analysis of the program code. There are several methods which can be used, most of them are suitable for automatization while others may be performed manually.

*Static path analysis* and *cyclomatic complexity* are two control flow methods commonly used for detecting potentially inaccessible or poorly structured code. Static path analysis counts the number of possible combinations of paths through the code while cyclomatic complexity counts the number of linearly independent paths (i.e., so that all possible paths are covered).

Figure 23 gives an example of path analysis. The program has been analysed producing the directed graph in Figure 23 which is then further analysed. Static path analysis gives 4 different combinations of paths while the cyclomatic complexity in this example is 3. The cyclomatic complexity \( c \) for a program graph consisting of \( A \) arcs and \( N \) nodes may be calculated as

\[
c = (A - N) + 2
\]
Other commonly used static analysis techniques include counting the number of lines of code (LOC), comment density (number of comment lines divided by LOC) or nesting depth (the number of nested function calls within the program).

In object-oriented programming depth of inheritance (the number of ancestors in the class inheritance hierarchy) and coupling (the number of classes a class depends on) are often used for complexity measurement.

Function Point Analysis can be used to calculate complexity by analysing the number of function points in the code, where the function points are typically related to outputs, inquiries, inputs, internal files, and external interfaces.

Cohesion is the degree to which the elements within a module belong together and can be referring to various aspects such as logical, function, temporal, procedural, sequential or communication aspects.

8.4.4 SRASW written in LVL

For SRASW written in LVL for components with PLr from c to e, the following additional measures with increasing efficiency (lower for PLr of c, medium for PLr of d, higher for PLr of e) are required or recommended:

Selection of tools, libraries, languages:

- Suitable tools with confidence from use: for PL = e achieved with one component and its tool, the tool shall comply with the appropriate safety standard; if two diverse components with diverse tools are used, confidence from use may be sufficient. Technical features which detect conditions that could cause systematic error shall be used. Checks should mainly be carried out during compile time and not only at runtime. Tools should enforce language subsets and coding guidelines or at least supervise or guide the developer using them.

- Validated function block (FB) libraries should be used — either safety-related FB libraries provided by the tool manufacturer (highly recommended for PL = e) or validated application specific FB libraries and in conformity with this part of EN ISO 13849.

- A justified LVL-subset suitable for a modular approach should be used, e.g. accepted subset of IEC 61131-3 languages. Graphical languages (e.g. function block diagram, ladder diagram) are highly recommended.

Where SRASW and non-SRASW are combined in one component:
• **SRASW and non-SRASW** shall be coded in different function blocks with well-defined data links
• there shall be no logical combination of non-safety-related and safety-related data which could lead to downgrading of the integrity of safety-related signals

**Software implementation/coding:**

• code shall be readable, understandable and testable
• justified or accepted coding guidelines shall be used
• data integrity and plausibility checks (e.g. range checks) available on application layer (defensive programming) should be used
• code should be tested by simulation
• verification should be by control and data flow analysis for PL = d or e.

**Documentation:**

• code documentation within source text shall contain module headers with legal entity, functional and I/O description, version and version of used library function blocks, and sufficient comments of networks/statement and declaration lines.

### 8.5 Verification and validation

Verification and validation are important activities during the whole software safety lifecycle. The main methods for verification are reviews and testing. Testing is also the preferred method to perform validation.

The following measures for SRESW, or SRASW written in FVL, for components with PLr a-d are given in the standard:

• software safety lifecycle with verification and validation activities
• where using software-based measures for control of random hardware failures, verification of correct implementation
• functional testing, e.g. black box testing

The following additional measures for SRESW, or SRASW written in FVL, for components with PLr c-d are given in the standard:

• coding verification by walk-through/review with control flow analysis
• extended functional testing, e.g. grey box testing, performance testing or simulation

For SRASW written in LVL for components with PLr a-e, the following requirements are given in the standard:

• functional testing

For SRASW written in LVL for components with PLr from c to e, the following additional measures with increasing efficiency (lower for PLr of c, medium for PLr of d, higher for PLr of e) are required or recommended:

**Review:**
The safety-related software specification shall be reviewed, made available to every person involved in the lifecycle.

Testing:
- The appropriate validation method is black-box testing of functional behaviour and performance criteria (e.g. timing performance);
- For PL = d or e, test case execution from boundary value analysis is recommended;
- Test planning is recommended and should include test cases with completion criteria and required tools;
- I/O testing shall ensure that safety-related signals are correctly used within SRASW.

Verification:
- Using review, inspection, walkthrough or other appropriate activities. Verification is only necessary for application-specific code, and not for validated library functions.

The subsequent subsections of Section 8.5 describe the methods for verification and validation in more detail.

8.5.1 Review
Reviews should be performed by independent persons not directly involved in developing the material to be reviewed. They can be from different areas in the organization or external persons. Feedback from earlier reviews may also generally improve the review. Often, active support and acceptance is necessary from management since extra resources are needed in the short term.

Reviews can be divided into three types depending on who has the initiative when the review is performed. In walkthroughs the creator has the initiative of guiding the reviewer through the documentation or source code while inspections are performed where the reviewer has the initiative. There are also informal reviews more similar to brainstorming activities where anyone has the initiative.

The more formal a review is, the more it can be controlled enabling the review to be more easily repeated and easier to maintain the quality. However, formal reviews generally require more resources than informal reviews.

8.5.2 Test methods
Software testing is performed by applying test vectors (input) in order to discover defects in the software and can be performed on various levels:
- Unit – no dependence on outside systems
- Integration – dependence on other units and outside systems
- System – complete system

Testing may be divided into black box and white box testing describing the point of view the tester takes when designing test cases. White box testing utilizes knowledge about the internal structure and source code of the system to derive test cases and
perform the tests. In black box testing, the testers are only aware of what the software is supposed to do, not how it does it. Grey box testing is similar to black box testing but utilizes some knowledge about the internal structure of the system for designing test cases.

Apart from functional requirements, testing may also be performed for non-functional requirements like performance testing, usability testing, fault injection and security testing. Fault injection is a way to accelerate the occurrences of faults for measuring the dependability and thoroughly testing the safety mechanisms of a system. Common methods for security testing include penetration testing, vulnerability scanning (vulnerability analysis, virus and malicious code detection systems) and fuzz testing (or fuzzing) where the system is subjected to invalid or random data in a semi-controlled way in order to reveal security deficiencies.

### 8.5.3 Test coverage

In white box testing, the concept of test coverage is important. Statement coverage is a simple test coverage metric denoting the percentage of statements in the program to be executed at least once by the applied test cases. Branch coverage is the corresponding metric taking into account the branches of each control structure as well. However, for safety critical systems, statement or branch coverage may not be sufficient for arguing for a satisfying test result. Instead condition coverage may be required where each boolean sub-expression is evaluated to true and false. Variants of condition coverage, often required by safety standards, include modified condition/decision coverage (MC/DC) or multiple condition coverage (MCC). These metrics view the software on the control flow level by dividing it into blocks/branches instead of statements and test cases are derived for traversing as many of the blocks/branches in as few tests as possible.

Some methods used for deriving test cases include analysis of requirements, equivalence classes and boundary values, e.g., see Figure 24. The figure shows an example of how to derive test cases for a function which is active for a speed > 60 km/h. The speed is represented as an integer and may be between 0 and 250 km/h.

---

**Figure 24** Deriving test cases
Test cases may also be derived by *error guessing*. These are test cases that may cause potential errors in the system when applied. They are typically derived based on data collected from previous experiences or expert judgement.

### 8.5.4 Validation

The purpose of the software validation is to check that the software fulfills the requirements specified in the safety-related software validation plan. It is important to follow this safety-related software validation plan. The results of validation shall be documented and action plans on detected errors are specified.

Testing shall be the main validation method.

It is preferable if the validation is carried out by persons that are independent of design (to an appropriate level).

In some situations it can be more efficient to only have one validation that covers both hardware and software, instead of having two different validation activities (one for hardware and one for software)

### 8.6 Software modifications

It is important to be careful when performing modifications in safety-related software. The standard says that appropriate software safety lifecycle activities shall be performed. This means that before you change the code it is important to investigate how this change will influence the earlier work performed and whether certain safety life cycle phases needs to be updated.

The following additional requirements on how to perform software modifications are given in the standard for SRESW, or SRASW written in FVL, for components with PL_c-d:

- *impact analysis and appropriate software safety lifecycle activities after modifications*

For SRASW written in LVL for components with PL_c from c to e, the following additional measures with increasing efficiency (lower for PL_c of c, medium for PL_c of d, higher for PL_c of e) are required or recommended:

- *After modifications of SRASW, impact analysis shall be performed to ensure specification. Appropriate lifecycle activities shall be performed after modifications. Access rights to modifications shall be controlled and modification history shall be documented.*

ISO 26262 may be used as an example to give guidelines on how to perform impact analyses. According to ISO 26262-8 8.4.3 the following shall be addressed when performing the impact analysis for each change request:
• the type of change request,
  NOTE Possible types of changes include: error resolution, adaptation, enhancement, prevention.
• the identification of the work products to be changed and the work products affected,
• the identification and involvement of the parties affected, in the case of a distributed development,
• the potential impact of the change on functional safety, and
• the schedule for the realisation and verification of the change

8.7 Parameterization

In the standard much focus is placed on software-based parameterization. The reason for this is that software-based parameterization can be seen as an untyped programming language that can influence the safety of the machinery.

Following requirement can be found in Chapter 4.6.4 in the standard:

Software-based parameterization of safety-related parameters shall be considered as a safety-related aspect of SRP/CS design to be described in the software safety requirements specification.

Parameterization shall be carried out using a dedicated software tool provided by the supplier of the SRP/CS. This tool shall have its own identification (name, version, etc.) and shall prevent unauthorized modification, for example, by use of a password.

![Diagram](image.png)

Figure 25 Parameterization carried out using a dedicated software tool

In Chapter 4.6.4 in the standard it is possible to find more detailed information about which requirements that are placed on parameterization tools and different alternative ways to fulfil these requirements.

8.8 Use of previously developed components

It is also possible to use pre-developed components (e.g. PLC:s) containing SRESW which not complies to the software requirements of EN ISO 13849-1 by fulfilling the following additional conditions:

• The component shall be commercially available, i.e. it is an off-the-shelf product that is already in use in other products on the market
• The hardware of the pre-developed component shall comply with all applicable requirements of EN ISO 13849-1. E.g. the structure, DCavg, and MTTF_D shall be specified in the datasheet for the component

If the above conditions apply, then the pre-developed component(s) may be used in the SRP/CS implementing safety functions limited to:

a) PL a or b and the SRP/CS complies to Category B, 2 or 3
b) **PL c or d** and the SRP/CS complies to Category 2 or 3 and where different (diversified) pre-developed components are used in the two channels

These requirements may for an example be used practically for:

- Argumentation that control systems containing SRESW and that were developed before EN ISO 13849-1 was released still may be used according to the current version of the standard
- Allowing a means for usage of subsystems containing especially advanced SRESW for which there are no sufficient state-of-the-art software development process available (e.g. video or LIDAR based sensor systems for obstacle detection)
9 Validation

The SRP/CS must be demonstrated as "safe enough" for the intended application. All safety functions must provide the intended level of safety. The procedures and conditions to be followed in safety validation are specified in standard EN ISO 13849-2.

The validation shall demonstrate that the specified safety functions, the category achieved, and the performance level achieved by the safety-related parts of a control system (SRP/CS) are designed in accordance with standard EN ISO 13849-1.

The validation of non-safety-related functionality will also be important for the quality of the machine, but is not addressed in this report.

It should be noted that the term “validation” is not clearly defined in EN ISO 13849-1. However, in IEC 61508-4, clause 3.8.2 validation is defined as: “confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled”.

Furthermore, the definition is supported by the following explanation (as a note): “Validation is the activity of demonstrating that the safety-related system under consideration, before or after installation, meets in all respects the safety requirements specification for that safety-related system.”

9.1 Validation planning

A validation plan should be written before the tests are started. All requirements of importance for the functional safety of the SRP/CS must be identified, often by making references to the safety requirements specification.

The validation plan shall identify and describe the requirements for carrying out the validation process. Any specific requirements on the test conditions must be explained, e.g. environmental conditions or operational conditions.

The validation plan shall also identify the means to be employed to validate the specified safety functions, and which test and analysis methods to be applied. If a specific test method or test standard will be applied, it shall be specified.

The validation plan should also explain which organisations, departments or individuals will be responsible for the validation. The level of independence of the validation staff from the development staff should be specified.

If there are previously validated safety-related parts, it may be enough to refer to previously performed validation. A safety component such as a safety-PLC will have a Declaration of Conformance, or an EC Type Approval, which may be used as a proof of previously performed validation activities. The achieved Performance Level (PL), the category, and the numerical values (MTTFd, DC) needed for deciding the overall PL of the SRP/CS must be available for the safety component.
9.2 Validation of safety requirements

The specification of the required characteristics of each SRP/CS shall be validated. Input/output logic will be the basis of the validation but also timing constraints such as reaction times, time discrepancy and timeouts need to be validated.

It will probably not be possible to test all combinations of input signals. The test cases have to be selected to achieve efficient testing with the available testing resources. A good understanding of the design of the SRP/CS will support the selection of test cases. The understanding may be helped by adequate documentation such as

- block diagram with a functional description
- circuit diagrams
- functional descriptions
- time sequence diagrams

A comprehensive test suite for functional testing is often the best way to document conformance with the specification. A well written specification of the safety functions may partly be written to support the test plan. A clear specification of verifiable and measurable outputs will support the test plan.

9.3 Validation of reached category

Safety validation of the category is very much about validation of the behaviour at fault. The effect of faults in the SRP/CS is studied to decide the effect on the control of the machine.

A Failure Modes and Effects Analysis (FMEA) is performed by listing all faults (using e.g. EN ISO 13849-2, Annex A – D) and deciding the effect on the control system. All components affecting the SRP/CS shall be included, and the fault mode of each component shall be analysed. An FMEA can require extensive work for a SRP/CS with many components. But a well performed FMEA will be a good support to declare that the behaviour at fault is well analysed.
### Figure 26  Example of an FMEA [IEC 60812]

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Component</th>
<th>Function</th>
<th>Failure mode</th>
<th>Failure effect</th>
<th>Detection method or symptom</th>
<th>Redundancy provided</th>
<th>Mode failure rate severity level</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.1</td>
<td>Heater system (12 off, 6 on at each end, only in use when machine non-operational)</td>
<td>All</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>NOTE: The machine may overheat if heaters do not turn off automatically when nothing</td>
</tr>
<tr>
<td>20.1.1</td>
<td>Heaters</td>
<td>To heat up enclosure</td>
<td>by sc/c burnt out heater</td>
<td>Reduced heating</td>
<td>a) Temp indication &lt;5° above ambient b) Supply, fuse, or circuit breaker monitored</td>
<td>All in parallel, no supply redundancy</td>
<td>1.2</td>
<td></td>
</tr>
<tr>
<td>20.1.2</td>
<td>Heater terminal box, terminals, cable</td>
<td>Connect supply to heaters</td>
<td>as sc/c terminal or cable cut hot sole, three, six or all heaters by sc/c terminals (bending)</td>
<td>Loss or reduction of heating, condensation</td>
<td>Loss of all heating, possible condensation</td>
<td>Temperature &lt;5° above ambient</td>
<td>0.5</td>
<td>One earth fault should not fail system</td>
</tr>
</tbody>
</table>

| Totals | | | | | | | 2.0 | |

© RISE Research Institutes of Sweden
EN ISO 13849-1 states that the following fault criteria shall be taken into account:

- if, as a consequence of a fault, further components fail, the first fault together with all following faults shall be considered as a single fault;
- two or more separate faults having a common cause shall be considered as a single fault (known as a CCF);
- the simultaneous occurrence of two or more faults having separate causes is considered highly unlikely and therefore need not be considered.

EN ISO 13849-1 also points out the possibility of fault exclusion. A fault is a compromise between technical safety requirements and the theoretical possibility of occurrence of a fault. Certain faults may have to be excluded from the safety validation to make it feasible. When faults are excluded, it must be justified in the technical documentation. The validation plan may explain which faults are not covered.

Note that fault exclusion is applied to single failure modes of a component individually, not to the component as a whole. It is not likely that every failure mode of a component can be excluded.

Annexes A, B, C and D of standard EN ISO 13849-2 lists faults possible to exclude for certain types of components.

Standard EN ISO 13849-1 explains that fault exclusion can be based on

- the technical improbability of occurrence of some faults,
- generally accepted technical experience, independent of the considered application, and
- technical requirements related to the application and the specific hazard.

The FMEA is often started with an analysis based on circuit diagrams. The effects of many faults will be fairly straightforward to decide by analysis. For certain SRP/CS, it may even be sufficient to base the validation of category only on analysis. But in most cases it will be required to physically insert some faults into the circuits of the SRP/CS. This will serve to confirm the analysis on complex designs, but also to gain experience and confidence is the control system.
9.4 Validation of reached performance Level

The Performance Level (PL) is validated by analysis of the MTTF, DCavg and CCF. The structure of the SRP/CS, i.e. the category, will also affect the PL.

The probability of failure of the SRP/CS can be calculated by considering all components and the probability of their fault modes. The analysis method is called Failure Modes, Effects and Diagnostics Analysis (FMEDA). This is similar to the FMEA used to determine the behaviour at fault, with the important difference that numeric values are included for the probability of failure of the components. (Figure 27.)

The FMEDA is based on the quantitative failure rates and the distribution of failure modes for all components. Also the probability of the system to detect internal failures via automatic on-line diagnostics is considered. The automatic diagnostic capability is important to achieve and maintain reliability in increasing complex systems.

Examples of failure rates of components are given in Annex B of standard EN ISO 13849-1. Failure rates can also be obtained from the technical report IEC 62380, and form reports summarising experiences from use in industry. (Figure 28.)

The probability of dangerous failure for every fault mode of each component can be added to calculate the probability of dangerous failure for the whole SRP/CS. Tool support will be needed since the calculations for most systems will require input of many figures and calculations for every fault mode. FMEDA software toolkits can be found on the commercial market. Own spreadsheet calculations may be sufficient for small systems.

Figure 27 Example of an FMEDA worksheet [www.ansys.org]
### Failure rate for $t_j=40^\circ C$

<table>
<thead>
<tr>
<th></th>
<th>$\lambda_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silicon transistors</td>
<td>0.75</td>
</tr>
<tr>
<td>- Bipolar; npn; pnp</td>
<td></td>
</tr>
<tr>
<td>- MOS p, n ; FET</td>
<td></td>
</tr>
<tr>
<td>Gallium Arsenide transistors</td>
<td>0.3</td>
</tr>
</tbody>
</table>

### Failure distribution

| Material       | State       | %  |
|----------------|-------------|
| Silicon        | Short-circuits | 85 |
|                | Open-circuits | 15 |
| Gallium-arsenide | Short-circuits | 95 |
|                | Open-circuits | 5  |

Figure 28  Example of reliability data for low-power transistors [IEC 62380]
10 Achieved Performance Level

Standard EN ISO 13849-1 describes a number of different methods of determining the achieved PL for the identified SRP/CS.

10.1 Safety function implemented by one SRP/CS

For cases where one SRP/CS implements a safety function (or part of a safety function) Figure 29 can be used to determine the achieved PL based on the category, the $\text{MTTF}_D$ (for each channel), and the $\text{DC}_{\text{avg}}$ of that SRP/CS.

Examples of such cases are:

- Embedded control systems built for a specific purpose implementing a whole safety function
- Subsystems only implementing part of a safety function intended to be reused in different applications and/or released on the market separately (e.g. motor drives or safety components)

If input-, logic and the output parts individually are built up by the same designated architecture they can together be seen as one SRP/CS and in this case Figure 29 can be applied directly to determine the PL for the complete safety function.

![Figure 29](image_url) Relationships between categories, DC$_{\text{avg}}$, MTTF$_D$ of each channel and PL [EN ISO 13849-1, Figure 5]
To be able to use this method, the architecture of the SRP/CS must be in accordance with a designated architecture (category) according to section 4 in this report. The next step after the category is decided is to decide which diagnostic coverage that is reached for the SRP/CS according to section 6 in this report, also see Table 16 below.

Table 16 Diagnostic coverage (DC) [EN ISO 13849-1, Table 5]

<table>
<thead>
<tr>
<th>Denotation</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>DC &lt; 60%</td>
</tr>
<tr>
<td>Low</td>
<td>60% ≤ DC &lt; 90%</td>
</tr>
<tr>
<td>Medium</td>
<td>90% ≤ DC &lt; 99%</td>
</tr>
<tr>
<td>High</td>
<td>99% ≤ DC</td>
</tr>
</tbody>
</table>

The last step after the category and diagnostic coverage is chosen is to decide the MTTFD reached for the SRP/CS according to section 5 in this report and below Table 17.

Table 17 Mean time to dangerous failure of each channel (MTTFD) [EN ISO 13849-1, Table 4]

<table>
<thead>
<tr>
<th>Denotation of each channel</th>
<th>Range of each channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>3 years ≤ MTTFD &lt; 10 years</td>
</tr>
<tr>
<td>Medium</td>
<td>10 years ≤ MTTFD &lt; 30 years</td>
</tr>
<tr>
<td>High</td>
<td>30 years ≤ MTTFD &lt; 100 years</td>
</tr>
</tbody>
</table>

When applying Figure 29 it is important to be careful when the MTTFD level chosen (MTTFD = Low, MTTFD = Medium or MTTFD = High) covers more than one designated architecture (category) and in this case it will be necessary to go into EN ISO 13849-1, Annex K and check exactly at which value of MTTFD the PL alters from one level to another.

10.2 Safety function implemented by a combination of different SRP/CS

In cases when the safety function is implemented by a combination of different SRP/CS (subsystems) which already has been evaluated according to Figure 29 in section 10.1 in this report, Figure 29 cannot be used for evaluating the achieved PL for that safety function. Note that EN ISO 13849-1 only supports combination of SRP/CS by cascading, see Figure 30. In addition, it is particularly important to verify the interfaces between the different SRP/CS when cascading, so their individual achieved PL does not become obstructed by the “new” interconnections.
Figure 30  Contribution of different combined SRP/CS to the total failure frequency of the safety function

Examples of such cases are:

- Implementing the safety function by using different SRP/CS with different designated architectures (categories)
- Implementing the safety function by using pre-developed SRP/CS (subsystems)

There are two means for estimating the total achieved PL for combined systems:

Firstly, to sum all the PFH_D-values for each included SRP/CS according to Figure 30. The PFH_D value could be obtained from either the data sheet (information for use) provided with the SRP/CS, or from Annex K in EN ISO 13849-1 if the MTTF_D, the structure and the DC_avg is known for the SRP/CS. The total PFH_D value is then matched to an achieved PL for the whole combination in Table 18 below.

The total PFH_D value is the only factor deciding the achieved PL for the combination. The category used for each included SRP/CS will not influence the achieved PL. It will not be possible to state an overall category for the combination, if the categories used for each included SRP/CS are different.

Table 18  Performance levels [see table 3 in EN ISO 13849-1]

<table>
<thead>
<tr>
<th>PL</th>
<th>Average probability of dangerous per hour 1/h</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>$\geq 10^{-5}$ to $&lt; 10^{-4}$</td>
</tr>
<tr>
<td>b</td>
<td>$\geq 3 \times 10^{-6}$ to $&lt; 10^{-5}$</td>
</tr>
<tr>
<td>c</td>
<td>$\geq 10^{-6}$ to $&lt; 3 \times 10^{-6}$</td>
</tr>
<tr>
<td>d</td>
<td>$\geq 10^{-7}$ to $&lt; 10^{-6}$</td>
</tr>
<tr>
<td>e</td>
<td>$\geq 10^{-8}$ to $&lt; 10^{-7}$</td>
</tr>
</tbody>
</table>

NOTE: Besides the average probability of failure per hour other measures are also necessary to achieve the PL.

Secondly, if the achieved PL is known for all SRP/CS included in the combination, but (by some reason) not the PFH_D-value for one or several of the SRP/CS, then Table 19 in this report may be used. This table is however conservative and may therefor limit the number of SRP/CS that may be combined.
It is important to point out that Table 19 can only be used for SRP/CS that are connected in serial (in effect no redundant architecture). When using Table 19 the first step is to identify how many (called $N_{\text{low}}$) SRP/CS that has got the lowest PL (called $PL_{\text{low}}$).

As an example if $PL_{\text{low}}$ is c and $N_{\text{low}}$ is less than or equal to two than the PL reached for the complete safety function will be $PL=c$. If $N_{\text{low}}$ instead had been higher than two the PL reached for the complete safety functions would have been decreased to $PL=b$.

### 10.3 Using SRP/CS developed according to IEC 61508 or IEC 62061

For cases where a subsystem is developed according to IEC 61508 or IEC 62061 (and thus is SIL-rated instead of PL-rated) but is intended to be used as a SRP/CS within scope of EN ISO 13849-1,

Table 20 may be used for conversion from SIL to PL.

An example of such a case is:

- Usage of a safety-PLC
Table 20  PL correspondence for SIL compliant items [EN ISO 13849-1, Table 4]

<table>
<thead>
<tr>
<th>PL</th>
<th>SIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>No correspondence</td>
</tr>
<tr>
<td>b</td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>2</td>
</tr>
<tr>
<td>e</td>
<td>3</td>
</tr>
</tbody>
</table>

Note that Table 20 cannot be used “the other way around” since it is not included in IEC 61508. Thus a PL-rated SRP/CS (i.e. which is developed according to EN ISO 13849-1) cannot be converted to a SIL-rated system using this table and then used as a subsystem within scope of IEC 61508 or IEC 62061. Also refer to IEC/TR 62061-1 for further information.
11 Experiences of safety-related machine control

RISE has experience from a large number of safety evaluations made on machine control systems during the past 25 years. We are a notified body according to the European machinery Directive, and perform EC Type Examinations of logic units.

The state-of-the-art of machine control is to trust use of SRP/CS to control safety critical functions even if failures may endanger the life and health of the operator. This is true also for a complex design based on connected and communicating programmable electronic systems.

Software and complex electronics may be trusted. The risks are kept at a tolerable level by applying adequate techniques and measures to avoid faults during development and production, and to detect and handle faults at run-time before they manifest themselves as critical failures of the SRP/CS.

The concern for adequate safety has to be present all through the safety life cycle. Risks should be considered already in the first concept design, and then all through the development life cycle. Safety considerations must exist also during production, use and maintenance of the machine.

The risk analysis will show how different risk reduction levels are necessary. There will be requirements for low, medium and high performance levels. It will not be economically feasible to design all SRP/CS with the highest possible risk reduction. Design efforts must be spent where best benefits are achieved.

There are two major international standards for machine control; EN ISO 13849-1 and EN 62061. They both address how to design safety-related machine control systems. Concepts used in the two standards are different, but much of the techniques and measures applied are comparable.

This report highlights some considerations for machine control systems, but there are also other aspects which must be addressed. The digitalization of industry will lead to an even stronger networking of machines, and the sharing of data between units which earlier used to be working independently. Aspects of security and connected systems will influence safety engineering even more in the future.

SRP/CS are often developed based on commercial-off-the-shelf (COTS) components. Safety components such as light grids and two-hand control devices are well established. We also see other components used in safety-related control. Safety-PLCs, radio remote controls and power drives for electrical motors comprises both safety functions and non-safety control functions. The proper use and connection of the safety functions will be important. There are also software modules which can be re-used in new safety-related functions. Even open-source software may be considered to be used in safety-related control if it can be shown to be safe enough. The handling of component-based design of SRP/CS will be a key technology in the future.

Another trend in machine control is the operator working closely to the machine and to robots servicing the production. Older safety designs were often based on guards to
close the risk area, and to keep the operator away from the risk area. There is now often an aim to let the operator be present in risk areas without shutting down the machine. Robot control and intelligent safety functions are trusted to protect the operator also in the risk area.
Appendix A  Examples of diagnostic techniques

Table E.1: Cyclic test stimuli by dynamic change of the input signals

| EXAMPLE |
| DC = 90% |

The input device (I) is cyclically stimulated during operation

+V (e) – Electrical signal
(m) – Mechanical signal

Failure

Controller (PLC or Embedded)

Figure A.1

Table E.1: Plausibility check, e.g. use of normally open and normally closed mechanically linked contacts

| EXAMPLE |
| DC = 99% |

The input device (I, e.g. interlocking guard) is stimulated during operation

Mech. link

Guard response

Controller (PLC or Embedded)

Figure A.2
Table E.1: Cross monitoring of inputs (or outputs) without dynamic test

<table>
<thead>
<tr>
<th>EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Operator or operative part of the machine)</td>
</tr>
</tbody>
</table>

I1 and I2 – Redundant input devices, e.g. incremental encoders, potentiometers or switches
(e) – Electrical signal

<table>
<thead>
<tr>
<th>Table E.1: Cross monitoring of input (or output) signals with dynamic test if short circuits are not detectable (for multiple I/O)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXAMPLE</td>
</tr>
<tr>
<td>(Operator or operative part of the machine)</td>
</tr>
</tbody>
</table>

I1 and I2 redundant input devices, e.g. potentiometers or switches
(e) – Electrical signal
(m) – Mechanical signal

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller (PLC or Embedded)</td>
<td>Controller (PLC or Embedded)</td>
</tr>
</tbody>
</table>

Figure A.3

Figure A.4
Table E.1 Cross monitoring of input (or output) signals and intermediate results within the logic (L), and temporal and logical software monitor of the programflow and detection of static faults and short circuits (for multiple I/O) 

**EXAMPLE**

<table>
<thead>
<tr>
<th>Input- or Output device(s)</th>
<th>Logic</th>
</tr>
</thead>
</table>

Figure A.5

Table E.1 Indirect monitoring (e.g. Monitoring by pressure switch, electrical position monitoring of actuators)  

**EXAMPLE**

<table>
<thead>
<tr>
<th>Input- or Output device(s)</th>
<th>Logic</th>
</tr>
</thead>
</table>

Figure A.6
Table E.1: Direct monitoring (e.g., electrical position monitoring of control valves, monitoring of electromechanical devices by mechanically linked contact elements)

<table>
<thead>
<tr>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>99%</td>
</tr>
</tbody>
</table>

EXAMPLE

![Diagram](image1)

S – Spool position sensor
(e) – Electrical signal

Table E.1: Fault detection by the process

<table>
<thead>
<tr>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 99%</td>
</tr>
</tbody>
</table>

EXAMPLE

![Diagram](image2)

Note: This example is intended to illustrate that feedback signals indicate an erroneous state/operation of the machine (process) due to an internal failure. However, these feedback signals may not identify the actual failed component.

S – Temperature sensor
(e) – Electrical signal

Figure A.7

Figure A.8
Table E.1: Monitoring some characteristics of the sensor (response time, range of analogue signals, e.g. electrical resistance, capacitance)  

<table>
<thead>
<tr>
<th>Response failure</th>
<th>Input device(s)</th>
<th>Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>sensors</td>
<td>Logic</td>
<td>Inputs</td>
</tr>
<tr>
<td>[mA]</td>
<td>(e) – Electrical signal</td>
<td></td>
</tr>
</tbody>
</table>

Monitoring e.g. linearity of the response  

S – E.g. temperature sensor  
(e) – Electrical signal

EXAMPLE

Table E.1: Simple temporal time monitoring of the logic (e.g. Timer as watchdog, where trigger points are within the program of the logic)

<table>
<thead>
<tr>
<th>t – Cycle time</th>
<th>WDT – WatchDog Timer</th>
<th>Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDT &lt; WDT_timeout</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

EXAMPLE

DC = 60%

DC = 0 - 99%

Main() {
   Init();
   while(1){
      application_code();
      trig_WDT(); //set GPIOx = 1 in y ms
   }
}
Figure A.11

Table E.1: Temporal and logical monitoring of the logic by the watchdog where the test equipment does plausibility checks of the behaviour of the logic

<table>
<thead>
<tr>
<th>DC = 90%</th>
</tr>
</thead>
</table>

EXAMPLE

<table>
<thead>
<tr>
<th>MCU (L)</th>
<th>USART</th>
</tr>
</thead>
<tbody>
<tr>
<td>RST</td>
<td>WDT</td>
</tr>
<tr>
<td>If t &gt; WDT_timeout OR value error</td>
<td></td>
</tr>
</tbody>
</table>

/* Code parts only returns the correct values if each code parts were correctly executed */

```c
main()
{
  Init();
  while(1){
    int WDT_msg = 0;
    WDT_msg = application_code_part1();
    trigg_WDT(WDT_msg); //always hA3
    WDT_msg = application_code_part2();
    trigg_WDT(WDT_msg); //always h39
    WDT_msg = application_code_part3();
    trigg_WDT(WDT_msg); //always hF8
    etc..
  }
}
```

Logic

Table E.1: Checking the monitoring device reaction capability (e.g. Watchdog) by the main channel at start-up or whenever the safety function is demanded or whenever an external signal demand it, through an input facility

<table>
<thead>
<tr>
<th>DC = 90%</th>
</tr>
</thead>
</table>

EXAMPLE

<table>
<thead>
<tr>
<th>MCU1 (MCU2 WDT)</th>
<th>MCU2 (MCU1 WDT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>USART</td>
<td>USART</td>
</tr>
<tr>
<td>GPIOx</td>
<td>GPIOx</td>
</tr>
</tbody>
</table>

Two MCUs act as each others WDT continuously during operation. At e.g. Start-up an initiation sequence is executed where both MCUs times out each other in order to verify that the corresponding safety functions are demanded.

Figure A.12
Table E.1: Monitoring of outputs by one channel without dynamic test

<table>
<thead>
<tr>
<th>Input device(s)</th>
<th>Logic</th>
<th>Output device(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example

Controller (PLC or Embedded)

Inputs

Outputs

R – E.g. shunt resistor or current transformer
V1 – Hydraulic or pneumatic valve
(e) – Electrical signal

Figure A.13

Table E.1: Redundant shut-off path with monitoring of the actuators by logic and test equipment

<table>
<thead>
<tr>
<th>Input device(s)</th>
<th>Logic</th>
<th>Output device(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example

Controller (PLC or Embedded)

Inputs

Outputs

K1, K2 – Motor contactors
(e) – Electrical signal

Figure A.14
Appendix B Bibliography

B.1 Directive


B.2 Standards


IEC/TR 62061-1:2010 “Guidance on the application of ISO 13849-1 and IEC 62061 in the design of safety-related control systems for machinery”

IEC TR 62380 Technical report “Reliability data handbook – Universal model for reliability prediction of electronics components, PCBs and equipment”, 2004-0
Standards can be obtained from your standardisation body (e.g. www.sis.se)

Appendix C Abbreviations

Table C.1 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_{10}$</td>
<td>Number of cycles until 10 % of the components fail (for pneumatic and electromechanical components)</td>
</tr>
<tr>
<td>C</td>
<td>Duty cycle</td>
</tr>
<tr>
<td>$c$</td>
<td>Cross monitoring</td>
</tr>
<tr>
<td>DC</td>
<td>Diagnostic Coverage</td>
</tr>
<tr>
<td>$DC_{avg}$</td>
<td>Diagnostic Coverage Average</td>
</tr>
<tr>
<td>E/E/PE</td>
<td>Electrical/Electronic/Programmable electronic</td>
</tr>
<tr>
<td>HW</td>
<td>Hardware</td>
</tr>
<tr>
<td>I (I1, I2)</td>
<td>Input device, e.g. sensor</td>
</tr>
<tr>
<td>$i_{in}$</td>
<td>Interconnecting means</td>
</tr>
<tr>
<td>L (L1, L2)</td>
<td>Logic</td>
</tr>
<tr>
<td>m</td>
<td>Monitoring</td>
</tr>
<tr>
<td>MTBF</td>
<td>Mean Time Between Failure</td>
</tr>
<tr>
<td>MTTF</td>
<td>Mean Time To Failure</td>
</tr>
<tr>
<td>MTTF_D</td>
<td>Mean Time To Dangerous Failure</td>
</tr>
<tr>
<td>MTTR</td>
<td>Mean Time To Restoration</td>
</tr>
<tr>
<td>O (O1, O2)</td>
<td>Output device, e.g. main contactor</td>
</tr>
<tr>
<td>OTE</td>
<td>Output of TE</td>
</tr>
<tr>
<td>PFH_D</td>
<td>Probability of Dangerous Failure per Hour</td>
</tr>
<tr>
<td>PL</td>
<td>Performance Level</td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable Logic Controller</td>
</tr>
<tr>
<td>PL_r</td>
<td>Performance Level Required</td>
</tr>
<tr>
<td>SIL</td>
<td>Safety Integrity Level</td>
</tr>
<tr>
<td>SRASW</td>
<td>Safety-Related Application Software</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Explanation</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SRESW</td>
<td>Safety-Related Embedded Software</td>
</tr>
<tr>
<td>SRP/CS</td>
<td>Safety-Related Part of a Control System</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>T₁₀</td>
<td>Mean time until 10% of the components fail</td>
</tr>
<tr>
<td>TE</td>
<td>Test Equipment</td>
</tr>
</tbody>
</table>
Through our international collaboration programmes with academia, industry, and the public sector, we ensure the competitiveness of the Swedish business community on an international level and contribute to a sustainable society. Our 2,200 employees support and promote all manner of innovative processes, and our roughly 100 testbeds and demonstration facilities are instrumental in developing the future-proofing of products, technologies, and services. RISE Research Institutes of Sweden is fully owned by the Swedish state.